INFORMATION BASED ADAPTIVE PATH PLANNING AND SAMPLING FOR ENVIRONMENTAL MONITORING

RAJAT MISHRA

(B. Tech, VIT University (Vellore), India)

SUPERVISORS

Associate Professor Mandar Chitre

Associate Professor Sanjay Swarup

EXAMINERS

Associate Professor He Jianzhong

Associate Professor Prahlad Vadakkepat

A THESIS SUBMITTED FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

NUS GRADUATE SCHOOL OF INTEGRATIVE SCIENCES AND ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2019

Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Rajat Mishra

August 2019

Acknowledgments

My research would not have been possible without the aid and support of a group of people. I take this opportunity to express my appreciation to them in this acknowledgement.

First and foremost, I would like to express my sincere gratitude to my PhD supervisors, A/Prof. Mandar Chitre and A/Prof. Sanjay Swarup, for providing me the opportunity to work in this field of research. Their commitment and passion for research and solving applied problems has motivated me to become a better researcher. Working with them has been a great learning experience and their guidance has always encouraged me to do quality research work. I could not have imagined having better supervisors for my PhD work. I would also like to thank my Thesis Advisory Committee (TAC) chair, A/Prof. Prahlad Vadakkepat, for his valuable feedback throughout the course of my work.

I also thank my fellow labmates in ARL[1] and Integrated Urban Waterways group at SCELSE[2]. I would like to express my gratitude to Mr. Koay Teong Beng and Mr. Wu Yusong for all their support in successfully completing the field experiments present in this thesis. I thank Dr. Aditya Bandla, Mr. Chua Si Hao and Mr. Ooi Qi En for taking the time out of their schedule to process the samples collected as part of this work. Adding to these names, I would like to thank Dr. Prasad Anjangi, Dr. Ahmed Mahmood, Dr. Hari Vishnu, Dr.

[1]Acoustic Research Laboratory (ARL), Tropical Marine Science Institute (TMSI), National University of Singapore - https://arl.nus.edu.sg/

[2]Singapore Center for Environmental Life Sciences Engineering, Singapore - http://scelse.sg/

Bharath Kalyan, Mr. Liang Jie Wong and Dr. Gourvendu Saxena for the many interesting discussions we had and for making my PhD journey enjoyable and fun. I would also like to thank Ms. Ong Lee Lin for organizing all the amazing lab get-togethers and helping me in research writing.

Finally, I am indebted to my late parents, Mr. Rakesh Mishra and Ms. Rajni Mishra, for instigating the dedication and motivation to always pursue my interests. Without their unconditional love and encouragement I would never have enjoyed so many opportunities. Special thanks go to my beloved elder sisters, Ms. Richa Tiwari and Ms. Rashi Dixit, for always supporting me in my professional and personal endeavours. Last but not the least, I thank all my family members and friends both in Singapore and abroad who have made my PhD journey a happy and meaningful one.

Abstract

The environmental processes taking place in oceans, lakes, rivers, and other water resources are of interest to a broad range environmental scientists. These processes typically span across large areas and exhibit spatio-temporal variations, which makes the task of environmental monitoring challenging. This task is increasingly being automated using robots such as drones and underwater vehicles.

The constantly changing spatial and temporal distribution of the environmental fields put a natural bound on the area that can be surveyed before the field varies significantly. Due to these natural bounds, the robot's path should be adjusted to capture the maximum amount of information within a limited amount of time and provide a good estimate of the environmental field. The branch of algorithms that solves this information maximization problem is termed as Informative Path Planning (IPP).

In this work, we present a comprehensive approach for both monitoring and physical sample collection for understanding the environmental processes. We first present the two IPP algorithms to perform the monitoring task with bounds on the mission time and provide a good estimate of the environmental field. These algorithms adapt the path during the task based on the recently collected information, which is termed as online IPP. We make use of Sparse Gaussian Processes (SGP) for the field estimation and path planning methods for coordinating a single robot or a team of robots. The performance of these algorithms are benchmarked against conventional lawn mower paths and validated through field experiments. These validation

experiments show that our frameworks outperform the conventional monitoring methods. Moreover, we examine the biological relevance of the field estimated using such frameworks and show that informed sampling can yield substantial information about the environment. We also perform a quality analysis of the samples collected using robots to demonstrate the use case of adaptive frameworks in environmental monitoring.

The use of online or offline IPP algorithms solves the problem of monitoring environmental fields. However, the physical sample collection generally happens after field estimation is finished. Such a sequential approach results in a temporal delay between the actual sample collection and the completion of the monitoring task. In an ideal scenario, these two actions should be performed at the same instance. This problem of simultaneous sampling and monitoring of an environmental field is addressed by another algorithm present in this work. This algorithm provides a framework to collect samples from the hotspots, while using SGP for estimating the field and decision making frameworks to decide the sampling location. Using simulations, we show that our framework performs well in estimating the field and meets the scientific objectives for collecting samples.

In addition to the development of these frameworks, vehicle designs have gone through modifications to improve the robot's endurance and maneuverability. Most of these new designs do not have the conventional AUV structure and thus using simple vehicle dynamics is not straight-forward. We present a data-driven framework to address this problem of system identification for AUVs. This framework is based on an artificial neural network, which learns a model of robot's system dynamics using field data. We use field data to benchmark the performance of our approach and show it outperforms the standard state space model.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Motivation for using Robots in Environmental Monitoring

Urban freshwater systems are critical components of densely populated large- and mega-cities [9]. The management and regulation of such systems are largely based on scientific recommendations, which are obtained using geochemical datasets [10]. The microorganism communities that play a key role in catalyzing the geochemical changes are generally considered as a single lumped variable [11]. Such an approach fails to identify and account for the spatial and temporal variations in environmental parameters and the interactions of associated micro-organism communities. On the contrary, such small-scale processes by these organisms add to the variations in the water quality and this influence can be observed at a macro-scale level [12]. Capturing such variations is critical for the development of water quality models in urban freshwater systems [13]. Hotspots or hot moments of chemical constituents are excellent locations for understanding the dynamics between the water resources and the associated microorganism communities. Furthermore, experiments for studying the microorganisms and its activities are generally driven by strong statistical analysis. This means the data capture methodology and the sampling regimes are equally important for understanding the microorganisms well [14].

In order to locate and capture these hotspots, the spatio-temporal distribution of the environmental parameters over the survey field is a necessary information. In this thesis, such survey problems are referred as monitoring tasks. A general

difficulty with monitoring tasks is modelling the spatial and temporal changes of the field. However, this difficulty can be mitigated by assuming the environmental fields are temporally static over a short time period. This assumption reduces the modelling complexity but adds a constraint on the survey time and thus the information collection process for estimating the field. The use of Informative Path Planning (IPP) frameworks is a good solution to such problems as these frameworks aim at collecting information efficiently. However, most of the these frameworks focus on path planning approaches to optimize for temporal constraint [15, 16, 17, 18], and do not address the problem of learning the model from a large dataset. A framework that can optimize both on planning as well as model learning is still an open problem, and this applies to both single- and multi-robot systems. We present two monitoring frameworks, *AdaPP* and *m-AdaPP*, to address these problems.

Once the hotspots are located, the scientists decide the location of taking a sample. We have termed such a task of sample collection as sampling task. Generally, the sampling task does not happen as frequently as the monitoring task but it can be used for solving the sample collection problem. In such an approach, the monitoring framework will provide an estimate of the field and this will be followed by sample collection, which will introduce a delay between monitoring and sampling tasks. In an ideal scenario, this delay should be as small as possible and some frameworks have suggested approaches to perform monitoring and sampling task simultaneously to reduce this delay [19, 20]. We present another single-robot framework that aims at solving this delay problem while also providing a control over the sampling behaviour.

Another recent advancement in using the robots for environmental monitoring is the development of new vehicles, which are designed to have higher endurance and increased maneuverability. However, this advancement has increased the complexity of vehicle dynamics and thus using off-the-shelf vehicle dynamics model for designing a controller is not correct. We suggest a data-driven approach to model such vehicles and compare its performance with a simple dynamics

model heavily used in designing controllers.

1.2 Goals

- To provide frameworks for environmental monitoring task and validate their performance via simulations and field experiments.
- To examine the biological relevance of the field estimated using the environmental monitoring frameworks.
- To design a framework which can perform monitoring and sampling task simultaneously using only the data collected during the survey.
- To provide a data-driven approach for system identification of vehicles with complex system dynamics.

1.3 Contributions

Our work primarily focuses on solving the field estimation and data collection problem for environmental monitoring. The specific contributions of this thesis are listed as follows:

- Formulated an adaptive single-robot framework that uses kernel information for adaptive path planning and provides a performance improvement in estimating environmental fields over conventional methods.
- Developed an adaptive multi-robot framework that efficiently coordinates the motion of a team of robots to provide a good estimate of the environmental field within a stipulated amount of time.
- Validated the performance of the multi-robot framework for providing better estimates compared to conventional lawn mower paths via field experiments.
- Examined the biological relevance of informed sampling using the fields estimated by the adaptive frameworks in freshwater reservoir.

- Developed a framework for collecting physical environmental samples while collecting the field data and estimating the field.
- Designed a system identification framework by using multi-layer perceptron and physics based features for estimating vehicle dynamics.

1.4 Thesis Organization

Chapter 2 provides a review of the relevant literature for the scope of this work. We discuss the existing methods and list the set of open problems that help in understanding the impact of our research.

Chapter 3 introduces the problems with the current monitoring approaches for experiments using single-robot and suggests an information based path planning framework for the estimating environmental fields. We discuss two single-robot frameworks in detail and provide comparison results with the current monitoring approaches via simulation using real field data. We term these two frameworks as *AdaPP* and *k-AdaPP*. The key difference between these two frameworks is in the planning iteration and Chapter 2 discusses this in detail. The results show that our algorithm performs well as compared to conventional monitoring approaches.

Chapter 4 provides an extension of the single-robot framework to a multi-robot setup and we term this multi-robot framework as *m-AdaPP*. This framework manages the coordination between the robots to provide an estimate of the field within a limited mission time. We discuss the use of dynamic programming in our finite horizon problem and explain our centralized approach for planning. We benchmark the performance of our this framework and provide results that show our multi-robot coordination can provide better field estimates in half or one-third the mission time set for single-robot framework. We further validate the performance of this framework via field experiments against the conventional lawn mower approach and show that our framework outperforms it.

In Chapter 5, we examine the biological relevance of the fields estimated using the multi-robot framework. We use the estimated fields to find the hotspots and

coldspots for each survey over different days and collect physical water samples to analyze the microorganisms in these samples. We use DNA sequencing to identify different microorganisms and show that the two regions, hotspots and coldspots, have significantly different communities for different survey areas and different survey days. This is an encouraging use case of the multi-robot algorithm as it signifies the importance of informed sampling for microorganism studies.

Chapter 6 introduces a framework for coordinating both physical water collection and estimation of the field as a combined tasks. We explain the use of exploration-vs-exploitation method for our combined task and suggest a modification to our single-robot algorithm to address this simultaneous monitoring (exploration) and sampling (exploitation) task. We term this framework as *SAM* and compare the performance of our algorithm with exploration-only and exploitation-only approaches and discuss the results in detail.

Robots used for environmental monitoring can cause disturbance in the environmental field that it is sensing. The first step to quantify this disturbance is to have an accurate vehicle's dynamics model. We suggest a neural network based method for estimating vehicle dynamics in Chapter 7. We discuss the universal approximation capability of neural networks and explain the use of this capability in approximating the vehicle model. We suggest the use of physics based features for estimating the vehicle dynamics and compare its performance with commonly used dynamics equation. We show our neural network based approach predicts more accurate system variables compared to the commonly used equations and thus can be used in designing efficient controllers.

Finally, we summarize our key results and present the future research directions in Chapter 8.

Chapter 2

Literature Review

2.1 Overview

The environmental processes taking place in oceans, lakes, rivers, and other water resources are of interest to a broad range of scientists. These processes typically span across large areas and exhibit spatio-temporal variations, which makes the task of environmental monitoring challenging. Increasingly, this task is being automated using Autonomous Underwater Vehicles (AUVs) [21], which is an advancement over the use of static sensor buoys for monitoring. This Chapter briefly reviews some of the existing work on AUVs and its application in environmental monitoring. Section 2.2 briefly explains the importance of environmental monitoring and sampling. Section 2.3 discusses various frameworks used for environmental monitoring, which expands to simultaneous sampling and monitoring frameworks in Section 2.4. Finally, Section 2.5 reviews some of the new AUV designs and addresses the problem with system identification of these new AUVs.

2.2 Environmental Monitoring and Sampling

Freshwater systems such as reservoirs or lakes are the main source of drinkable water and therefore, these systems are critical components of densely populated large- and mega-cities [9]. The management and regulation of such systems has largely been based on scientific recommendations, informed purely by indicators such as chemical and geological characteristics [10]. However, environmental processes in these freshwater systems are heterogeneous and exhibit both spatial

and temporal variations [22, 23, 24]. The freshwater systems are also a home to communities of microorganisms that play a key role in impacting the water-quality indicators, and to the variations in the water-quality parameters [12]. These microrganism communities have both feed-forward and feed-back interaction with the large communities of plants and animals centered around waters, which is termed as freshwater biome. Furthermore, rare places and rare events have the potential to exert a disproportionate influence on the movement of elements at the scale of landscapes and ecosystems [12, 25]. Capturing such variations is extremely important for the development of water-quality models in the freshwater systems [13, 26], which is generally overlooked due to lack of prior information on environmental parameters gradients. Hotspots of chemical constituents and microbial activity are excellent locations where high variations in environmental parameters and associated microbial communities is available with respect to the bulk water in the aquatic environments.

2.2.1 Importance of Hotspots

Water quality hotspots in aquatic systems in urban areas can be both harmful and beneficial to the ecosystem, depending on the biological and chemical characteristics of the hotspot [27, 28, 29]. For example, hotspot formation for nitrogen (N) and phosphorus removal (P) are beneficial in natural water systems, especially in the receiving waters of urban areas, such as reservoirs [27, 28]. On the other hand, hotspots of contaminants formed due to flow characteristics of water body might influence the local microbial ecosystem by changing dissolved oxygen (DO) levels and adversely impact the natural processes by altering the communities of microorganisms [30, 31]. Interestingly, urban freshwater systems use aerators to improve water quality in reservoirs, which develops artificial hotspots of dissolved oxygen (DO). However, how such artifical hotspots impact the existing microorganism communities is not well explored. Moreover, most of these scientific experiments that study the effects of these hotspots are generally centered around specific site locations, which are spatially small in comparison to

the expanse of water resources. It has been shown that a good spatial-temporal resolution of an environmental parameter not only improves the quality of the data but also drives the scientific conclusions [32]. Therefore, based on the experimental design, it can be beneficial to increase the survey area of environmental parameters and help the scientists to easily locate the hotspots.

In addition to finding the hotspots, it is equally important to sample these regions and discover the associated microbial communities. Sampling from hotspots of oxygen minimum zones has helped in answering a microorganism's role in terrestrial loss of N in inland waters [33]. The importance of hotspot sampling is not limited to microbial communities only but it can be also used to study the effects of fish population on biological and chemical parameters of water resources [34]. Such studies show that sampling from hotspots can provide us the insights on the micro and macro ecology basis of development of these hotspots and explain the dynamics of freshwater systems. Therefore, estimating the distribution of hotspots and sampling from the maximum variations of environmental parameters such as dissolved oxygen is critical in understanding the environmental processes in the freshwater systems.

2.2.2 Current Practices in Environmental Monitoring

Buoys and Floats equipped with environmental sensor are used to monitor water-quality across different water resources such as oceans and freshwater systems. One of the widely used platforms is Argo Floats [35], which has helped in various scientific studies [36, 37, 38, 39]. More recently, robotic systems such as AUVs are being increasingly utilized as fundamental data-gathering tools by scientists, catering to the need of monitoring and sampling the environmental processes [21]. A major portion of the AUVs are now designed to carry out scientific data collection missions [40, 7, 41, 42, 16]. Such robot-aided data collection has been also used in explaining biological processes [43, 44]. However, the use of these robots is still limited due to the complex spatio-temporal nature of the environmental parameters. In order to mitigate these drawbacks, information-

based planning frameworks have been suggested for environmental monitoring missions, making these robots more directed towards scientific data collection. These frameworks are discussed in detail in the following sections.

2.3 Frameworks for Environmental Monitoring

Environmental processes are normally spread across large distances, generally in the range of kilometers, and vary both spatially and temporally. In order to monitor these processes through environmental parameters such as pH or dissolved oxygen, it is ideal to do multiple blanket coverage of the survey area. A simple approach for doing multiple blanket coverage is to place static buoys using information from environmental modelling [45, 46]. Such an approach will provide good temporal resolution, however, it will be a resource intensive approach as each buoy will require environmental sensors, which will demand regular maintenance. This motivated developing actuated sensors, AUVs equipped with sensors, which can be used to provide both spatial and temporal resolution of an environmental field. Similar to buoy placement frameworks, AUVs can use the information from the environmental model and guide its motion to provide better coverage of the survey area. The frameworks that use such approaches of planning based on information are termed as Informative Path Planning (IPP) frameworks, which selectively samples the field and produces an approximation of the environmental field. The Table 2.1 states some of the IPP frameworks used for environmental monitoring.

One of the challenges in estimating environmental fields is the data collection process. In general, the IPP mitigates this challenge by evaluating informative paths using an information measure for unobserved locations[15, 16, 17]. The robot then traverses the path which provides maximum information as per a predefined criterion and collects data to give an estimate of the environment. In general, the IPP frameworks have three components: collecting data while traversing, adapting the robot's path to provide a good approximation of the field and learning a model of the environmental field. The first component is self

Framework	Single-Robot	Multi- Robot
Singh et al., 2006 [47]		×
Zhang et al., 2008 [48]	×	
Low et al., 2008 [49]		×
Binney et al., 2010 [50]	×	
Low et al., 2011 [17]		×
Cao et al., 2013 [4]		×
Kemna et al., 2017 [51]		×
Ma et al., 2017 [52]	×	
Hitz et al., 2017 [53]	×	
Mishra et al., 2018 [54]	×	

Table 2.1: A list of some Gaussian Process based Informative Path Planning frameworks used for environmental monitoring.

explanatory, whereas, the last two components are the key characteristics which define the behaviour of all IPP frameworks.

Adapting the robot's path to provide a good approximation represents the re-planning capability based on gathered information. Based on the frequency of this re-planning, the IPP frameworks can be classified as: Non-adaptive algorithms (offline) which commit to a path and do not adjust based on new observations and Adaptive algorithms (online) which alter the pre-planned paths on-the-fly based on the new observations. Several non-adaptive algorithms have been suggested in the past to solve for near-optimal paths [47, 18, 55, 56] using prior information of the field. However, the environmental parameters generally undergo temporal changes and thus the prior information may no longer be valid for planning for the robot's path. In contrast, adaptive algorithms have the advantage of allowing re-planning based on current observations.

The IPP frameworks can also be classified based on the number of robots it can coordinate. Any framework that can coordinate more than two robots can be classified as a multi-robot IPP framework [47, 17, 57, 51], whereas, frameworks

designed to coordinate just one robot are classified as single-robot frameworks [48, 53, 54]. Each of these two classes of frameworks have its own advantage. The planning step for single-robot frameworks is less complex as compared to multi-robot frameworks, however, covering large survey areas with a single robot may not feasible due to the limited on-board resources of a robot. On the contrary, multi-robot frameworks can easily mitigate the problem of covering large survey areas by division-of-labour but the coordination of the team of robots to efficiently utilize each robot's resources adds significant computational overhead to multi-robot IPP frameworks. Moreover, multi-robot frameworks gather more data in a short amount of time, which requires faster model learning techniques to achieve real-time performance for the overall system. Such problem of model learning are currently not addressed in the multi-robot frameworks and thus limits the usage to small datasets or small survey areas.

The next important component in the adaptive algorithm is the online approximation of the survey area as this approximation governs the future waypoints in a robot's path. For example, in the case of water quality monitoring, a good approach will be to use off-the-shelf simulators like Delft3D [58] or Regional Oceanic Monitoring System (ROMS) [59]. However, these simulators generally run on high performance clusters and such computational power is usually not available in robotic platforms. One good approach to combine these simulators with path planning is presented in [60]. In this approach, ROMS use the data from various sensors to produce velocity profiles on a remote server, which can then be used by the robot for path planning. However, in areas where the sensors for ROMS are not present or the spatial resolution of ROMS's forecast is poor, such an approach will not work.

A commonly used approach in Geostatistics [22, 23] is to assume that the spatio-temporal environmental field is realized from a probabilistic model called Gaussian Processes (GPs). The computational power required for learning a Gaussian Process model is comparatively much less than that required by simulators. Therefore, this approach has been used several times for path

planning [16, 48, 57, 49, 17]. In [53], GPs and an information criterion were used to plan paths for an AUV to segment the environmental field into three different level sets. Similarly, using GPs, a path planning algorithm based on entropy and information criterion is suggested in [4]. In all of these works, GP regression is done using all the data collected during the survey. In a practical scenario, a water quality sensor [61] can sense data at a frequency of 1 Hz and thus running a robot with this sensor for an hour will provide $3,600$ data points for learning the model. In simple terms, this means that the data collected during a survey can increase rapidly and therefore, the conventional method of doing GP regression is not feasible. This problem can be solved using sparse GPs. An explanation of how the sparse GPs can be integrated into a path planning framework is discussed in [52]. This recent work is directed towards long-term monitoring and overcomes the spatial and temporal changes by updating the GP model based on an information criterion. It is a good single-robot framework, however, the sparse GP point selection can be still improved with a more data-driven sparse GP variants such as [62]. Moreover, this framework may not work well for time sensitive missions as the planning step only plans for few waypoints ahead and does not adapt the path while traversing these few waypoints. Interestingly, the combination of such sparse GP models and time sensitive mission planning for the single- and multi-robot frameworks is still missing.

2.3.1 Sparse Gaussian Processes

GP models are commonly used for non-parametric regression problems [63], such as spatial data modeling [64], image thresholding [65] and soil modeling [66]. For spatial data regression problems, the training data set $\mathbf{D}$ consists of N input vectors $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^{N}$, where each vector is two dimensional, and corresponding target values $\mathbf{y} = \{y_n\}_{n=1}^{N}$. Moreover, in a standard GP regression formulation it is common to assume a Gaussian noise model to accommodate measurement noise. The likelihood of observed values $\mathbf{y}$ can be given as $p(\mathbf{y}|\mathbf{f}) = \mathcal{N}(\mathbf{y}|\mathbf{f}, \sigma^2\mathbf{I})$ where $\mathbf{f}$ is the underlying latent function and $\sigma^2\mathbf{I}$ is the noise term. Placing a

zero mean prior and a covariance function given by $\mathcal{K}(\mathbf{x}_n, \mathbf{x}_{n'})$ and parameterized by θ, the distribution for a new input $\mathbf{x}$ is given by

$$p(y|\mathbf{x}, \mathbf{D}, \theta) = \mathcal{N}(y|\mathbf{k_x}^\top(\mathbf{K}_N + \sigma^2\mathbf{I})^{-1}y, K_{\mathbf{x},\mathbf{x}} - \mathbf{k_x}^\top(\mathbf{K}_N + \sigma^2\mathbf{I})^{-1}\mathbf{k_x} + \sigma^2), \tag{2.1}$$

where $[\mathbf{k_x}]_n = \mathcal{K}(\mathbf{x_n}, \mathbf{x})$, $[\mathbf{K}_N]_{n,n'} = \mathcal{K}(\mathbf{x}_n, \mathbf{x}_{n'})$ and $K_{\mathbf{x},\mathbf{x}} = \mathcal{K}(\mathbf{x}, \mathbf{x})$. As it can be observed from Eq. 2.1, the computation time for large datasets will be high as the prediction, and even the training, scales with N^3 due to inversion of the covariance matrix, where N is the total number of datapoints. Sparse GPs overcome this problem by having sparse approximation of the full GP using only M points, where $M \ll N$. In general, the selection of this subset of M points is based on some information criterion [67].

A common problem with information criterion based sparse GP method is the absence of a good method to learn the kernel hyperparameters, because the subset selection and hyperparameter optimization are generally done independently. Moreover, when using *automatic relevance determination* [68] covariance function, learning bad hyperparameters can adversely affect the prediction performance. SPGP framework solves this problem by constructing a GP regression model which finds the active subset and hyperparameters in one smooth joint optimization.

2.3.2 Sparse Pseudo-inputs Gaussian Processes

In a standard GP model [63] with zero mean prior, the kernel function is solely responsible for estimating the mapping between the input vector and the target values as shown in Eq. 2.1. Therefore, assuming the hyperparameters of the kernel function are known, the predictive function is effectively parameterized by $\mathbf{D}$. In the case of SPGP, this parameterization is done using the pseudo data set $\overline{\mathbf{D}}$ of size $M \ll N$, which has pseudo-inputs $\overline{\mathbf{X}} = \{\overline{\mathbf{x}}_m\}_{m=1}^M$ and corresponding pseudo targets $\overline{\mathbf{f}} = \{\overline{f}_m\}_{m=1}^M$. The pseudo targets are denoted as $\overline{\mathbf{f}}$ instead of $\overline{\mathbf{y}}$

because these targets do not represent the observed values and therefore, adding the noise variance σ^2 can be omitted. The actual prediction distribution has the noise variance and is given as

$$p(y|\mathbf{x}, \mathbf{D}, \theta) = \mathcal{N}(y|\mathbf{k_x}^\top \mathbf{K}_M^{-1}\overline{\mathbf{f}}, K_{\mathbf{x},\mathbf{x}} - \mathbf{k_x}^\top \mathbf{K}_M^{-1}\mathbf{k_x} + \sigma^2), \tag{2.2}$$

where $[\mathbf{K}_M]_{mm'} = \mathbf{K}(\overline{\mathbf{x}}_m, \overline{\mathbf{x}}_{m'})$ and $[\mathbf{k}_x]_m = K(\overline{\mathbf{x}}_m, \mathbf{x})$, for m $= 1, 2..., M$. On comparing the Eq. 2.1 and Eq. 2.2, one can clearly observe the reduced computation burden for the inversion of covariance matrix, from a matrix $\mathbf{K}_N$ with $N \times N$ entries to a matrix $\mathbf{K}_M$ with $M \times M$ entries. Following the derivation in [62], the predictive distribution given a new input $\mathbf{x}_*$ is:

$$p(y|\mathbf{x}_*, \mathbf{D}, \overline{\mathbf{X}}) = \mathcal{N}(y_*|\mu_*, \sigma_*^2), \tag{2.3}$$

where

$$\mu_* = \mathbf{k}_*{}^T \mathbf{Q}_M^{-1} \mathbf{K}_{MN} (\mathbf{\Lambda} + \sigma^2 \mathbf{I}) \mathbf{y} \tag{2.4}$$

$$\sigma_*^2 = K_{*,*} - \mathbf{k}_*^\top (\mathbf{K}_M^{-1} - \mathbf{Q}_M^{-1}) \mathbf{k}_* + \sigma^2. \tag{2.5}$$

The derivation of $\mathbf{Q}_M$ is omitted here, however, the main cost in computing it is the inversion of a diagonal matrix [62]. Using the spatial data as input, μ_* will represent the mean predicted field and the variance σ_*^2 will constitute to the uncertainity in this prediction. Moreover, the scalar environmental fields can be non stationary [4] and up to a certain extent, SPGP is capable of modeling non-stationary GP processes through its pseudo-inputs, which gives it an edge over other sparse GP methods.

2.4 Informed Sampling and Monitoring Frameworks

Sampling of water hotspots is an important task in many environmental monitoring activities, providing a means for us to understand the various processes

taking place in our water resources. Environmental fields such as temperature or chlorophyll content, are scalar quantities which exhibit spatio-temporal variations [22]. Moreover, to achieve the objective of sampling the hotspots, prior knowledge of the field's spatial and temporal variations is necessary. Such prior information regarding the variability is generally not available but by using some recent adaptive frameworks [48, 15, 4, 57, 54], it is possible to obtain a good approximation of the environmental field. We define such frameworks as monitoring or exploration-only frameworks.

In general monitoring frameworks impose a temporal constraint on the field estimation task. These temporal constraints are used to exploit the quasi-static characteristic of the environmental fields. In simple terms, the fields are assumed to be temporally constant for short duration of time, spanning anywhere between minutes to several hours [22, 24]. For the sample collection task, the monitoring frameworks can be used to obtain an approximation of the field, which can be followed by sampling from the hotspots of the approximated field. Such an approach will require the monitoring task to be completed before the sampling of hotspots can begin. This sequential order of tasks will introduce a temporal gap between the field estimation and the physical sample collection and ideally this gap should be as low as possible.

The framework in [69] suggests an on-line multi-choice hiring algorithm. This framework presents an algorithm for making irrevocable sample selection decisions from a set of possible sampling candidates. Following this work, the algorithm in [19] uses a formulation of secretary-hiring problem [70] and probabilistic model for field estimation. In both of these frameworks, the field estimation and sample collection occur in an on-line manner but only for predefined paths, such as lawn mower or yo-yo motion. Such an approach, which predefines the paths, reduces the adaptive nature in the task of obtaining a good approximation of the field, and introduces a bias in the sampling task.

Recently, [20] suggested another framework for simultaneous monitoring and sampling. In this framework, the sampling and monitoring tasks are distributed

between the two members of a team, consisting of an explorer robot and a sampler robot. The sampling is based on a look-back secretary-hiring formulation, which controls the sampler robot's motion. The formulation of this framework is suitable for the scenarios where a robot can either be configured as an explorer or a sampler, within a team consisting of two or more robots. However, our objective is to develop a single robot framework for both sampling and monitoring tasks without the need for prior information about the environmental field. Such a single robot framework can then be easily extended to a multi-robot system, where the sampling and monitoring roles can be shared across members of the multi-robot team.

2.5 System Identification

The environmental monitoring task has motivated significant technological advancements in the field of Autonomous Underwater Vehicles (AUVs) [71, 72, 73, 74, 75, 1, 3]. Generally, the dynamics of such AUVs are described using six degrees of freedom and their respective differential equations of motion [76]. These equations have parameters that represent the nonlinear components of AUV dynamics. However, in practice, it is quite difficult to estimate these parameters and therefore, certain assumptions are made to linearize this model for easier estimation [77]. These linearized models are actually the simplified equations of inter-dependent state variables, which are computed using regression analysis of AUV motion data. However, due to linearization, these models usually fail to predict the dynamics accurately during complex manoeuvres. Moreover, AUV designs which have different propulsion mechanisms or have significantly different body structure as compared to industrial AUVs like Iver, Gavia and Remus cannot use the linear physics model defined in [76, 77] and a large proportion of soft robots are developed for environmental monitoring face this challenge [78, 79, 80, 81, 2]. Therefore, for such systems, the linearized model needs to be estimated again and tuned to compensate for the change in dynamics. One way of modelling these systems is to use a linear representation of the AUV's

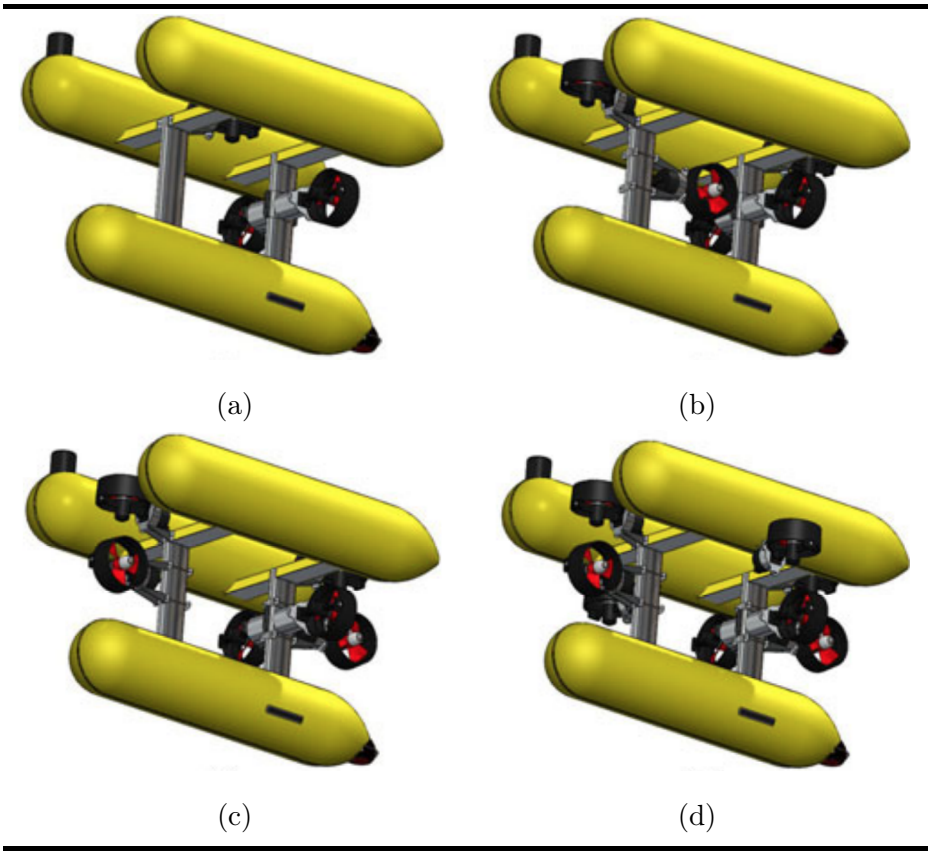
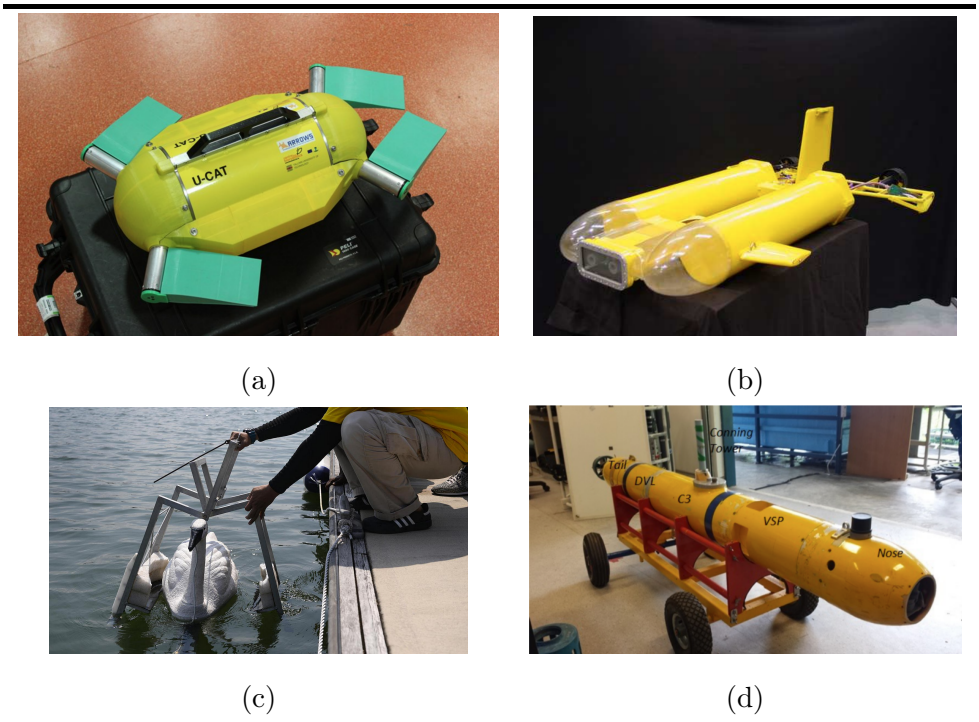


Figure 2.1: Girona 500 [1] in different thruster configurations: (a) 3 thrusters and 3 Degrees of Freedom (DOF)(b) 5 thrusters and 5 DOF, (c) 6 thrusters and 5 DOF and (d) 8 thrusters and 6 DOF.

state variables and control inputs, also known as output equation of a state-space model. In comparison to the traditional approach, this is an easier method for estimating unknown dynamics as it utilizes the motion data directly without requiring detailed analysis of the AUV's physics model.

Other than the physics-based approach and the state-space model, some other approximation-based techniques have also been used in the past for learning the dynamics model [82, 83, 84, 85]. In particular, the feedforward neural networks have the capability of approximating any continuous function [86] and therefore, they are good candidates for such applications. Inspired by this, a linearly parameterized neural network was used to estimate dynamics of a surface vessel [87]. However, the neural network presented in [87] has no hidden neurons and



(a) (b)

(c) (d)

Figure 2.2: AUV with new propulsion mechanisms and designs: (a) U-CAT [2] robot by the Centre for Biorobotics, Tallinn University of Technology, Estonia, (b) a hybdrid AUV, Starbug [3], by CSIRO, Australia, (c) NUSwan and (d) STARFISH AUV with a vector thruster by Acoustic Research Laboratory, National University of Singapore, Singapore.

its input features are directly connected to its output layer. It is shown in [88] that if a complex mapping exists between the input and output units, a large hidden layer is required in between to estimate the mapping perfectly. Therefore, in order to develop an efficient dynamics model and predict complex manoeuvres accurately, a Multi-Layer Perceptron (MLP) type of neural network appears to be a promising solution.

An MLP is a fully-connected feedforward neural network used for function approximation. In an MLP, most of its neurons have a nonlinear activation function, the standard choices of which are signum, sigmoidal or hyperbolic tangent. Reference [89], along with [90], showed that the use of rectifiers as activation functions in different neural networks improved their discriminative performance. Following this approach, in [43] the performance of different

activation functions was compared, and the results demonstrated that rectifier neurons were better at finding minima during training for classification tasks, as well as for contextual analysis. These promising results led to recent advancements in the area of Deep Learning. In [91], a rectifier neural network coupled with a quadratic function of the helicopter's state variables, accurately predicted its acceleration during complex manoeuvres.

Chapter 3

Adaptive Monitoring using a Single Robot

In Chapter 2, we discussed the Informative Path Planning frameworks used for environmental monitoring and listed some of the single-robot frameworks. These frameworks generally have a budget for planning such as fixed mission time or the number samples a robot can collect. Moreover, these frameworks commonly estimate the environmental field using GP, which has a training complexity of $\mathcal{O}(N^3)$ where N is the number of total training points. In this chapter, we discuss our single-robot framework which uses the Sparse GPs for estimating the fields and makes use of decision making techniques for efficient path planning.

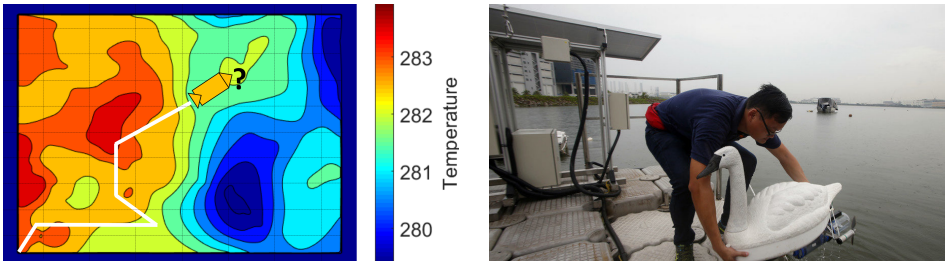
3.1 Estimating Scalar Fields using a Robot

Monitoring environmental processes is critical in understanding our water resources. It has helped scientists examine algal blooms and phyisco-chemical interactions at a micro scale and most of these examinations have been done using static sensors [45]. However, the water resources are generally spread over large areas and thus static bouys are only capable of providing a sparse representation of the environmental process. This has motivated the development of actuated sensors or robots [21], which can be used to provide both spatial and temporal resolution of an environmental field. In a typical deployment scenario, the robots are given the task to survey an area and provide an estimate of the environmental field. These robots mostly make use of the existing technologies present in static bouys and their use can be maximized by coordinating their motion to provide an estimate of the environmental field. The use of IPP (IPP)

frameworks have provided a new approach to solve this monitoring task problem and in this section, we discuss some of the problems and applications of the existing IPP frameworks.

One of the challenges in estimating environmental fields is the data collection process. It is a challenge because the robots generally have limited time to collect data before the field varies significantly. Therefore, this data collection process needs to be carefully coordinated to make efficient use of mission time. In general, the IPP mitigates this challenge by selecting paths that have high probability of providing a good estimate of the field [15, 16, 17]. This selection of paths in IPP is generally based on an information criterion, which is the main difference between IPP and other planning frameworks. The robot then traverses the path which provides maximum information as per the criterion and collects data to give an estimate of the environment. Generally, the IPP frameworks have three components: collecting data while traversing, adapting the robot's path to collect informative data and learning a model of the environmental field. The first component is self explanatory, whereas, the last two components are the key characteristics which define the behaviour of all IPP frameworks.

Our aim is to obtain a good estimate of scalar environmental fields, such as chlorophyll concentration, conductivity or temperature. This can be achieved by collecting representative data of the spatio-temporal field using robots. In general, the temporal changes in environment occur over a period of days or months [92, 93, 94]. Therefore, it is safe to assume that these scalar fields will be temporally static for a period of few minutes. Such an assumption is useful for reducing the modeling complexity of the environmental field. However, this assumption also imposes a temporal constraint on the data collection process. Therefore, imposing this constraint on the mission time for monitoring tasks is important. Some of the IPP algorithms that impose constraints on mission time are discussed in [50, 15, 57]. However, all these frameworks estimate the environmental fields using full Gaussian Process (GPs), which suffers from longer training times and thus affects the overall planning performance. A framework



(a) (b)

Figure 3.1: (a) A scenario similar to the transect sampling task presented in [4]. The environmental field here is the sea surface temperature of an area in Sea of Japan on January 21, 2018. Taken from MUR SST dataset [5]. (b) An autonomous surface robot, NUSwan, capable of water quality monitoring is being retrieved after mission.

that uses Sparse GPs for estimating fields is discussed in [52]. This framework presents a solution for persistent monitoring where it handles the temporal dynamics by using the user set thresholds. In this chapter, we introduce two frameworks that provide the flexibility of sparseness as well as the ability to perform monitoring tasks in limited mission time and benchmark its performance against lawn mower paths.

In our first framework, the planning and learning components of our IPP framework are decoupled. We denote this framework as *AdaPP* here on-wards. This frameworks learns the model of the environmental fields using GP and uses the model variance to plan for future locations. The information exchange in this framework between the learned model and the planning step only occurs at the start of each planning step. Such a decoupled approach is simple and provides a framework to make use of sparse GPs for field estimation as well as finishing the monitoring task within a fixed amount of time. Moreover, such an approach also means the planning component of the IPP framework is entirely based on the

information provided only once and does not make full use of the information gained using the GP model. We introduce another online IPP framework named as *k-AdaPP*, which integrates the GP's kernel information directly into the planning step. This framework uses a similar GP method as *AdaPP* to get an estimate of the field and the corresponding variance in this estimate. In comparison to the first framework, *k-AdaPP* uses the updated model information as well as the remaining mission time for generating next waypoints. We test our performance with other conventional methods for estimating environmental field using a sea surface temperature dataset provided by NASA Jet Propulsion Laboratory [5]. The results show that the predicted fields using *k-AdaPP* and *AdaPP* are a good approximation of the ground truth and outperforms all the conventional methods.

3.2 Problem Formulation

Broadly, our problem statement is to find a path for a robot and collect representative data to provide a good estimate of the environmental field and finish this task within a fixed amount of time T. This statement can be represented as

$$\underset{\tilde{\mathbf{P}}\in\tilde{\Lambda}_t}{\arg\min}\frac{1}{|\tilde{\mathcal{X}}|}\int_{\tilde{\mathcal{X}}}\left(\mathcal{Y}(\mathbf{x})-\hat{\mathcal{Y}}(\mathbf{x},\mathbf{D}_t\cup\mathbf{D}_{T-t}^{\tilde{\mathbf{P}}})\right)^2 d\mathbf{x}, \tag{3.1}$$

such that

$$\mathcal{T}(\tilde{\mathbf{P}})\leq T \text{ and} \tag{3.2}$$

$$\tilde{\mathbf{P}}_0=\mathbf{x}_t, \tag{3.3}$$

where $\mathcal{Y}(\cdot)$ is a function of the field over the spatial domain $\tilde{\mathcal{X}}$, $\hat{\mathcal{Y}}(\cdot,\cdot)$ is the estimated function of the field at time t using the collected data D_t and the data yet to be collected $D_{T-t}^{\tilde{\mathbf{P}}}$ by traversing a path $\tilde{\mathbf{P}}$. Moreover, $\tilde{\Lambda}_t$ in (3.1) represents a set of paths originating from $\mathbf{x}_t$ and $\tilde{\mathbf{P}}$ denotes one of the element of Λ_t with $\tilde{\mathbf{P}}_0$ as the starting point of these paths. All the paths in the set Λ start from the robot's current location $\mathbf{x}_t$, given by (3.3). Finally, the function $\mathcal{T}(\cdot)$ provides

an estimate of the time to traverse a path. In our problem statement, we have defined the measure of goodness as a low mean squared error over the complete spatial domain. The current form of the problem statement is not solvable as we cannot get the information about $\mathcal{Y}(\cdot)$ without sampling or visiting locations and thus without actually traversing a path $\tilde{\mathbf{P}}$, we cannot obtain the target values $\mathbf{y} = \{y_i\}_{i=t}^{T}$ for yet to be visited locations. Interestingly, we can make use of characteristics of a GP model to make the problem (3.3) solvable. The function $\hat{\mathcal{Y}}(\cdot,\cdot)$ is learned using a GP model and it can be written as $\mathcal{N}(\mu_*, \sigma_*^2)$, where μ_* should represent a close approximation of $\mathcal{Y}(\cdot)$ if the learned GP model is a good fit and the overall variance σ_*^2 is low. Therefore, we can re-write (3.1) as

$$\underset{\tilde{\mathbf{P}}\in\tilde{\Lambda}_t}{\arg\min} \frac{1}{|\tilde{\mathcal{X}}|} \int_{\tilde{\mathcal{X}}} \sigma_*^2(\mathbf{x}, \mathbf{D}_t, \tilde{\mathbf{P}}) d\mathbf{x}. \tag{3.4}$$

It is important to note that we have replaced $\mathbf{D}_{T-t}^{\tilde{\mathbf{P}}}$ with just $\tilde{\mathbf{P}}$ as we can get an estimate of the variance without sensing the target values and only the locations x given by $\tilde{\mathbf{P}}$ is sufficient. However, the estimated variance depends on $\mathbf{D}_{T-t}$ and it will be updated using (2.5) whenever the robot collects more data $\mathbf{D}_t$. Therefore, our planning problem can be seen as collecting good data such that the overall variance becomes is low.

The problem statement given by (3.4) is in continuous domain $\hat{\mathcal{X}}$. This means the number of paths in the set $\tilde{\Lambda}_t$ will be large and searching for the optimal path $\tilde{\mathbf{P}}^*$ that satisfies our problem statement will be difficult. A common approach to reduce such complexity is to discretize the continuous domain $\tilde{\mathcal{X}}$ into a grid $\mathcal{X}$. In this scenario, each location $\mathbf{x}_t$ will generally have 8 neighbours and thus for each location the decision will be to select which of these neighbours to visit. Using this discretization approach, our problem can be represented as

$$\underset{\mathbf{P}\in\Lambda_t}{\arg\min} \frac{1}{|\mathcal{X}|} \sum_{\mathcal{X}} \sigma_*^2(\mathbf{x}, \mathbf{D}_t, \mathbf{P}), \tag{3.5}$$

where Λ_t represents the set of paths generated by connecting spatial locations and $\mathbf{P}$ represents one of the path in this set. The constraints for these paths $\mathbf{P}$ given

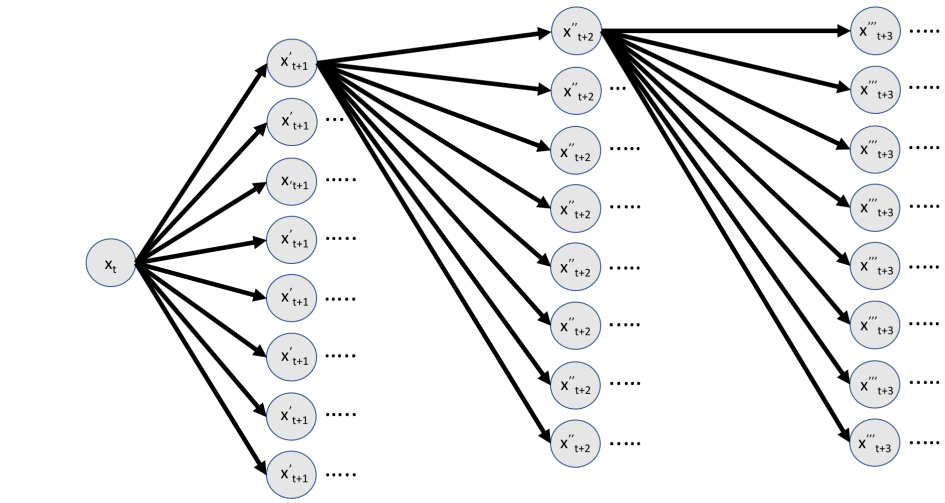


Figure 3.2: An illustration of Λ_t as a state space. The first state is the location of the robot $\mathbf{x}_t$ and time t. The next time step $t+1$ shows the 8 neighbours $\mathbf{x}'_{t+1}$ of this first state. This is followed by further branching of the states, however, we have branched only one of the state to show the growing state space for our search problem.

by (3.3) and (3.2) are still applicable with only change of shifting from continous domain to a discrete domain. In simple terms, the set Λ_t will represent a state space with the first state as $\mathbf{s}_{\mathbf{x}_t} = \{\mathbf{x}_t, t\}$ and each subsequent state connected to all its neighbours given by $\mathcal{E}(\mathbf{s}_{\mathbf{x}_t})$, where $\mathcal{E}(\cdot)$ is a function that provides the spatial locations of all the neighbours for the next time step. In the worst case scenario, each of the state will have a branching factor of maximum possible value of function $\mathcal{E}(\cdot)$, which is 8. This will result in a large state space as shown in Fig. 3.2 and finding an optimal solution within a short period of time will not be feasible. In literature, such a search problem that cannot be solved in real-time using the limited computation power are termed as **NP**-hard problem. In our framework, we solve this search problem by efficiently sampling the state space and coordinating the robot motion over the limited mission time.

3.3 Adaptive Path Planning using a GP model

In this section we explain our approach for solving the search problem as explained in Section 3.2. We make use of concepts such as spatial decomposition and Dynamic Programming to efficiently search through the state space.

3.3.1 Decomposition of Survey Area

A general approach to reduce the size of the search problem is to decompose the survey area into smaller sub-regions. This decomposition is commonly defined by a criteria, which provides a good representation of the field. One such approach is *GreedySubset* [47], where each sub-region is formed using some prior information. However, most of the survey tasks do not have prior information of the field and thus such methods will not be applicable.

Our approach for spatial decomposition is both simple and computationally fast. We use the estimated variance in the predicted field for decomposing the survey area into smaller regions. Our survey area is discretized into C cells, given by $\mathbf{c} = \{c_1, c_2 ... c_C\}$. The representative point ${}^{c_b}\mathbf{x}$ and variance σ^2_{*,c_b} of a cell c_b containing Z spatial points are given as:

$$\sigma^2_{*,c_b} = \frac{\sum_{z=1}^{Z} \sigma^2_{*,\,{}^{z}\mathbf{x}}}{Z}, \tag{3.6}$$

$${}^{c_b}\mathbf{x} = \frac{\sum_{z=1}^{Z} {}^{z}\mathbf{x}\sigma^2_{*,\,{}^{z}\mathbf{x}}}{Z}, \tag{3.7}$$

where $\sigma^2_{*,\,{}^{i}\mathbf{x}}$ is the variance of a spatial point ${}^{i}\mathbf{x}$. These quantities represent the mean variance and variance weighted average for all locations in a cell c_b. As (3.6) and (3.7) are are easy to compute, the survey area can be decomposed into representative sub-regions at a faster rate.

3.3.2 Planning Framework using Dynamic Programming

One of the problems in our path planning is the temporal constraint on surveying an environmental field. This problem becomes more challenging as no prior information is given to facilitate the path planning. Our framework mitigates this problem by re-planning after each step and selecting the next action based on the information gained from the collected data $\mathbf{D}$ and the remaining mission time. This selection aims at informed data collection within the limited mission time to satisfy (3.4). Therefore, this step of obtaining the next action is critical to make our planning framework efficient. We facilitate the explanation of our framework by introducing a simple single robot algorithm with no constraints on mission time first.

Single-robot algorithm with no constraints

In a scenario, where the environmental field does not vary temporally and thus there are no constraints on mission time, the task of selecting next actions is simple. Given the robot's current location ${}^{c}\mathbf{x}_t$ at time t and spatially decomposed survey area, the next action can be obtained using policy iteration:

$$V({}^{c}\mathbf{x}_t) := \max_{a_t \in \mathbf{A}_{{}^{c}\mathbf{x}_t}} [R({}^{c}\mathbf{x}_t, a_t) + \gamma V({}^{c'}\mathbf{x}_{t+1})], \tag{3.8}$$

$$\pi({}^{c}\mathbf{x}_t) := \arg\max_{a_t \in \mathbf{A}_{{}^{c}\mathbf{x}_t}} V({}^{c}\mathbf{x}_t), \tag{3.9}$$

where

$$R({}^{c}\mathbf{x}_t, a_t) = \frac{\sigma^2_{*,c'}}{||{}^{c}\mathbf{x}_t - {}^{c'}\mathbf{x}_{t+1}||},$$

${}^{c'}\mathbf{x}_{t+1}$ is the next cell on taking the action a_t in a cell ${}^{c}\mathbf{x}_t$, $\mathbf{A}_{{}^{c}\mathbf{x}_t}$ is a set of all possible actions in cell ${}^{c}\mathbf{x}_t$ and γ is the discounting factor. In this scenario we can use Dynamic Programming (DP) to obtain the optimal policy $\pi^*(\cdot)$ and the next action using $\pi^*({}^{c}\mathbf{x}_t)$. This is a simple and straightforward approach to get optimal solutions for temporally static fields. However, an important to note is that the diagonal movements are penalized in the reward function $R({}^{c}\mathbf{x}_t, a)$.

This is to optimize the usage of time, and does not put constraints on the overall mission time. We term this simple algorithm as single-robot DP.

Information Collection with Constraints on Mission Time

As mentioned before, the single-robot DP algorithm works only for scenarios where there is no limitation on mission time. However, we have to impose a constraint on the mission time to collect sufficient information from the environmental fields before it varies significantly. The actions provided by the single-robot DP algorithm may not be optimal after imposing temporal constraints. This is because the actions for finite horizon problems are generally different from the actions in infinite horizon problems as infinite horizon decisions do not factor a temporal bound. Therefore, the direct usage of single-robot DP algorithm will not meet our planning objective of surveying an area in limited time.

Introducing time constraint to the single-robot DP algorithm is not straightforward. The selection of next action in temporally constrained planning frameworks has to be based on the information already collected as well as on the remaining time. In our framework, the selection process is aimed at reducing the overall model variance to provide a good approximation of the environmental field. This selection process starts with the updated variance map and we make use of spatial decomposition as explained in Section 3.3.1 to reduce the variance map into a set of cells. Therefore, the optimal action a_t^* at time t is given as

$$a_t^* = \underset{a_t \in \mathbf{A}_{c_{\mathbf{x}_t}}}{\arg\max} [\, \mathcal{U}(a_t) + \eta \vartheta_{T-t}(a_t)], \tag{3.10}$$

where $\mathcal{U}(a_t)$ is a function that gives the mean variance of the cell that will be visited on taking action a_t, η is a discounting factor, $\mathbf{A}_{c_{\mathbf{x}_t}}$ is a set of all the possible actions for a robot in cell c and $\vartheta_{T-t}(a_t)$ represents the potential of reducing variance within the remaining time $T-t$ on taking the action a_t.

In an ideal scenario, we should plan till the end of mission time for all the possible actions and select the action at current time t that reduces the maximum variance. Such an approach would result in a large tree search problem with each

Algorithm 3.1: Single-robot Adaptive Path Planning Framework

Data: Starting point (${}^{c}\mathbf{x}_t$), Total mission time (T), SPGP Parameters (M)
Result: Predicted Field μ_*
`/* Initialization */`
Run sampling decomposition for grid size G and assume constant variance σ_*^2
`/* Algorithm Loop */`
while $t < T$ **do**
  Construct the set $\mathbf{A}_{{}^{c}\mathbf{x}_t}$
  for $a_{t+1} \in \mathbf{A}_{{}^{c}\mathbf{x}_t}$ **do**
    Estimate $\vartheta_{T-t}(a_t)$ by simulating planning using *single-robot DP* and GP model
    Calculate and store $\mathcal{U}(a_t) + \eta\vartheta_{T-t}(a_t)$
  end
  Use 3.10 to get a_t^*
  Take the action a_t^* and collect training data $\mathbf{D}_t$
  Update the time $t := t + 1$
  θ = Full-GP using $[\mathbf{y}_t(1:M), \mathbf{X}_t(1:M)]$
  $\overline{\mathbf{X}} = \mathbf{X}_t(rand(M))$
  $[\mu_*, \sigma_*^2]$:= Run SPGP($\mathbf{y}_t$, $\mathbf{X}_t$, θ, $\overline{\mathbf{X}}$)
  Run sampling decomposition for Grid Size G and σ_*^2
end

node represented by a tuple of the robot's future location and the remaining time $T - t$, where the branching factor for each node is defined by set of available actions $\mathbf{A}_{{}^{c}\mathbf{x}_t}$. This search problem is similar to (3.5), however, this search is over the spatially decomposed grid instead of spatial points. A common solution to such a search problem is to use value function approximation and learn a model for $\vartheta_{T-t}(\cdot)$, however, doing this in an online manner with less data points may not deliver good results. Moreover, we are only interested in an estimate of $\vartheta_{T-t}(\cdot)$ to obtain the next action and not in the absolute value. In our framework, we compute an estimate of $\vartheta_{T-t}(\cdot)$ using a combination of single-robot DP algorithm

and kernel information to quickly traverse through the large search space. At each planning iteration, we simulate the actions given by the set $\mathbf{A}_{c_{\mathbf{x}_t}}$ parallely. These parallel simulation start with finding the neighboring cell ${}^{c'}\mathbf{x}_t$ that the robot will visit for each of the available actions. Simulating any of the robot's actions has two challenges, one is updating the variance when a robot moves from one cell to another and the second challenge is selecting an action from the set of actions available for the next cell c'_{t+1}. We have two approaches for obtaining an estimate of the change in variance, one is arbitarily reducing the variance to the noise term σ^2 and second is to use kernel information to estimate the updated variance. The first approach is relatively simple and faster to compute, however, it does not provide a good estimate of change in variance as compared to the second approach. We term our first approach as *AdaPP* framework.

Using the kernel information to estimate the change in variance is a more direct approach for integrating the model information into planning. It helps in providing a more realistic estimate of the posterior variance, which is completely based on the data collected till time t. We term this framework that makes use of kernel information as *k-AdaPP*. These two approaches solve only our first challenge, however, these approaches do not provide an answer to selecting an action for the new cell c'_{t+1}. We can use the single-robot DP algorithm to find the optimal move for the new cell, however, this move will disregard the mission time constraint on the planning. As we are interested in obtaining an estimate of $\vartheta_{T-t}(\cdot)$, we still select the move suggested by the single-robot DP algorithm but reduce the mission time by the time that it will take for the robot to move from the cell c'_{t+1} to the cell c''_{t+2}. We repeat these two steps of estimating the change in variance and using single-robot DP algorithm to find moves for the new cell, and simulate the paths for the remaining mission time.

After finishing the simulation for one of the action, we estimate the variance map and calculate the area under the curve (AUC). This AUC is a representation of the estimated remaining variance in the model at the end of mission time for taking an action at time t. This is also illustrated in the Fig. 3.3. We perform

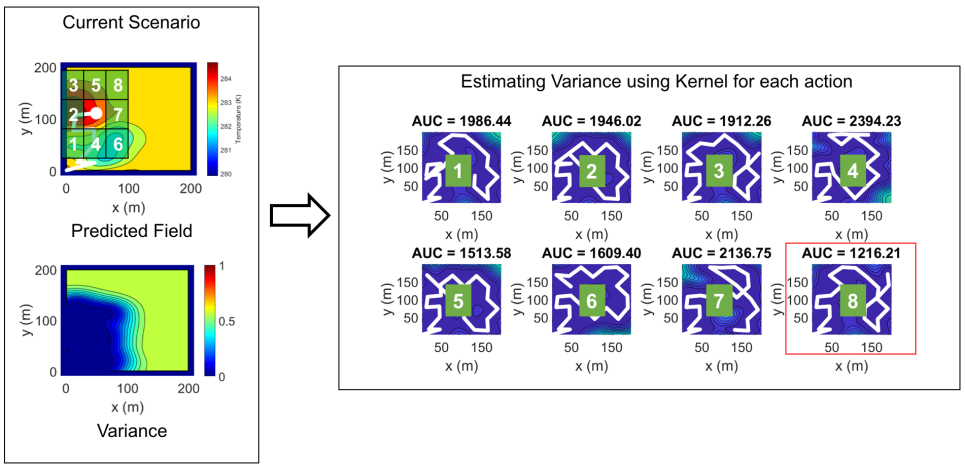


Figure 3.3: An illustration of the planning step of k-AdaPP. The images in the left box are the predicted field and the corresponding variance, which is estimated using the data collected by the robot. In the current scenario, the robot can move to 8 neighbouring cells and thus the planning framework simulates 8 different paths from each of these cells. All these candidate paths are present in the box on the right. Each of the candidate path have an associated Area Under the Curve (AUC) value, which represents the remaining variance at the end of mission. The framework selects the path with minimum AUC, highlighted by a red box.

this procedure for all the actions given by the set $\mathbf{A}_{c_{\mathbf{x}_t}}$ and simulate the paths to get an estimate of resulting variances. We calculate the AUCs from the estimated resulting variances and use it to provide an estimate of $\vartheta_{T-t}(\cdot)$ at the current time t for each action a. Once the estimates of all the actions at current time are available, the next action is obtained using (3.10). We have also summarized these steps in Algorithm 3.1. Overall, our framework uses two approaches for reducing the posterior model variance and coordinates the robot's motion to collect sufficient data for providing a good estimate of the field.

3.3.3 SPGP for Feild Estimation

The kernel function in the SPGP formulation is used in estimating both the mean and variance of the GP model, as explained in Section 2.3.2. Moreover,

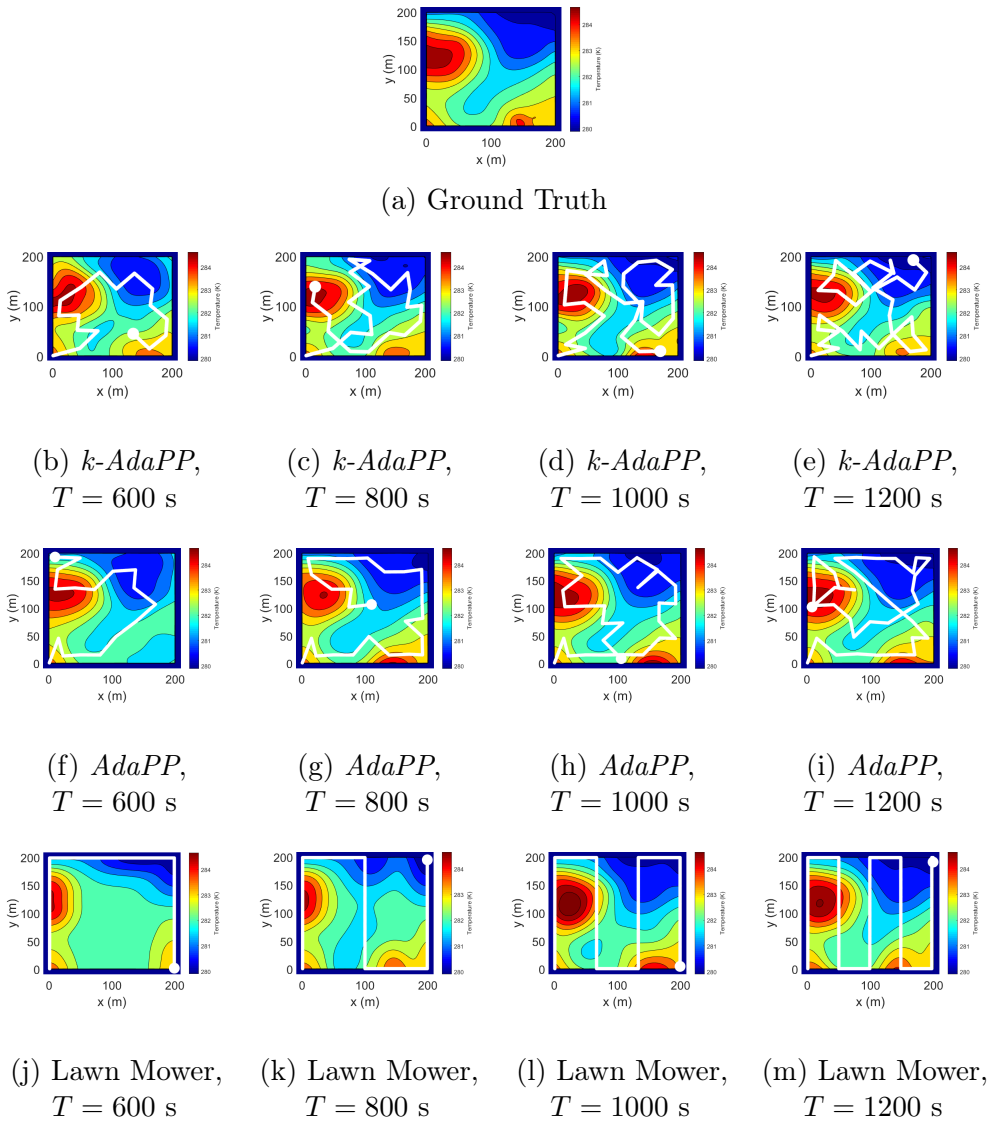
our planning iteration depends on the kernel function to determine the change in variance and thus plays an important role in the performance of our framework. We are interested in modeling envrionmental fields and a commonly used kernel function for such geostatistic modelling is *automatic relevance determination*, which is defined by $K(\cdot,\cdot)$:

$$K(\mathbf{x}_n, \mathbf{x}_{n'}) = \alpha \exp\left(\frac{1}{2}\sum_{l=1}^{2} b_l (x_{n,l} - x_{n',l})^2\right), \tag{3.11}$$

where α, b_1 and b_2 are the parameters of the kernel function, $\mathbf{x}_n$ and $\mathbf{x}_{n'}$ represent two different locations and $x_{n,l}$ represents the value for l dimension of $\mathbf{x}_n$. After including the noise term σ^2, the hyperparameters of the sparse GP are given by $\theta = \{\alpha, b_1, b_2, \sigma^2\}$ and pseudo-inputs $\overline{\mathbf{X}}$. These hyperparameters are learned by maximizing the marginal likelihood as mentioned in [62]. Once these hyperparameters are known, the variance for the locations not visited by the robot is estimated using (3.11). Moreover, this same equation is used to estimate the change in variance and get an estimate of $\vartheta_{T-t}(\cdot)$, which is critical for taking the decision given by (3.10).

3.4 Simulation Results

In this section, we do a comprehensive analysis of our framework using simulations. We use the Sea Surface Temperature (SST) data of the Sea of Japan provided by the Jet Propulsion Laboratory [5] for simulating environmental field. We extracted the temperature data for an area of 200 $\times$ 200 km^2 and mapped it to an area of 200 $\times$ 200 m^2. This mapping was done to conserve the features of an environmental field and have an area that can be explored within a realistic value of mission time, T. The Fig. 3.4.a shows a visual representation of this field. We implemented our algorithm in MATLAB. For implementation of SPGP, we used the code provided by the authors of [62] and modified it for our spatial regression application. The simulations were done on a hexa-core Intel i7 processor with 32 GB of RAM.



(a) Ground Truth

(b) *k-AdaPP*, $T = 600$ s

(c) *k-AdaPP*, $T = 800$ s

(d) *k-AdaPP*, $T = 1000$ s

(e) *k-AdaPP*, $T = 1200$ s

(f) *AdaPP*, $T = 600$ s

(g) *AdaPP*, $T = 800$ s

(h) *AdaPP*, $T = 1000$ s

(i) *AdaPP*, $T = 1200$ s

(j) Lawn Mower, $T = 600$ s

(k) Lawn Mower, $T = 800$ s

(l) Lawn Mower, $T = 1000$ s

(m) Lawn Mower, $T = 1200$ s

Figure 3.4: Simulation results for a 200×200 m^2 sea surface temperature field. (a) is the ground truth obtained using the MUR dataset [5]. (b)-(e) represents the field estimated using *AdaPP* for different $T = \{600 \text{ s}, 800 \text{ s}, 1000 \text{ s}, 1200 \text{ s}\}$ respectively. Similarly, (f)-(i) are the field estimated using *AdaPP* and (j)-(k) represent the fields estimated using lawn mower paths. It can be observed that the fields estimated using our frameworks *k-AdaPP* and *AdaPP* are visually more similar to the ground truth as compared to the fields estimated using lawn mower patterns. Moreover, the performance for all the approaches improve with increasing the mission time T. These performance become comparable for $T = 1200$ s as the robot would have collected sufficient information using any of the simulated approaches.

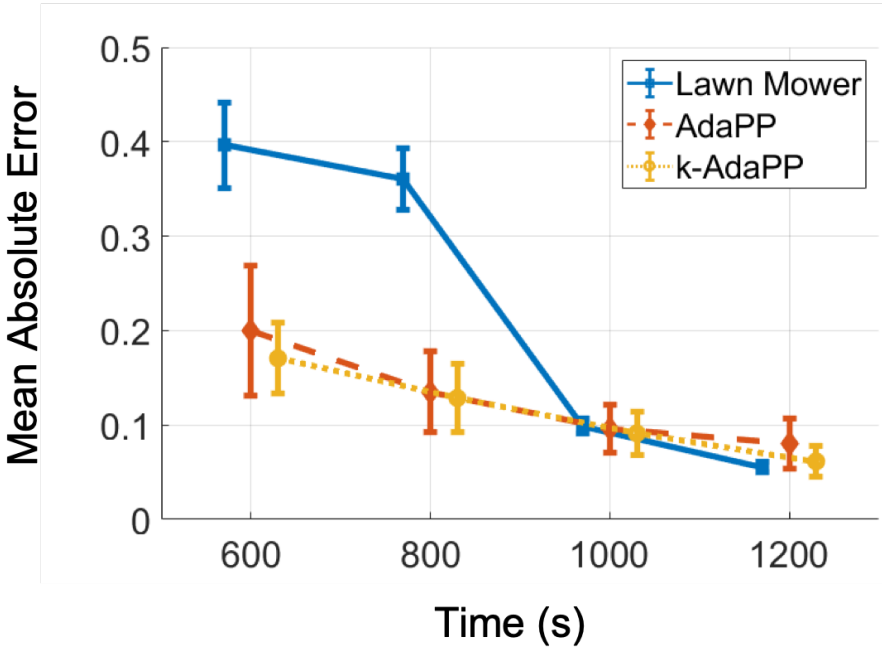


Figure 3.5: A performance comparison between Lawn Mower paths, k-AdaPP and AdaPP using Mean Absolute Error (MAE) in estimating the SST field for different mission times over 50 runs. The MAE in estimating the temperature fields using *k-AdaPP* for low mission time $\{600\text{ s}, 800\text{ s}\}$ is less than the MAE for *AdaPP* and Lawn Mower paths. This is a good indication that integrating more information from the GP model improves the overall planning performance.

3.4.1 Performance Comparison with Lawn Mower Paths

We used the field as shown in Fig. 3.4(a) and simulated the monitoring task for four different mission times, $T = \{600\text{ s}, 800\text{ s}, 1000\text{ s}, 1200\text{ s}\}$ and the grid size G as 30 m. The vehicle is assumed to be traveling at constant speed of 1 m/s. We also estimated the field using the lawn mower paths for these four static periods, assuming constant vehicle speed and zero turning radius. The total number of pseudo-input points were kept at $M = 50$ and the initial guess for SPGP kernel parameters were obtained by running a full GP regression on a subset of M points.

The results of our simulations are shown in Fig. 3.4 and the Mean Absolute Error (MAE) over 50 runs in estimating the field is shown in Fig. 3.5. It is clear

from the results that the field predicted using *AdaPP* and *k-AdaPP* algorithm are a better match of the ground truth compared to the field predicted using lawn mower paths, specially for shorter mission periods. Moreover, the approximation improves with increasing mission time T for all the approaches. This is an expected trend as our frameworks and the lawn mower paths cover more area and get more time to collect data, which improve the GP model's performance.

3.4.2 Performance Improvement by Integrating Kernel Information

We compared our two frameworks, *AdaPP* and *k-AdaPP*, to examine the improvement in performance by integrating model information into the planning. We calculated the MAE for each mission time to check whether our frameworks are a good solution to our original problem statement given by (3.1). These results are present in Fig. 3.5. The *k-AdaPP* framework performs significantly better for smaller mission times such as $T = \{600s, 800s\}$, whereas, the performance improvement for longer mission times is not significant.

The significant improvement in the *k-AdaPP*'s performance over the decoupled framework *AdaPP* for smaller mission times is encouraging. This reflects that *k-AdaPP*'s decision making process is improved by using the kernel information in path planning. However, the the performance for longer mission times is similar and it is expected as both the frameworks get sufficient time to collect substantial information about the field and thus yield similar results.

3.4.3 Performance of Spatial Decomposition Approach

We first simulated to show the importance of weighted decomposition algorithm as described in Section 3.3.1. We simulated 10 independent runs of *k-AdaPP* framework using weighted and non-weighted decomposition. The total time T was kept as 800 seconds for each of these simulations. These runs were done for three grid sizes, $\{20\text{ m}, 30\text{ m}, 40\text{ m}\}$, and we used RMSE to do a comparison between the two approaches. The result for this comparison are shown in Fig. 3.6.

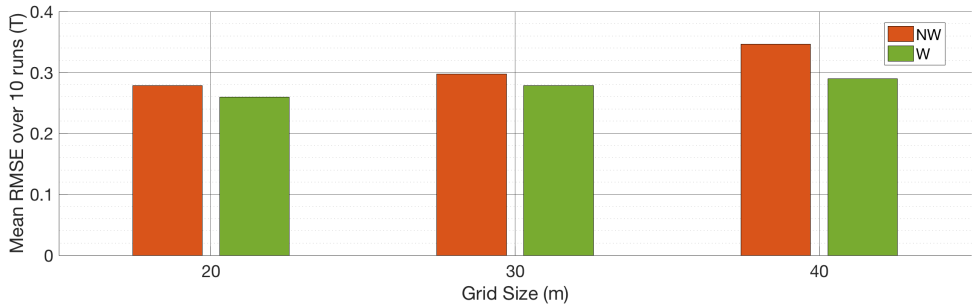


Figure 3.6: A comparison between weighted (W) and non-weighted (NW) spatial decomposition using the mean RMSE values over 10 runs in estimating the SST field for different monitoring strategies. The performance using the weighted (NW) spatial decomposition given by green bars is better than the performance using non-weighted (W) spatial decomposition for all the grid size G. The mission time for all the simulations was kept as $T = 800$ s.

It can be observed that the average performance with weighted decomposition is always better than non-weighted decomposition. It is also important to note that as the grid size increases, the improvement in performance of the weighted decomposition becomes more significant. Ideally, we would like to have the grid size which will give us lowest RMSE value but this will also result in increasing the state space for decision making. Based on our compute resources, we selected the grid size of 30 m for all the remaining simulations.

3.4.4 Performance Comparison with Other Estimation Approaches

In addition to lawn mower patterns, some of the commonly used approaches for collecting data are random walk and purely greedy techniques. We compared the performance of our framework against these conventional approaches as well. Simulating a random walk is straightforward, however, we modified our decision strategy to simulate a pure greedy approach. At each decision step, the greedy algorithm selected the neighbouring cell with maximum variance and this was repeated till the end of mission time.

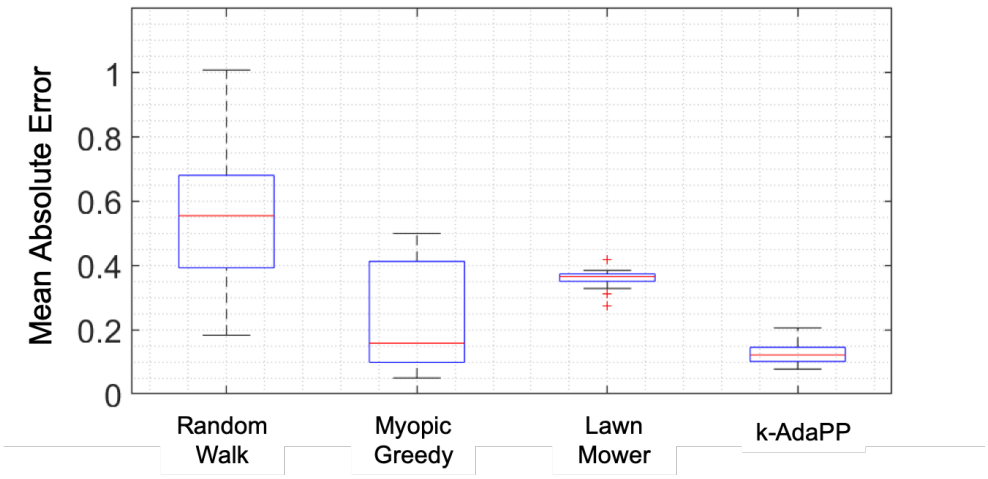


Figure 3.7: Mean Absolute Errors (MAEs) over 10 runs in estimating the SST field for different monitoring approaches. The error values for the fields estimated using Random and Lawn Mower are significantly higher as compared to other methods. The Greedy algorithm has the lowest mean MAE value among all the benchmark models, however, our framework *k-AdaPP* even outperforms this benchmark model. Our framework's better performance as compared to the Greedy algorithm is an indication that coordinating the robots based on remaining mission time is important in providing a good estimate of the field.

We simulated 50 runs for each approach and calculated the Mean Absolute Error (RMSE) for each run. The mission time for all the runs was kept constant as $T = 800$ s. The result for MAE are present in Fig. 3.7, where it can be clearly observed that our framework outperforms all the other approaches. Interestingly, the mean performance of greedy algorithm is better than lawn mower because the greedy behaviour is always targeted towards myopic information gain. This is a positive indication of using information dynamically instead of using no information for path planning. However, our framework optimizes on information as well as mission time in comparison to the myopic greedy approach, and this is one of the reason why our method outperforms the pure greedy approach.

3.5 Summary

In this chapter, we discussed two frameworks for estimating the scalar environmental field using a single robot with the constraints on mission time. Our two frameworks integrated the model information into the planning iterations using different approached. Our *AdaPP* framework updates the cell of a variance to the measurement noise to simulate the planning step. However, the *k-AdaPP* uses the kernel map to update the variance map based on the path and thus provide a more accurate estimate of change in variance. Both of our frameworks used a sparse GP method for estimating the environmental fields.

The simulations were performed using sea surface temperature data and the results show that our frameworks provide a better approximation of the environmental field compared to traditional lawn mower paths and other monitoring method. We also compared the performance between our *k-AdaPP* framework with other commonly used data gathering approaches and the results show that our framework outperforms all the other simulated methods. These results provide a good evidence for using kernel information in path planning. The two main contributions discussed in this chapter were learning a scalar environmental field using a sparse GP model and a single-robot adaptive framework based on the GP's kernel function.

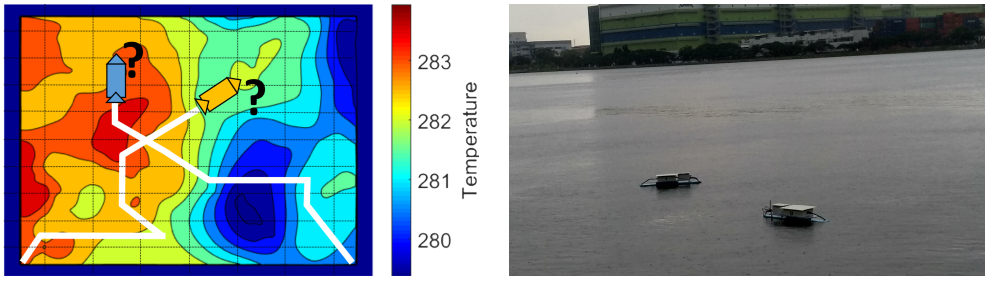
Chapter 4

Multi-Robot Adaptive Monitoring

Using a team of robots for environmental monitoring of large water bodies is an emerging approach. The idea behind this approach is to further reduce the mission time for surveying environmental as compared to the mission time required for single robot. However, the team of robots should be coordinated to make efficient use of the mission time and provide a good approximation of the environmental field. In this chapter, we suggest an online multi-robot framework *m-AdaPP* to handle this coordination. We test our framework for estimating an environmental field with no prior information. We also benchmark the performance in field against conventional approaches such as lawn mower patterns. Although we focus on monitoring application, this approach is general and can be used for exploring unknown environments.

4.1 Practical Constraints in Using a Team of Robots

Our objective is to obtain a good approximation of a scalar environmental field, such as temperature, conductivity or chlorophyll concentrations, using a team of robots within a fixed amount of time. We discussed some approaches of performing environmental monitoring in Chapter 3 using a single robot. A common problem in using single robot is the limitation on the area it can cover within a fixed amount of time. This also means that a single robot can only collect a fixed amount of information within a fixed amount of time and thus reducing the amount of time will also reduce the total collected information. Such problems with single robot scenarios can be easily resolved by using a team



(a) (b)

Figure 4.1: (a) A multi-robot scenario for estimating Sea Surface Temperature. (b) Our robots deployed in a local reservoir to perform adaptive monitoring to estimate Dissolved Oxygen in water.

of robots. More robots can be used to collect more information, however, these robots should be coordinated to collect the information efficiently. In this chapter, we discuss a framework that can be used to provide coordination for a team of robots.

An entropy based method for multi-robot operation [4] was proposed where a set of waypoints were obtained using dynamic programming. However, this framework was for transect environmental fields, where robots can only move forward and generate waypoints using the information from previous locations. Another multi-robot framework is [95], which uses lawn mowers to obtain preliminary information followed by a leader robot making decisions to adapt the lawn mower pattern for the team of robots. Such an approach is helpful for adapting lawn mower patterns, however, following these straight vertical paths will consume time in collecting repetitive information. A similar approach is described in [96], where the robots maintain a formation and adapt the formation to cover a larger area. Vehicles with motion constraints such as gliders can make use of these frameworks but most of the robots do not have such strict motion constraints.

We are interested in a multi-robot framework that can be used for a team of robots such as AUVs that have less motion constraints and finish the monitoring task within a fixed amount of time. Moreover, an important component missing in the multi-robot frameworks is the computation time for making decisions. The computation time can be ignored in cases where it is insignificant compared to the overall mission time. However, our task is to finish collecting data within a short amount of time and thus computation time will become an important component of our overall mission time. For example, lets assume each decision iteration takes about 5 seconds and during a mission of 600 seconds, the decisions are taken about 20 times. In this example, the computation time will consume more than 15% of the mission time and thus leaving even less time for data collection.

In this chapter, we suggest a multi-robot IPP framework *m-AdaPP* for estimating a scalar environmental field. Our aim is to coordinate a team of robots to get a good approximation of a scalar field and finish the overall mission in a fixed amount of time. We make use of the SPGP model to provide an estimate of the field and the corresponding variance. The paths are evaluated to minimize the overall variance and we have included the time taken for this evaluation in our overall mission time. We test the coordination and field estimation performance of our framework using a sea surface temperature dataset in simulations. We also compare our framework's performance against the conventional lawn mowers patters for estimating the environmental fields and show that our framework performs well.

4.2 Problem Formulation

Our problem statement is to coordinate a team of H robots to provide a good estimate of an scalar environmental field within time T. This problem statement is similar to the problem statement of the framework in the previous chapter, however, the data collection is being done by a team of robots instead of a single robot. We can re-write (3.5) for this case as

$$\arg\min_{\widehat{\mathbf{P}}\in\widehat{\Lambda_t}} \frac{1}{|\mathcal{X}|}\sum_{\mathcal{X}} \sigma_*^2(\mathbf{x}, \widehat{\mathbf{D}_t}, \widehat{\mathbf{P}}), \tag{4.1}$$

where $\widehat{\mathbf{P}}$ is a set containing one path for each robot and given as

$$\widehat{\mathbf{P}} = \{{}^1\mathbf{P},\ {}^2\mathbf{P},\ {}^3\mathbf{P}, \ldots\ {}^H\mathbf{P}\}. \tag{4.2}$$

Similarly, $\widehat{\mathbf{\Lambda}}$ is a set containing all the paths for each robot and it is given as

$$\widehat{\Lambda}_t = \{{}^1\Lambda_t,\ {}^2\Lambda_t,\ {}^3\Lambda_t, \ldots\ {}^H\Lambda_t\}. \tag{4.3}$$

In (4.2) and (4.3), the set of path ${}^i\mathbf{P}$ and the set of collection of paths ${}^i\Lambda_t$ represent the candidate paths for robot i. Moreover, we have replaced $\mathbf{D}_t$ with $\widehat{\mathbf{D}_t}$ in (4.1) to include the data collected from all the robots and thus $\widehat{\mathbf{D}_t} = \{{}^1\mathbf{D}_t, {}^2\mathbf{D}_t, {}^3\mathbf{D}_t \ldots\ {}^H\mathbf{D}_t\}$, where ${}^i\mathbf{D}_t$ is the data collected by robot i till time t. The mission time constraints to solve (4.1) will be

$$\mathcal{T}({}^i\mathbf{P}) + \tau \leq T - t \text{ and} \tag{4.4}$$

$${}^i\mathbf{P}_0 = {}_i\mathbf{x}_t\ \forall i \in [1, H], \tag{4.5}$$

where the new addition τ in comparison to (3.2) represents the computation time for each decision and ${}_i\mathbf{x}_t$ represents the location of robot i at time t. The constraints given by (4.4) represent that each robot will have less than $T - t$ time available for collecting data. However, we can absorb τ inside $\mathcal{T}({}^i\mathbf{P})$ if the computation can be done while traversing. This will require taking a decision for the next location while collecting data. The current formulation given by (4.1) will not allow it as the decision made at time t is possible only after collecting all the data $\widehat{\mathbf{D}_t}$ till time t. However, we can use the data $\widehat{\mathbf{D}_t}$ to make a decision for the next location $\mathbf{x}_{t+1}$ and collect more data while travelling from $\mathbf{x}_t$ to $\mathbf{x}_{t+1}$. This will change the problem statement to

$$\underset{\widehat{\mathbf{P}} \in \widehat{\Lambda_{t+1}}}{\arg\min} \frac{1}{|\mathcal{X}|} \sum_{\mathcal{X}} \sigma_*^2(\mathbf{x}, \widehat{\mathbf{D}_t}, \widehat{\mathbf{P}}), \tag{4.6}$$

such that

$$\tau \leq \mathcal{T}(\{{}^i\mathbf{x}_t, {}^i\mathbf{x}_{t+1}\}) \text{ and} \tag{4.7}$$

$$\mathcal{T}({}^i\mathbf{P}) \leq T - t \text{ and} \tag{4.8}$$

$${}^i\mathbf{P}_0 = {}_i\mathbf{x}_{t+1} \ \forall i \in [1, H] \ , \tag{4.9}$$

where $\widehat{\Lambda_{t+1}}$ represents the set of all paths for each robot i from its next location ${}^i\mathbf{x}_{t+1}$. This formulation changes (4.4) to (4.8) but introduces a new constraint given by (4.7), which represents that computation time should be less than or equal to the time taken by the robots to travel to the next location. Similar to our state space representation in (3.2), the set $\widehat{\Lambda_{t+1}}$ can be visualized as a state space too. This state space will be a convolution of multiple state spaces given by $\{{}^i\Lambda_{t+1}\}$ and its starting state given as $\mathbf{s}_t = \{{}_i\mathbf{x}_{t+1}\} \ \forall i \in [1, H]$. The state space of the set $\widehat{\Lambda_{t+1}}$ will be large and thus it can be classified as a **NP**-hard search problem. Therefore, we need a framework to efficiently sample this state space and provide a good estimate of the environmental field.

4.3 Multi-Robot Planning Framework

We suggest an algorithm, named as *m-AdaPP*, to efficiently search through the state space given by $\widehat{\Lambda_{t+1}}$ and collect data using the kernel information to get a good estimate of our field. This algorithm follows the basic IPP framework and thus has the three components, which are planning, model learning and collecting data. As discussed in the section before, we learn the model and plan for the next location while the robots are travelling and collecting more data. We make use of the Spatial Decomposition approach explained in the previous chapter and reduce our search space by discretizing the grid in to cells.

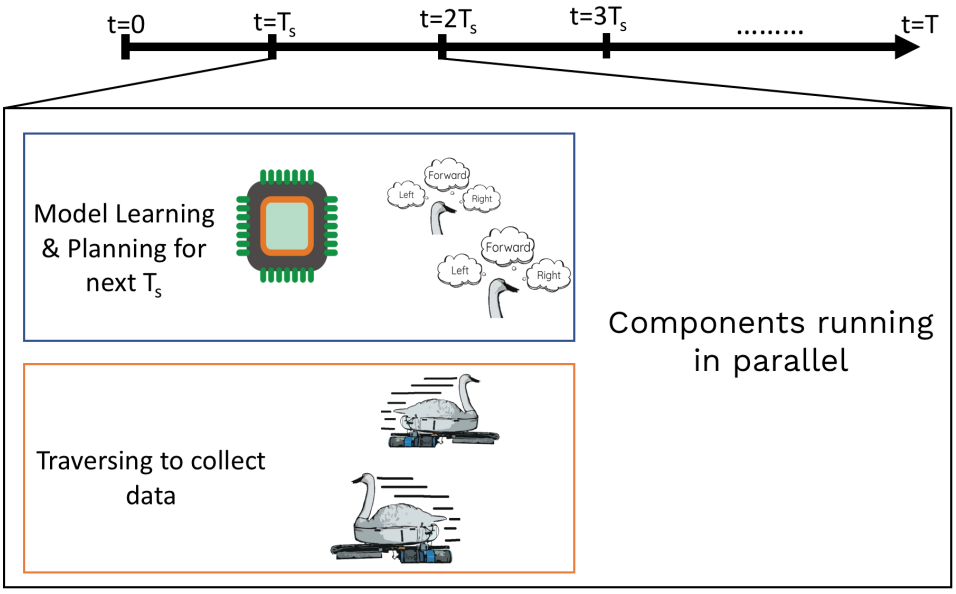


Figure 4.2: An illustration showing execution of one step of our framework. The GP model learning and planning for next waypoint occurs in parallel while the robot is collecting data from the field. Such an approach can be used for efficient use of mission time.

4.3.1 Multi-Robot Path Planning

There are three constraints on our planning as explained in the previous section. These are the limits on each robot's total mission time T, bounds on the computation time used for planning and each robot's starting point. Although the planning is done over cells instead of locations, this does not mean that robots do not collect data while traveling from one cell to another. The data is collected as and when the sensors provides a scalar value of the field, defined by the sensor's frequency rate. This data is then stamped against the current location of the robot and used for estimating the overall environmental field.

In the discretized are, the representative location of each cell will change based on the variance in that cell. This will result in each robot traversing different lengths of paths and therefore, it will mean robots will reach its next

waypoint at different time. Therefore, the update of the collected data will be asynchronous and planning decisions will be made using partial information. We bring snychronization between the team of robots by dividing the total time T into intervals of T_s, where at the end of every interval the robots are assumed to have reached their waypoints, and the future waypoints are generated while traversing to these point. This time interval T_s should be sufficient for a robot to reach the neighboring cells even when traversing at the average speed.

Multi-Robot Path Planning with no constraints

We make use of *single-robot DP* algorithm to explain our multi-robot path planning algorithm. Extending the single-robot algorithm to a multi-robot scenario requires two modifications. First, the robots should be coordinated to explore an area in a collaborative manner. This is similar to the problem which is solved using a general sequential algorithm in [47]. This sequential algorithm decides a path for one robot first, which is followed by path allocation to second robot and then sequentially to the remaining robots. However, we are concerned only with the next waypoint instead of the complete path. The second necessary modification is to prevent the collision between two robots, which can be achieved by having negative rewards for each robot's current location. As we are interested in planning for one-time ahead, the update rules for multi-robot case can be given as

$$V({}^{c'}_{i}\mathbf{x}) = \max_{a \in \mathbf{A}_{{}^{c'}_{i}\mathbf{x}}} [\mathcal{R}({}^{c'}_{i}\mathbf{x}, a) + \gamma V({}^{c''}_{i}\mathbf{x})], \tag{4.10}$$

$$\pi({}^{c'}_{i}\mathbf{x}) = \arg\max_{a \in \mathbf{A}_{{}^{c'}_{i}\mathbf{x}}} V({}^{c'}_{i}\mathbf{x}), \tag{4.11}$$

where

$$\mathcal{R}({}^{c'}_{i}\mathbf{x}, a) = \begin{cases} \frac{\sigma_{*,c'}}{||{}^{c'}_{i}\mathbf{x} - {}^{c''}_{i}\mathbf{x}||} & \text{if } {}^{c''}_{i}\mathbf{x} \notin {}_{1:H}\mathbf{x} - \{{}^{c'}_{i}\mathbf{x}\} \\ -\epsilon & \text{if } {}^{c''}_{i}\mathbf{x} \in {}_{1:H}\mathbf{x} - \{{}^{c'}_{i}\mathbf{x}\} \end{cases},$$

${}_{1:H}\mathbf{x}$ represents the current location of all the robots, ${}_{i}^{c'}\mathbf{x}$ represents the location the robot i will reach at time $t+1$ and ϵ is the value of the negative reward. We run one full cycle of policy iteration using DP for robot 1 and obtain the optimal policy given by $\pi^*(\cdot)$. Using this policy, we get the future location of the robot 1, given by ${}_{1}^{c''}\mathbf{x} := \pi^*({}_{1}^{c'}\mathbf{x})$, and update this new location for robot 1 in the location set of all robots ${}_{1:H}\mathbf{x}$. This update of the location in the set of location ${}_{1:H}\mathbf{x}$ makes sure that robot 2 and remaining robots do not visit the same cell where the robot 1 will be at the next time step. We run such cycles sequentially for all the H robots and obtain the respective next waypoints. We name this algorithm the *multi-robot DP*.

Multi-Robot Path Planning with Temporal Constraints

Introducing time constraints to this multi-robot framework is not straightforward. The new waypoints generated using the above framework may not be optimal given the remaining time $T-t$. Therefore, we need to find a combination of actions for different robots that would reduce the overall variance within the remainging time. Let Φ_{t+T_s} denote this combinatorial set of all actions $A_{{}_{i}^{c'}\mathbf{x}}$ $\forall i \in [1, H]$ at time $t+T_s$. We define another combinatorial set φ_{t+T_s}, which is a subset of Φ_{t+T_s} representing one action for each robot. From the set Φ_{t+T_s}, we remove the states where the next action for two or more robots will result in a collision. Therefore, the optimal combination of action $\varphi^*_{t+T_s}$ at time $t+T_s$ can be given by

$$\underset{\varphi'_{t+1}\in\Phi_{t+1}}{\arg\max}\ \mathcal{U}(\varphi'_{t+T_s}) + \eta\vartheta_{T-t-T_s}(\varphi'_{t+T_s}), \tag{4.12}$$

where $\mathcal{U}(\varphi'_{t+T_s})$ is a function that gives the sum of variances of cells that will be visited due to the combination of actions in φ'_{t+T_s}, η is a discounting factor and $\vartheta_{T-t-T_s}(\varphi'_{t+T_s})$ represents the potential of reducing uncertainty within the remaining time $T-t-T_s$ by taking the combination of actions given by the set φ'_{t+T_s}.

The problem given by 4.12 is of the similar form as the single robot planning problem given by 3.10. We can use the kernel based planning as explained in

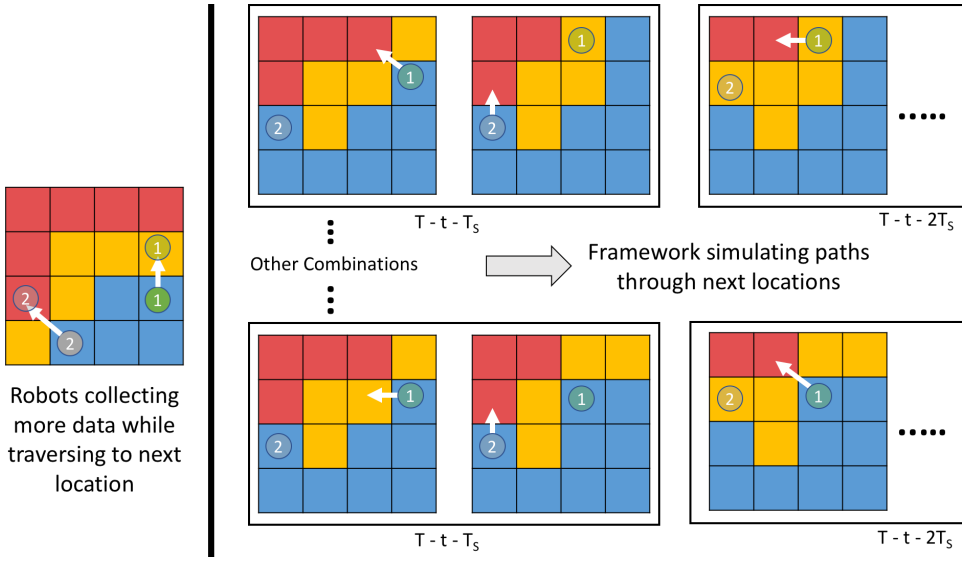


Figure 4.3: A concept diagram explaining the sequential planning in our multi-robot framework *m-AdaPP*. The left-most grid shows the robots traversing a path to the next waypoint. In parallel, the framework is planning for the next of actions assuming the robots have already reached the location. The framework simulates paths and updates the variance for the remaining time and select the actions that minimize the overall variance.

the previous chapter to find an estimate of $\vartheta_{T-(t+T_s)}(\cdot)$ and solve this problem. However, the problem here is for a team of robots rather than a single robot and thus the decision is over a set of actions φ'_{t+T_s} for multiple robots as compared to set of actions for a single robot. This problem requires coordinated movement of each robot to minimize the overall variance. Therefore, direct use of the planning framework for the single robot is not feasible in this case.

We solve the problem of coordination between the robots by using *multi-robot DP* at each simulated planning iteration and provide a combination of actions. We do this in two steps. First, we run one full iteration of *multi-robot DP* and obtain a set of actions φ'_{t+T_s}. Second, we reduce the total time by T_s and update the variance of the cells based on the paths the robots will take due to the actions given by φ'_{t+T_s}. We re-run the *multi-robot DP* algorithm to find the next set

Algorithm 4.1: *m-AdaPP* - multi-robot Adaptive Path Planning

Data: Starting points (${}_{1:H}\mathbf{x}$), Total mission time ($T - t - T_s$), SPGP Parameters (M)
Result: Estimate Field μ_*
`/* Initialization */`
Run sampling decomposition using resolution G and set σ_*^2 as constant
`/* Algorithm Loop */`
while $t < T$ **do**
  Construct the set Φ_{t+T_S}
  foreach $\varphi'_{t+T_s} \in \Theta_t$ **do**
    Estimate $\vartheta^{\varepsilon}_{T-t-T_s}(\varphi'_{t+T_s})$ by simulating planning using *multi-robot DP* and SPGP Kernel
    Calculate and store $\mathcal{U}(\varphi'_{t+T_s}) + \eta\vartheta_{T-t-T_s}(\varphi'_{t+T_s})$
  end
  Use Eq. 4.12 to get φ'_{t+T_s}
  Wait for time interval given by T_s to be over
  Take the actions given in $\mathbf{s}^*_{\mathbf{t}}$
  Update the time $t = t + T_s$ and training data $\mathcal{D}_{t+T_s}$
  θ = Full-GP using $[\mathbf{y}(1:M), \mathcal{X}(1:M)]$
  $\overline{\mathcal{X}} = \mathcal{X}(rand(M))$
  $[\mu_*, \sigma_*^2]$:= Run SPGP($\mathbf{y}$, $\mathcal{X}$, θ, $\overline{\mathcal{X}}$)
  Run sampling decomposition using resolution G and σ_*^2
end

of actions φ'_{t+2T_s} using the updated variance. We iterate over these two steps till the mission time is over $t = T$. Using this approach, we get an estimate of $\vartheta_{T-(t+T_s)}(\cdot)$ and thus we can evaluate the value of the combination φ'_{t+T_s} given by (4.12). Similarly, we can use this to find the values for all the combinations given by the set Φ_{t+T_s}. Once we have the values for all the actions, we can use (4.12) to find the set of actions for the robots for time $t + T_s$. An example of these steps is illustrated as a diagram in Fig. 4.3. All these steps are repeated whenever the training dataset $\mathbf{D}$ is updated, which will be at a regular interval of T_s and thus bring the adaptive nature to the *m-AdaPP* framework. Our overall

framework is presented in Algorithm 4.1.

There are two important points to note about our framework. First, the decisions are made sequentially but it does not mean the robots also move sequentially. Once a decision is made, all the robots move to its next location simultaneously within time T_s. Second, the calculation of $\vartheta_{T-(t+T_s)}(\cdot)$ for one set of action in Φ_{t+T_s} is independent of the other set of actions. This provides an opportunity to estimate the value for $\vartheta_{T-(t+T_s)}(\cdot)$ for all set of actions in parallel. This helps in reducing the overall computation time of our framework.

4.3.2 Field Prediction Using SPGP

We make use of the same kernel function used in single-robot frameworks. It is defined by $K(\cdot,\cdot)$:

$$K(\mathbf{x}_n, \mathbf{x}_{n'}) = \alpha \exp\left(\frac{1}{2}\sum_{l=1}^{2} b_l (x_{n,l} - x_{n',l})^2\right), \tag{4.13}$$

where α, b_1 and b_2 are the parameters of the kernel function, $\mathbf{x}_n$ and $\mathbf{x}_{n'}$ represent two different locations and $x_{n,l}$ represents the value for l dimension of $\mathbf{x}_n$. After adding the Gaussian noise model, the hyperparameters of the sparse GP are given by $\theta = \{\alpha, b_1, b_2, \sigma^2\}$ and pseudo inputs $\overline{\mathbf{X}}$.

4.4 Experimental Results

We performed two set of experiments using our framework. One was to examine the coordination within the team of robots and other was to compare the performance of our framework with the performance of conventional lawn mower patterns. This section discusses both of these experiments.

4.4.1 Simulations to Test the Coordination Efficiency

We used the real field data of Sea Surface Temperature (SST) provided by the Jet Propulsion Laboratory [5]. We extracted the data for two areas of 200×200 km^2 each and mapped it to two fields with area 200×200 m^2. This scaling was

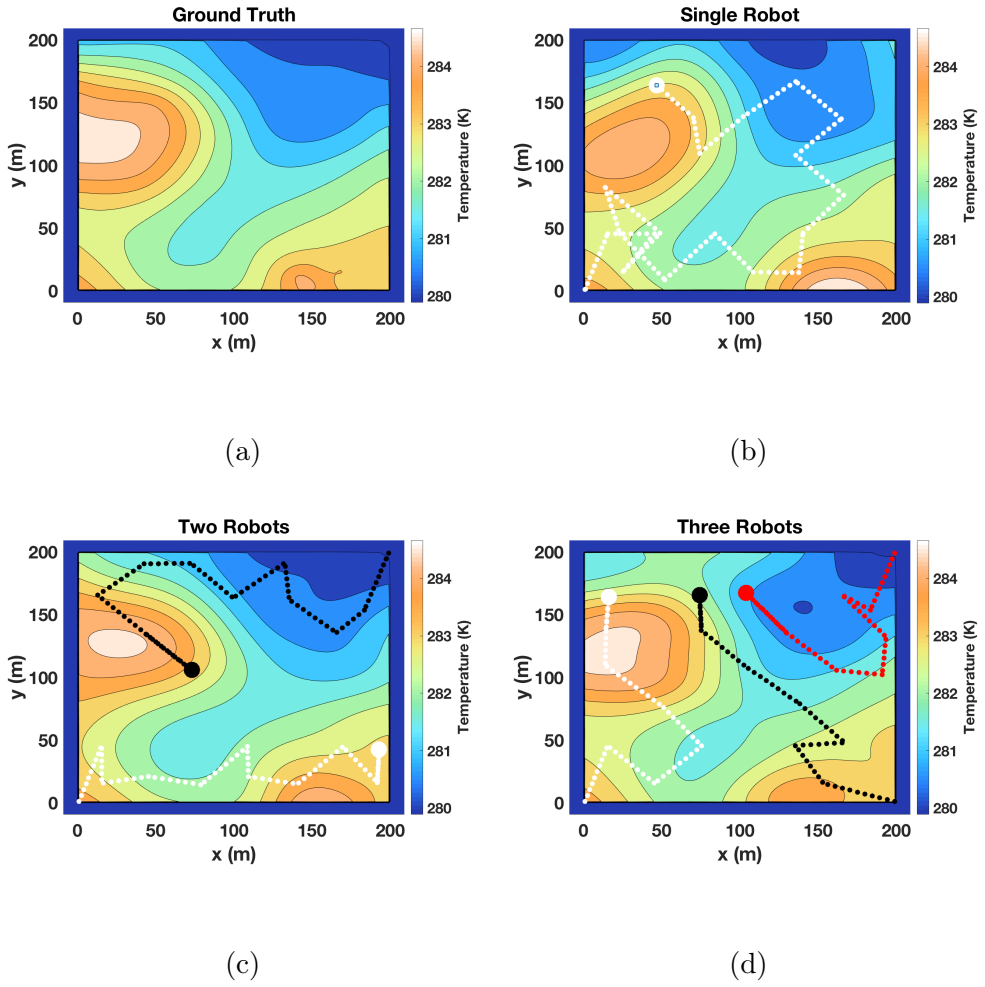


Figure 4.4: Simulation results of *m-AdaPP* for estimating a temperature field. (a) represents the ground truth and (b) represents the field estimated using one robot. Similarly, (c) and (d) represent the field estimated using two robots and a team of three robots, respectively. The mission time for (b) is $T = 2400$ s, (c) is $T = 1200$ s and (d) is $T = 800$. It can be observed that the hot and cold regions estimated using different team of robots is correct. This shows that our framework efficiently coordinates the team of robots and makes efficient use of mission time to collect good representative data.

done to retain the features of a temperature field and have an area which can be explored within a practical value of mission time T. We denote Fig. 4.4(a) as Field 1 and Fig. 4.5(a) as Field 2 for the following discussions.

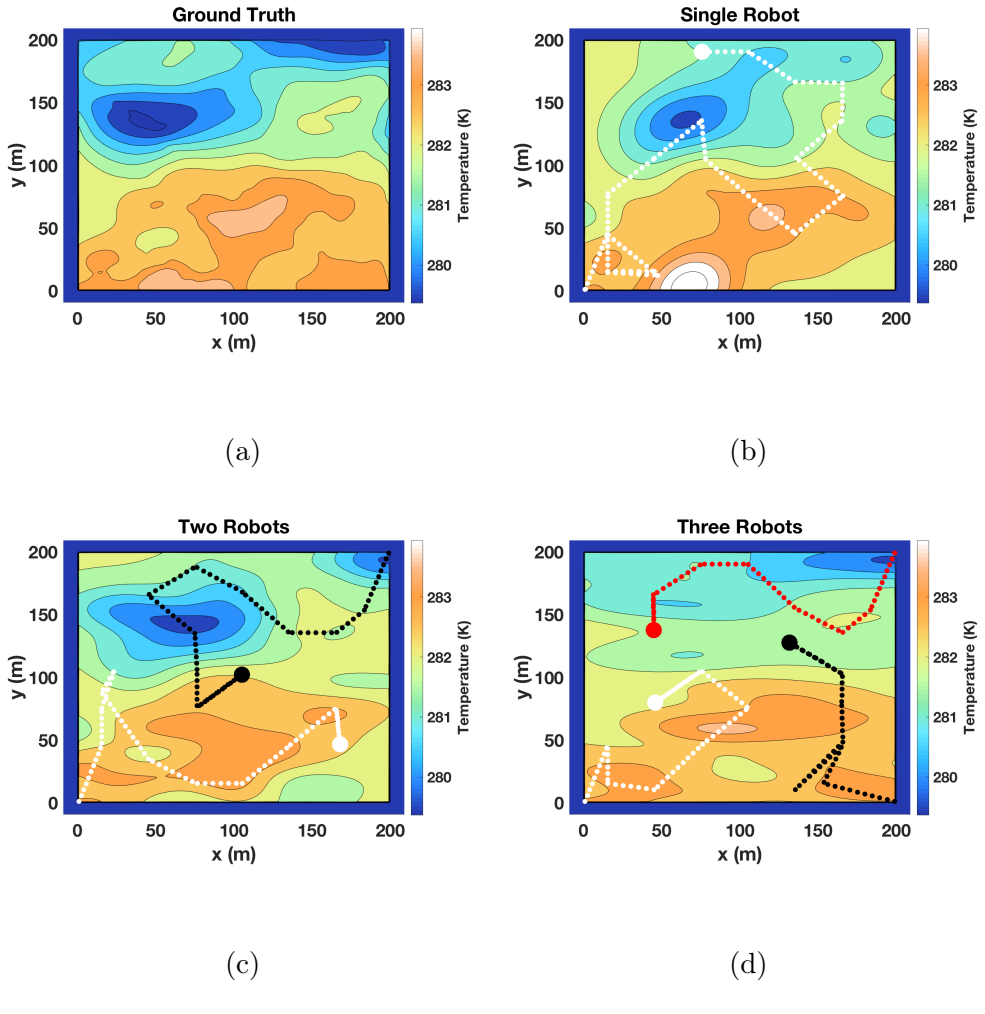


Figure 4.5: Another set of results of our framework on estimating a temperature field from the SST dataset. (a) represents the ground truth and the remaining figures (b)-(d) represent the field estimated using one, two and three robots, respectively. Similar to the previous figure, the field estimated using different team of robots is comparable. These results are another example showing that our framework coordinated the team of robots well.

The maximum speed of the robots used in environmental monitoring is generally low. This is to make sure that the robots do not cause substantial disturbance to the environment it is sensing. For example, the maximum speed of our water quality sensing robot, NUSwan [6], is 1 m/s. However, the average speed of NUSwan with external disturbances such as strong winds or waves is

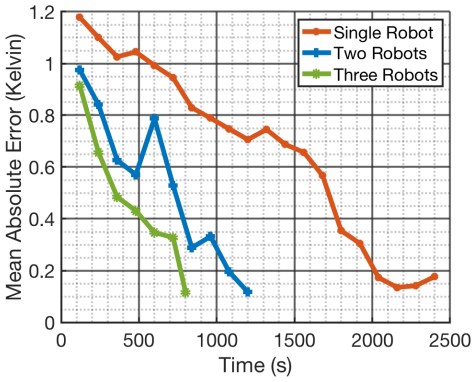
about 0.3 m/s. We use this speed to define the value of T_s. Following the grid size $G = 30$ m from the previous chapter, the average time required for travelling from one cell to another cell will be atleast 100 s. Therefore, we set the value of T_s as 120 giving the vehicle sufficient time to reach the next cell.

We learned the SPGP model with $M = 50$ pseudo data points. Similar to the single robot framework, we used a subset of M points of the total dataset and run a full GP regression to initialize the hyperparameters of our kernel function. We implemented our algorithm in MATLAB. For SPGP, we took the MATLAB code provided by the authors [62] and modified it for spatial regression application. The simulations were done on a hexa-core Intel i7 processor with 32 GB of RAM.

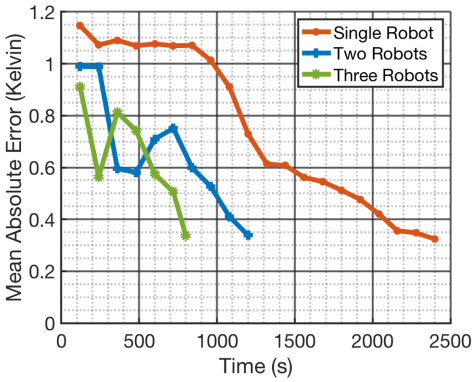
We simulated teams consisting of a maximum of three robots. We examined the coordination within the team of robots by providing less mission time for the teams with more number of robots. This means that the team of two robots will have less time compared to a single robot. If the framework is able to coordinate this team of two robots well, the performance of these two simulations should be comparable. For our simulation setup, we set the mission time T as 2400 s for single robot, 1200 s for a team of two robots and 800 s for a team of three robots.

It is important to note that the mission time T for single robot here is 2400, which is much higher than the mission times set in the previous chapter. This difference is due to assumed speed and a direct comparison of the mission times is not correct. Instead a relationship can be seen in terms of distance travelled, a vehicle with speed 0.3 m/s will travel around 700 m in 2400 s. Whereas, the same vehicle with speed increase to 1 m/s will travel the same distance in 700 s. Therefore, our limit on mission time in the current setup is not substantially different from the setup in previous chapter. Moreover, our average computation time for the team of three robots after parallelization was about 23 s, which is much less than T_s and thus satisfies the constraint on τ given by (4.7).

The results of the fields estimated using *m-AdaPP* are present in Fig. 4.4 and Fig. 4.5. It is clear from the figures that the estimated hot and cold regions by our framework are correct and the overall estimated fields are similar for



(a)



(b)

Figure 4.6: Mean Absolute Error (MAE) in estimating the temperature field using different teams of robots. (a) shows the error in estimating the field given by Field 1 and (b) shows the error in estimating Field 2. The similar MAE values for different teams of robots with different mission time T provide a more objective evidence that our framework is capable of coordinating the teams well.

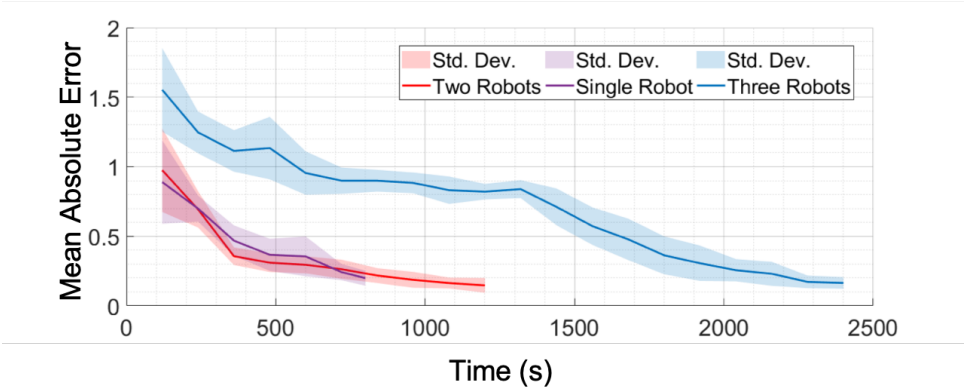


Figure 4.7: The Mean Absolute Error over 10 runs for different teams of robots. The result show the error in estimating the field given by Field 1. The consistent performance of our multi-robot framework over multiple runs provides the evidence that our framework is robust.

different teams of robots. We also calculated the mean absolute error (MAE) as a measure of performance in estimating the fields. We use this metric to examine the coordination efficiency of our framework. The MAE results are present in Fig. 4.6.

It can be observed from Fig. 4.6 that our framework's performance is similar for different teams of robots. The mission time for each team of robots is proportional to the number of robots in each team. This means that the amount of data collected by a single robot in $T = 2400$ s will be similar to the amount of data collected by a team of two robots in $T = 1200$ s. The only way the performance between these two setups will be similar if the framework efficiently coordinates the motion of the two robots to collect informative data within the limited mission time. Therefore, the similar MAE values in Fig. 4.6 for different teams of robots and for different fields is a good indication that our framework is capable of coordinating the team efficiently. We also repeated the simulations

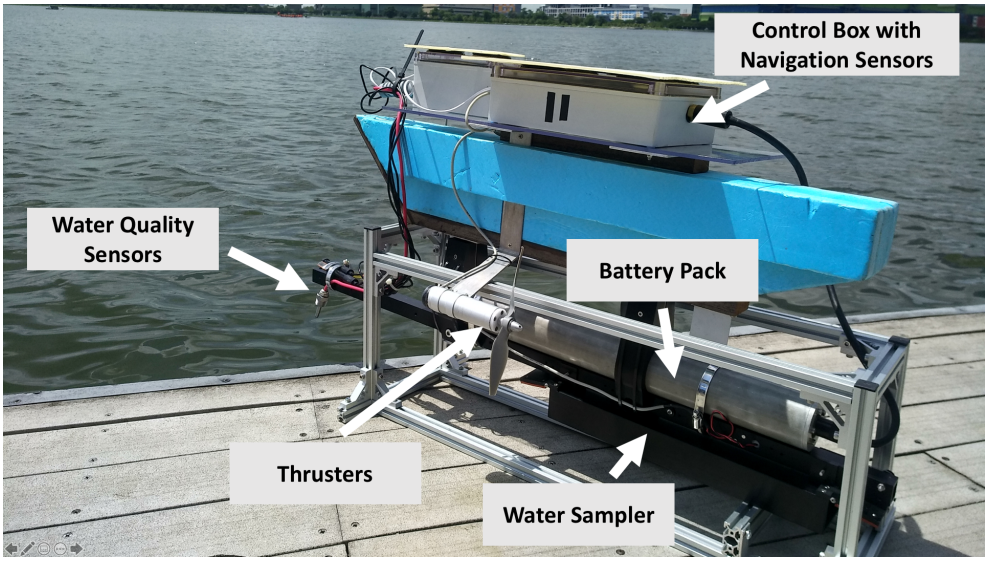


Figure 4.8: One of the robots we used in our field experiments. It is a variant of NUSwan [6]. This figure shows various components present in our robot. Our robots are capable of navigating autonomously once a waypoint is given. It is equipped with general water quality sensors and provides real-time updates of the physical and chemical parameters of water. Moreover, our robots use a middleware which enables them to receive waypoints from remote servers and provide the mission relevant information back to the server for future planning.

over 10 runs for each team of robot and recorded the mean absolute error. These results are present in Fig. 4.7 and it can be observed that our framework shows consistent performance over multiple runs.

4.4.2 Field Experiments for Performance Comparison with Lawn Mower Paths

We tested the performance of our framework against the conventional approaches such as estimating fields using Lawn Mower paths via field experiments. We developed two variants of the NUSwan [6] robot as shown in Fig 4.8. These robots were equipped with general water quality sensors such as Dissolved Oxygen, Conductivity, pH and Oxidation-Reduction Potential. Moreover, these robots

used on-board navigation sensors to reach the locations given by the framework. Our framework *m-AdaPP* was hosted on a cloud server, which can be accessed by our robots using GPRS internet connection. This cloud server was a compute instance provided by Amazon Web Service with capability to run 16 threads in parallel. This capability is crucial for our framework as it significantly reduces the computation time for making planning decisions. We optimized our framework to run smoothly on this compute instance. Both the robots posted the data to this server every 5 seconds.

For consistency, the mission time for the team of two robots for our field experiments is same as the mission time we used for two robots in our simulations, which is $T = 1200$ s and $T_s = 120$ s. In general, lawn mower paths are defined by the number of legs, where each leg is a straight path parallel to one of the axis of the survey area. Therefore, the lawn mowers are defined in terms of lengths rather than time. If the speed of the vehicle is constant, lawn mowers can be defined in terms of time but speed of the vehicle in field can vary due to external disturbances.

Imposing the temporal constraints directly on the lawn mower paths can result in abruptly stopping the lawn mower pattern. Therefore, we assume an average speed of the robots and use this average speed to calculate the total length of the lawn mower for the mission time $T = 1200$ s. We set this average speed as 0.5 m/s. It is important to note that this average speed is higher than the average speed mentioned earlier. This difference is to factor that the vehicle will be mostly moving in a straight line and thus inertia of the vehicle will help in maintaining a higher speed. Using the average speed of 0.5 m/s and mission time $T = 1200$ s, we set the length of lawn mower as 600 m.

We selected a survey field of area 150×150 m^2 in a local reservoir and used our robots to estimate the field of Dissolved Oxygen over this area. The estimated fields using the lawn mower patterns and our frameworks are present in Fig. 4.9 and Fig. 4.10, respectively. The mission time for the lawn mower paths was 1236 s and thus our assumption of a higher average speed was correct. The black and

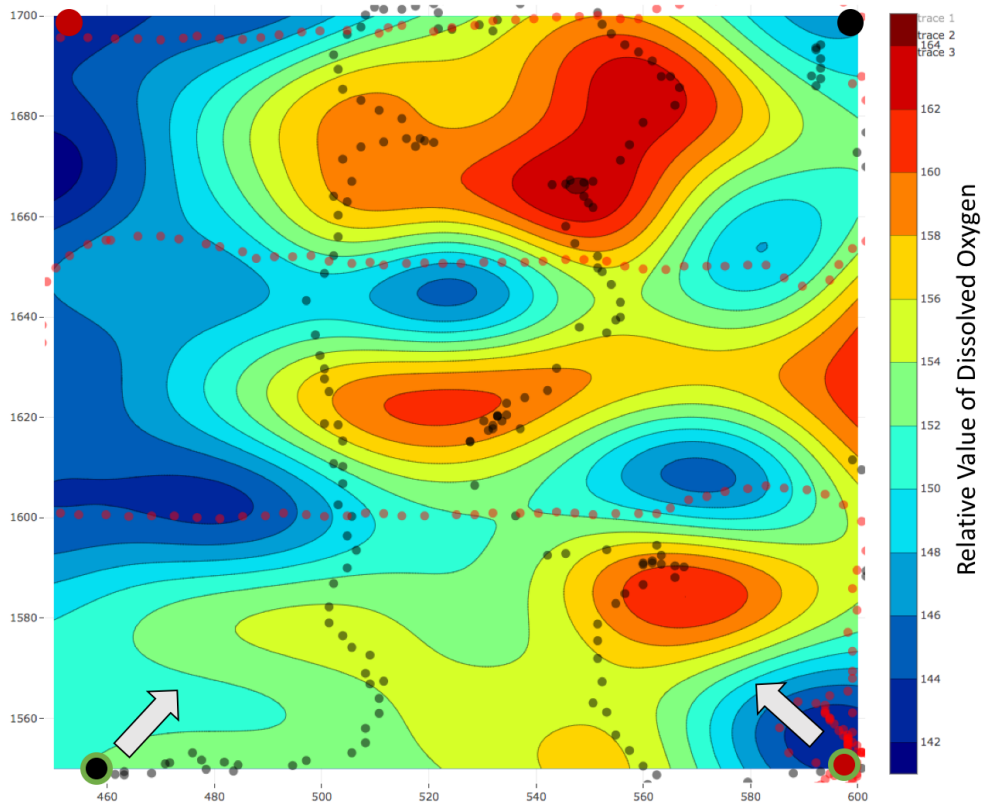


Figure 4.9: Field estimated using lawn mower patterns with a team of two robots. The estimated field is for relative Dissolved Oxygen for an area of 150×150 m^2 in a local reservoir. The black and red circles with large radius and no outline represent the starting locations of the robots. Similarly, the black and red dots represent the locations of the data collected. Finally, the black and red circles with large radius and a green outline reflect the end location of each robot and arrow represents the direction towards starting location. The total mission time for this experiment was $T = 1236$ s.

red circles with large radius and no outline represent the starting locations of the robots in Fig. 4.9 and Fig. 4.10, whereas, the circles with green outline represent the end location of the robots.

There was no prior information about the ground truth for this field of Dissovled Oxygen (DO). Therefore, we collected a test dataset to measure the performance of our framework and the lawn mower paths. This test dataset was

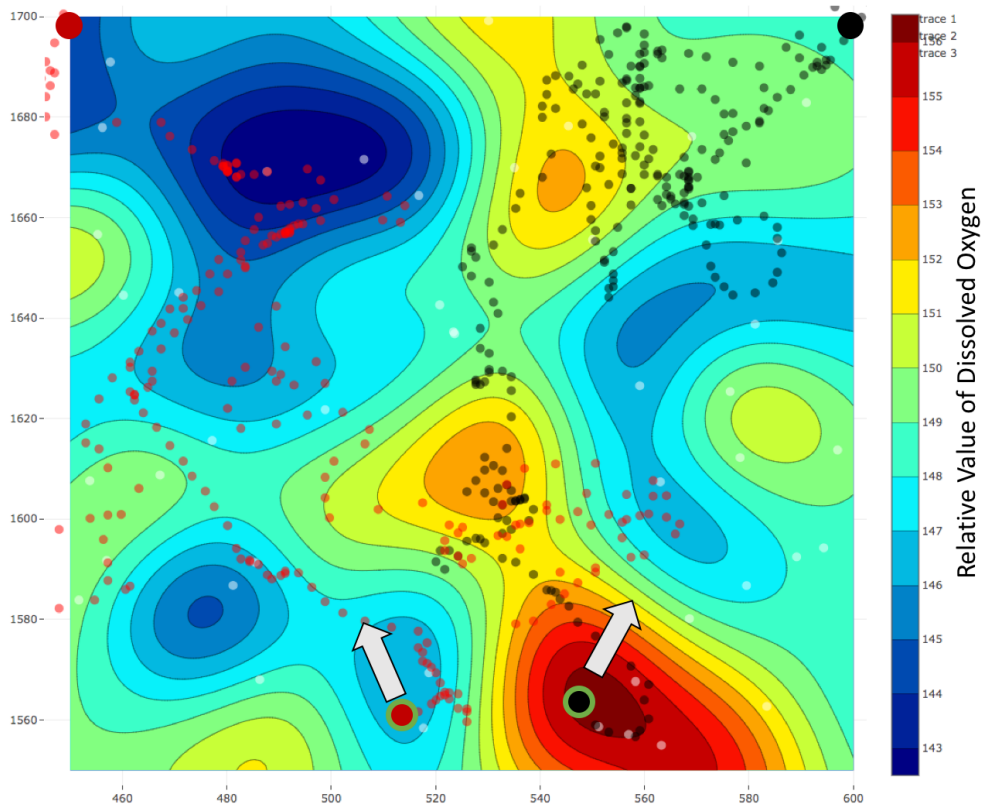


Figure 4.10: Field estimated using a team of two robots and our framework *m-AdaPP*. The estimated field is for relative Dissolved Oxygen for an area of 150×150 m^2 in a local reservoir. The black and red circles with large radius and no outline represent the starting locations of the robots. Similarly, the black and red dots represent the locations of the data collected. Finally, the black and red circles with large radius and a green outline reflect the end location of each robot and arrow represents the direction towards starting location. The total mission time for this experiment was $T = 1200$ s with $T_s = 120$ s. It is interesting to observe that data collected using this team of robots was dense in a few regions, whereas, sparse for the remaining regions. However, our framework still performs better as compared to the lawn mower pattern and this is a field validated result that collecting representative data (adaptive framework) can perform better as compared to collecting data with repetitive information (lawn mowers).

collected while robots were travelling back to its respective starting location after finishing the mission. This dataset contained both the locations as well as the ground truth data for the respective locations. We obtained the estimated DO

Estimation Using	RMSE	MAE
Lawn Mower Patterns	6.6	4.8
m-AdaPP	3.9	2.8

Table 4.1: The Root Mean Square Error and Mean Absolute Error in estimating the field of Dissolved Oxygen. These errors were calculated by using the test data collected by the team of robots while returning to the starting location. Our framework gives about 50% improvement in performance as compared to the fields estimated using lawn mowers.

value for these locations using the learned models and calculated the errors in estimation using the collected ground truth data. These results are present in Table 4.1. Our framework performs significantly better as compared to the lawn mower paths. Such performance in field experiments is a confirmation that our *m-AdaPP* framework is able to provide a good estimate of the environmental field.

4.5 Summary

We explained a framework for monitoring scalar environmental fields using a team of robots with bounds on overall mission time. We used the kernel information of the sparse GP model to explore the combinations of move available to the team of robots and collect informative data. We simulated the framework using real world data and the results show that our framework is capable of coordinating a team of robots efficiently. We also simulated multiple runs of the framework to test the robustness in our performance and the results show consistent results across multiple simulations.

We designed two robots based on the NUSwan vehicle for monitoring reservoirs in Singapore. Using this team of robots, we validated the performance of our framework in the field against conventional methods such as using lawn mower paths. The estimation error for these field experiments were based on the test data collected after finishing the monitoring task and the results show that

our framework outperforms the lawn mower approach. Overall, we explained and validated our contribution for using a team of robots to estimate a scalar environmental field in this chapter.

Chapter 5

Biological Relevance of Adaptive Frameworks

In this chapter, we examine the biological relevance of the fields estimated using the adaptive frameworks. We make use of the multi-robot adaptive framework discussed in the previous chapter and examine the quality of the samples using the standard lab-based methods. We show that the fields estimated using the frameworks can provide reasonable scientific information and thus provide a strong use case for environmental monitoring.

5.1 Adaptive Frameworks and Scientific Experiments

Adaptive monitoring frameworks have been used commonly for estimating scalar environmental fields such as chlorophyll concentration and temperature. However, the biological relevance of these estimated field is still not well established. There can be multiple interpretations of biological relevance such as estimating the hotspots or tracking a certain phenomenon which has a scientific value. However, we use this term to establish a more fundamental relationship, which is relevance of the estimated fields in understanding the interactions at a micron scale.

Some of the works have attempted to establish the connection between the fields estimated using robots and using these estimated fields to understand various environmental phenomenons. One such work is tracking the hydrocarbon plumes and bio-degradation at the Deepwater Horizon site [75, 97]. This work focused on developing a framework to observe the bio-degradation of hydrocarbon plume and it is a good example of tracking a biological phenomenon to understand

(a) (b)

Figure 5.1: Our team collecting samples after the robots have finished estimating an environmental field in Pandan Reservoir. These samples were collected using the robots and brought back to the deployment station as shown in (a). The water samplers were detached from the robot to transfer the water into a container for further lab analysis.

it at a macro-scale. However, the changes at micro-scale can result in a substantial shifts in the environmental fields [43]. The experiments in [43] were done using robots and a network of sensors. This work makes use of the fields estimated using the network and the robots to explain the changes at a micro-scale. However, it does not make full use of the fields estimated to guide the sampling strategy for studying the changes at micro-scale. Another interesting approach for establishing scientific relevance is discussed in [19]. In this approach, the authors designed two frameworks, one is to make the sampling decisions and another one to estimate the concentration of a pathogen based on the sensor values. This framework makes full use of sampling based on the information, however, the focus of this work was to select samples from the pre-defined path of an underwater glider and aimed at estimating the concentration of a particular pathogen.

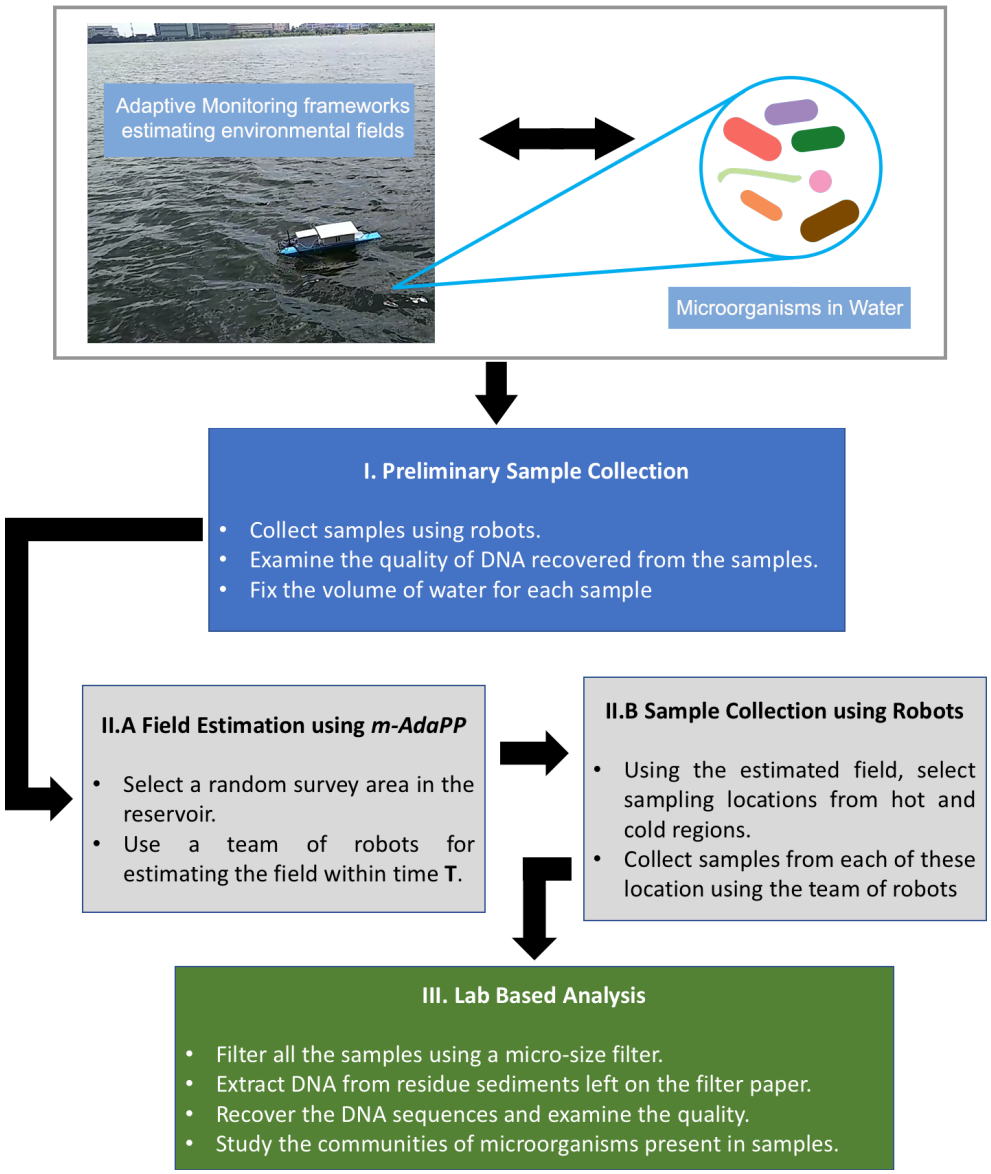


Figure 5.2: An illustration explaining the motivation for examining the fields and the flow of our experiment to examine the biological relevance of the fields estimated using environmental fields.

5.2 Experimental Setup

5.2.1 Objective

We are interested in a comprehensive examination of the fields estimated using the environmental fields. Generally, the frameworks are used to the estimate

the fields of a particular physico-chemical parameter. This estimation results in regions of hot and cold spots, which represents the region with high and low values of a particular parameter respectively. We aim to examine these estimated regions and study their relevance from a micro-scale perspective, which is identifying the microorganisms in the sample and using this identification to get differentiation between the samples. In addition, we aim to examine the quality of the collected samples by using the conventional DNA quality scores, commonly known as Phread Quality Score [98]. These quality scores can help in identifying the condition of the samples collected by checking the error probabilities in the determining the sequence of DNA [99]. A conceptual diagram explaining our experimental setup is present in Fig. 5.2.

5.2.2 Preliminary Sampling

A preliminary test was done to examine the sample volume collected from robot's sampler. The robots generally have a maximum capacity for collecting a sample, for example, our robots can collect a maximum of 1 L of water and thus it is necessary to examine whether one sample per hot or cold region will be enough. These preliminary tests were done using the robot's sampler and the locations for these samples were selected randomly, where no estimation of the field was available. We iterated over small quantities of water from these samples to find the correct water volume that has enough DNA content to facilitate other lab analysis. In each of these iteration, we was passed a fixed volume of water through a micro-size filter and the DNA was extracted from the residue on this filter. We examined the quality of DNA using gel electrophoresis.

5.2.3 Sample Collection using *m-AdaPP*

We used a team of two robots to survey the areas in a local water resource, Pandan Reservoir. For convenience, we used our adaptive framework *m-AdaPP*, which is explained in detail in the previous chapter. We used the framework to estimate Dissolved Oxygen (DO) as it plays an important role in the activities

of microorganisms and often used to define the water quality [100, 101, 102]. Following is an overview of the sampling protocol we followed:

1. Randomly select a survey area of 75 m × 75 m in the reservoir.
2. Use a team of two robots and *m-AdaPP* framework to estimate the field of Dissolved Oxygen using the on-board water quality sensors.
3. Once the estimation is finished, select the location of samples from hot and cold regions of the predicted field.
4. Use the robots to collect 1 L of water samples from the selected regions for lab analysis.

We repeated this sample collection over large temporal delays (days) and small temporal delays (hours). The collected sample were analyzed for quality of the extracted DNA as well as to find distinguishing characteristics between hot and cold regions. The following section provides an overview of these analysis.

5.2.4 Materials and Methods for Processing the Samples

Each collected sample was passed through a micrometre filter and the residue on these filters were put in a deep freezer until the DNA extraction was initiated. The residue from the filtering process contains various microorganisms which were present in the water sample. The DNA extraction from these residues were done using standard DNA extraction kits. This extraction was followed by DNA sequencing, which is used to identify basic DNA pairs and form a DNA sequence.

The recovered DNA sequences were then matched with an existing DNA database, SILVA v132 database [103], to find the exact identities of all the microorganisms present in the sample. This step required denoising of the recovered DNA sequences and this was done in R programming language using the dada2 package [104]. These steps are standards for identifying microorganisms in any water sample and we followed these steps for all our samples. However, this section only provides an overview of our sampling protocol and more details are available in Appendix A.

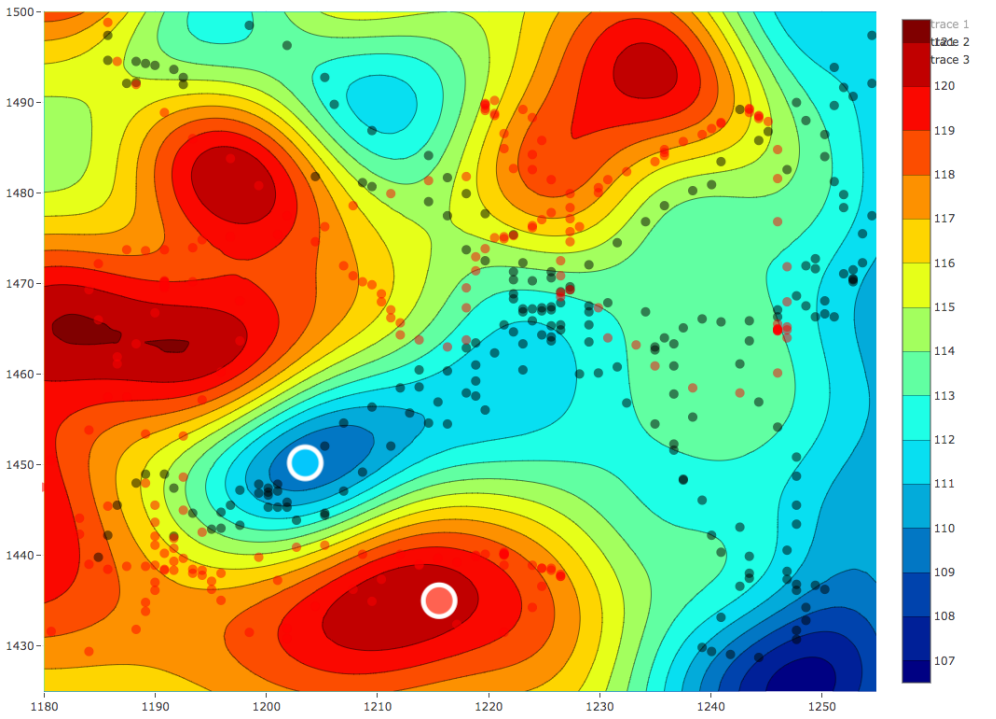
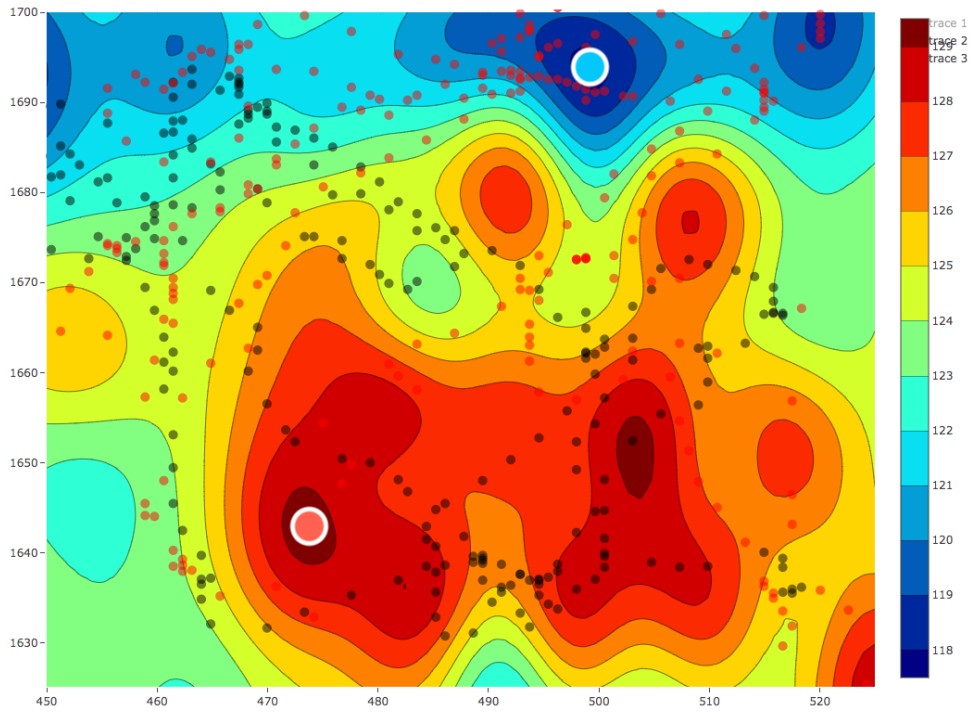


Figure 5.3: The estimated field for relative value of Dissolved Oxygen for a 75×75 m^2 area in Pandan Reservoir on 28$^{\text{th}}$ February, 2019. The red and black dots respectively represent the paths of the two robots. The red and blue circle with white outline represent the samples collected from the hot and cold region respectively.

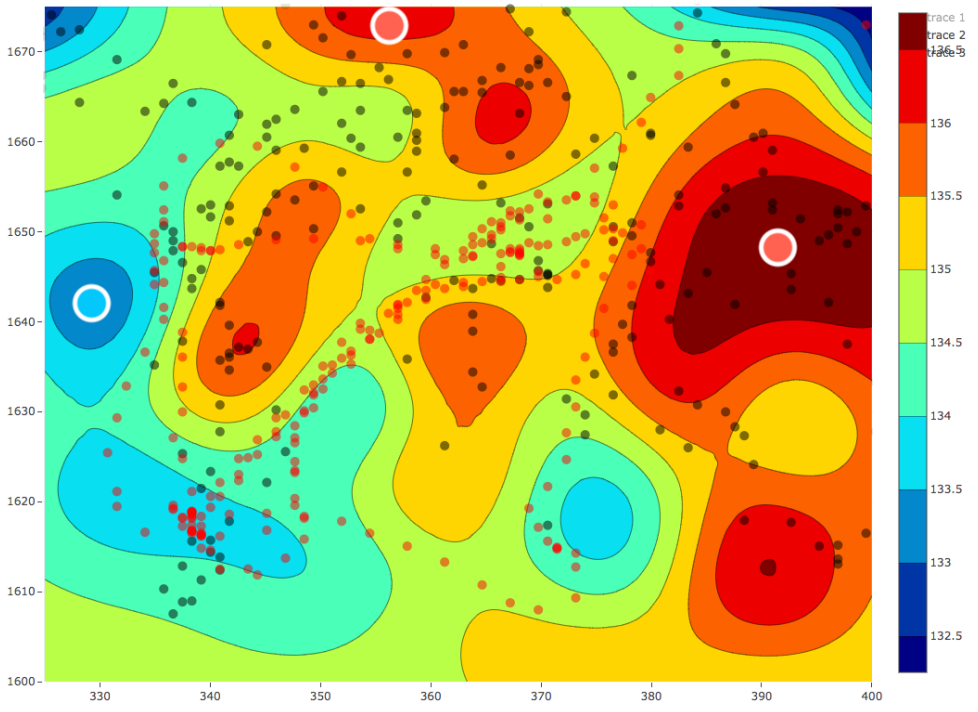
5.3 Field Experiments

We started with preliminary field experiments to identify the volume of water required for lab analysis. We collected 3 samples randomly from the local reservoir and followed the steps as explained in Section 5.2.2. The results from gel electrophoresis analysis showed that a volume of 20 ml was enough to get DNA for sequencing. Therefore, we collected 1 sample per hot or cold region.

Three field estimation tasks were done using our framework. Two out of these three estimation tasks were on the same day with temporal difference of 1 hour. Each of these estimation tasks were given a mission time of 20 minutes. Following was the overall schedule of our experiments - 1030 hours on 28/02/2019, 1305 hours on 04/03/2019 and 1425 hours on 04/03/2019.

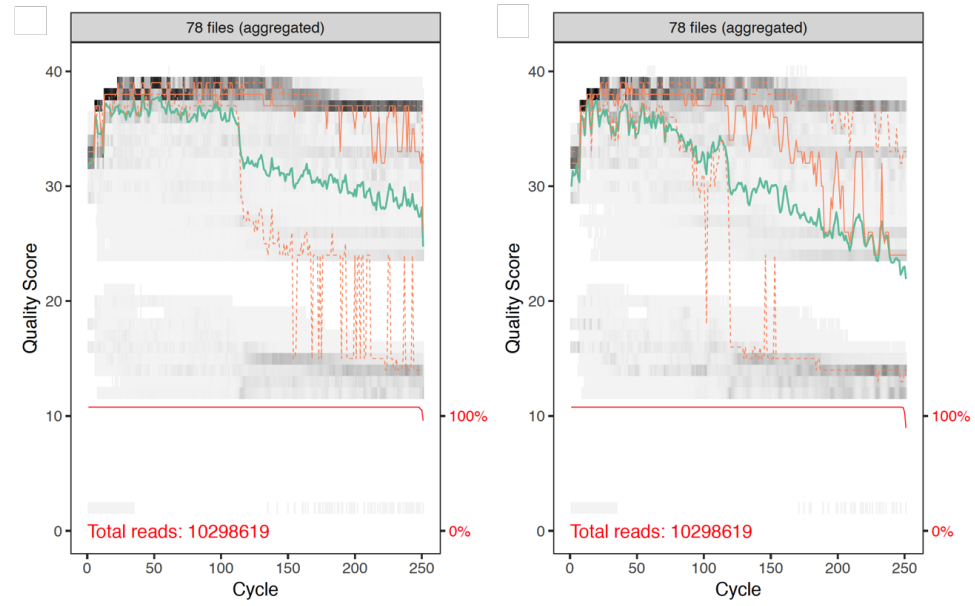


(a)

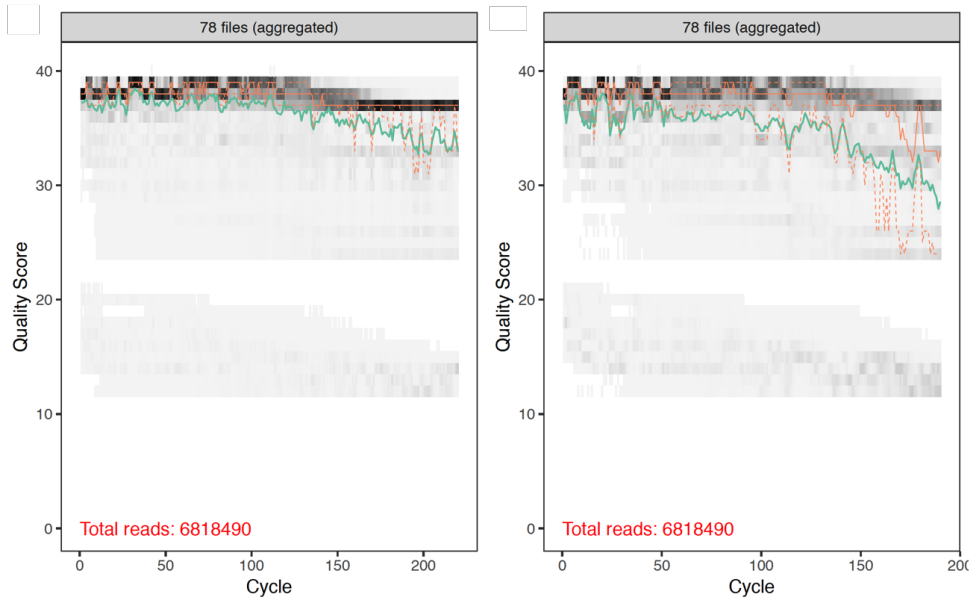


(b)

Figure 5.4: (a) The estimated fields of relative values of Dissolved Oxygen for a 75×75 m^2 area in Pandan Reservoir on 04$^{\text{th}}$ March, 2019 at (a) 1305 hours and (b) 1425 hours. The red and black dots respectively represent the paths of the two robots. The red and blue circle with white outline represent the samples collected from the hot and cold region respectively.



(a)



(b)

Figure 5.5: The Pherd Quality Score for (a) the raw DNA sequences obtained from DNA sequencing and (b) the denoised DNA sequences retained after the filtering process. The dashed orange lines represent the variance in predicting the DNA pairs in each nucleotide position in 250 nucletotide long sequence reads, which is represented by the X axis. The green line represents the mean quality score and the non-dashed orange line represents the median quality score for each nucleotide in the DNA, respectively. Similarly, the gray-scale heat map in the background of each plot shows the distribution of the quality score for each nucleotide.

The field estimated on 28/02/2019 is shown in Fig. 5.3 and the fields estimated on 04/03/2019 are shown in the Fig. 5.4. All the values of Dissolved Oxygen in these figures are a relative measure of DO instead of the true values. We manually selected the locations to sample using the estimated field and used the robot to collect 1 L of water. These sampling locations are shown as red and blue circles with white outline in respective figures (Fig. 5.3 and Fig. 5.4). Overall we collected 3 samples from the cold regions and 4 samples from the hot regions of all the estimated fields. These samples were then sent for lab analysis as described in Section 5.2.4.

The Fig. 5.5 shows the quality score of the collected samples from our three experiments. The quality score in this figure was calculated using Phread Quality Score, which is a standard approach to measure the quality of samples. The results in Fig. 5.5 are encouraging as the scores of the raw DNA sequences recovered using DNA sequencing and the denoised DNA sequences are good.

The high quality scores also mean that the collected samples can be used for finding the microorganisms present in the sample with high certainty. Therefore, we further analyzed the DNA sequences to find the exact microorganism present in our samples and examine the differences between hot and cold regions estimated by our framework.

The Fig. 5.6 shows results from analysis of Principal Coordinate [105], which is a commonly used method to find the dissimilarities between the communities of microorganisms in each sample. These dissimilarities are calculated based on the differences between the abundances of various microorganisms in each sample and with the help of ordination, it is condensed a 2 dimensional plot. Each of the points in this 2D plane were then marked with the label of hot or cold region based on whether the samples were collected from the regions of relatively higher DO levels or lower DO levels. The robustness of the association between microbial communities and DO levels is evident based on the three experiments. It can be clearly observed that the community of microorganisms living in the hot regions are substantially different from the community living in the cold regions

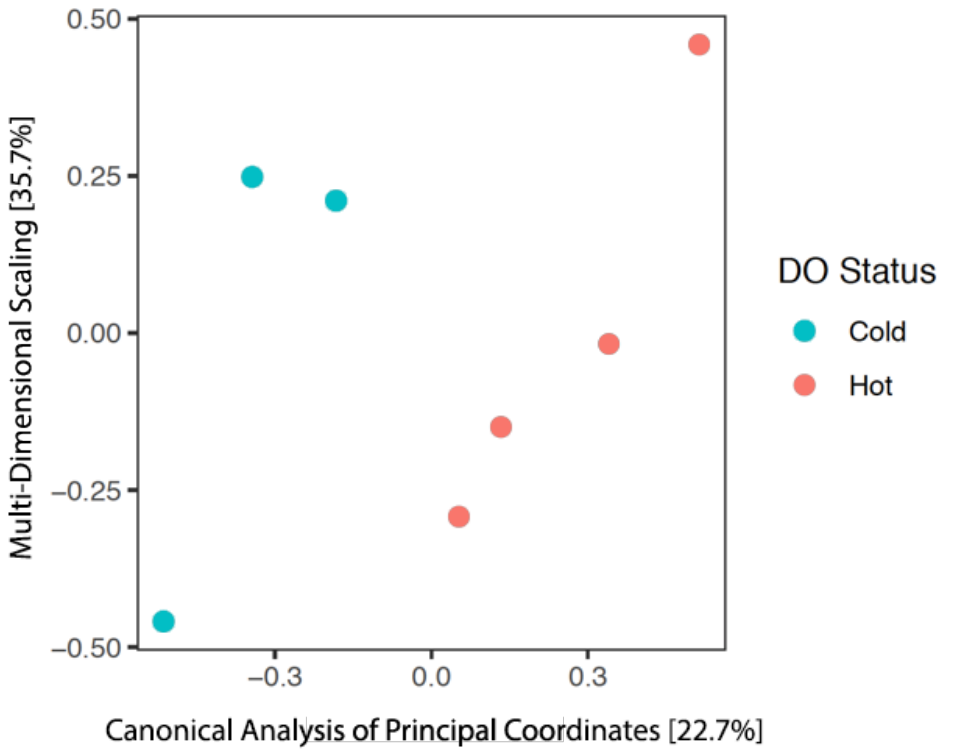


Figure 5.6: The results from Analysis of Principal Coordinates and ordination for the microbial communities within each of the 7 samples. The red dots represent the samples collected from the hot regions, whereas, the cyan dots represent the samples from cold regions of the estimated field. Percentage values of each axis represent the variation explained.

of the estimated fields. Therefore, these preliminary results are a good validation for the use of adaptive frameworks in not only estimating the heterogenous and dynamic environments but also helping in understanding the interaction within microbial communities.

5.4 Summary

In this chapter, we examined the biological relevance of the fields estimated using our multi-robot framework, *m-AdaPP*. We used the framework to estimate fields and find the regions of high (hot) and low (cold) concentrations for each survey area. This experiment was repeated over 2 days for different spatial locations.

After completing each survey, we collected physical water samples using our robots and used standard scientific protocols for analyzing the communities of microorganisms in the samples. These standard lab-based methods were sample filtering, DNA sequencing and assembling the DNA to identify different microorganisms. The results show the samples collected using our framework have a good quality score. Moreover, we analyzed our samples collected from hot and cold regions and found communities to be distinct.

Chapter 6

Adaptive Monitoring and Informed Sampling

In this chapter, we discuss the framework for informed sampling based on the fields estimated using adaptive monitoring. The frameworks explained in the Chapters 3 and 4 can be only used to provide an estimate of the field. However, most environmental studies require physical sample collection from the regions of scientific interests. These regions could be hot spots or cold spots of the estimated field. Therefore, we present a framework here which can be used for both field estimation and sample collection simultaneously from areas of scientific interest.

6.1 Exploration-vs-Exploitation in Monitoring and Sampling

We are interested in combining both the tasks of field estimation and sample collection into a single on-line framework. This can be achieved by collecting representative data such as temperature measurements to get a good approximation of the field, and simultaneously use this approximation to collect physical samples. Similar to monitoring frameworks, the combined approach can make use of the temporal constraints for field estimation and sampling task, which helps in reducing the model complexity of the environmental field. It is important to note that our problem definition falls in the category of exploration-vs-exploitation dilemma. The exploration characteristics come from the objective to get a good approximation of the field, while the exploitation behaviour come from collecting the samples from the areas of scientific interest.

Figure 6.1: Our NUSWan [7] being deployed for monitoring and water sample collection in a freshwater lake in Singapore. It has a water sampler which can be used to collect upto 1 L of water.

In Chapter 3, we discussed a single robot adaptive framework *k-AdaPP* for estimating environmental fields. At each step, *k-AdaPP* evaluates the potential of all the neighbouring locations and selects the one which minimizes the overall variance, thus making it an exploration only framework. In this chapter, we present a single robot framework to address the task of simultaneous sampling and monitoring of an environmental field and name this framwework as *SAM*.

We define the areas of hot regions as the areas of scientific interest, however, this definition can be changed to suit any other sampling design. Therefore, our framework aims to collect samples from the hot regions, while using a sparse variant of Gaussian Process (GPs) to estimate the environmental fields. Moreover, the sampling decisions are made using a combination of Upper Confidence Bound (UCB) algorithm and a user defined utility function, which provides scientists the

freedom to customize the sampling criteria as per the experiment's requirement. This user defined utility functions brings an adaptive behaviour to the sampling decision process and thus it provides a strong use case for various scientific experiments. We test the performance of our framework using sea surface temperature dataset [8] and show that our framework *SAM* is able to provide good estimates of the field as well as pick samples from the hot regions.

6.2 Problem Formulation

Our problem statement is to collect O samples from the hot regions of the estimated field, where the field estimation task is done simultaneously and has to finish within a fixed amount of time. This can be represented as

$$\mathbf{S} = \underset{{}^{i}\mathbf{x}, {}^{j}\mathbf{x} \in \mathbf{P}}{\arg\max} \sum_{i=1}^{O} \mathcal{Y}({}^{i}\mathbf{x}|\mathbf{D}_t, \mathbf{P}) \times \prod_{j=1, j\neq i}^{O} \mathcal{G}({}^{i}\mathbf{x}, {}^{j}\mathbf{x}), \tag{6.1}$$

such that

$$\mathcal{T}(\mathbf{P}) = T - t, \tag{6.2}$$

where $\mathbf{S}$ is a set containing O spatial locations from the survey area and $\mathcal{Y}({}^{i}\mathbf{x}|\mathbf{D}_t, \mathbf{P})$ is the value of the environmental parameter of a spatial location ${}^{i}\mathbf{x}$ in a path $\mathbf{P}$, which is an element of the set of paths Λ_t explained in Chapter 3. The function $\mathcal{G}({}^{i}\mathbf{x}, {}^{j}\mathbf{x})$ is used to discount the field value for a spatial location ${}^{i}\mathbf{x}$ given a sampled location ${}^{j}\mathbf{x}$ and T is the total mission time. In short, (6.1) represents the selection behavior for a set of spatial locations, which is controlled by the definition of the utility function $g(\cdot, \cdot)$ and field's distribution $f(\cdot)$. The summation in 6.1 represents the constraint on collecting all the samples from hot regions, a higher value of summation will come from samples collected from higher values of the field. However, the utility function adds a weight to each of the sample collected and thus changing the sampling behaviour to suit a scientific objective.

Broadly, we have two optimization problems to solve, one is the estimation of

the scalar environmental field and the other is the selection of samples satisfying (6.1). To have a standard terminology, we consider the task of field estimation as a monitoring or exploration problem and the task of physical sample collection as a sampling or exploitation problem. As discussed before, solving these two problems simultaneously represents an exploration-vs-exploitation dilemma. Our algorithm *k-AdaPP* is an exploration-only framework, which aims at minimizing the overall variance of the estimated field, and shows that we are able to get a good approximation of the ground truth. We use this idea of minimizing variance to address our exploration problem. We also leverage some of the algorithm's features such as spatial decomposition and sparse GP for estimating fields. However, to address the exploration-vs-exploitation dilemma we add certain elements to the *k-AdaPP* framework and these additions are discussed in the following sections.

6.2.1 Upper Confidence Bound Algorithm

Exploitation in a sequential decision making problem is a well researched area [106, 107, 108]. One of the commonly used approach for solving such problems is the Upper Confidence Bound (UCB) algorithm. In a general UCB formulation, the goal is to find the action that provides the maximum reward from the set of possible actions. The UCB algorithm uses the combination of estimated mean and variance of the rewards for each action and simply selects the action that provides maximum possible reward. The definition of the maximum possible reward defines the behaviour of the decision process and it has been formulated to be used in different applications [109, 110].

6.3 Monitoring and Sampling using Information

In this section, we explain our exploration-vs-exploitation framework *SAM*. We explain the use of *k-AdaPP* to provide an estimate of the field and the candidate paths that can be used to minimize the variance in the remaining time. We also explain the unification of these paths and selection of sampling location.

6.3.1 Using Kernel Information in Planning

Similar to the exploration-only frameworks in Chapter 3, we use sparse GP, SPGP, to estimate the field and the associated variance. At each planning iteration, our framework simulates future paths through all the neighboring cells and selects the path that minimizes the overall variance in the model. An interesting characteristic of using GPs or Sparse GPs is the estimated variance, which can be estimated using just the kernel model. In our field estimation task, this means that the change in model variance for any spatial location can be approximated without physically traversing to it. Another interesting characteristics for Sparse GPs is the prediction complexity for estimating the variance is completely independent of N and does not suffer from increasing number of training points. The combination of these two characteristics makes sparse GPs an ideal tool for quickly approximating the change in variance for all the spatial location yet to be visited.

The exploration framework in *SAM* makes use of the kernel properties of Sparse GP for evaluating different paths through the neighboring cells similar to *k-AdaPP*. The exploration framework simulates candidate paths through each neighboring cell and estimates its capability in minimizing the overall variance in the the model. Our objective is to also collect samples from the hots regions of the estimated field and therefore, merely satisfying low overall variance will not solve the problem given by (6.1).

It is important to note that in the *k-Adapp* framework, the decision is made after calculating the resulting variance of all the paths from the neighbouring locations. We define this set of path for each neighbouring location as Υ_t and resulting variance as $\mathbf{V}$. This set is different from the set Λ_t as it contains only one path for each neighbouring location. The set Υ_t will be a subset of Λ_t as it is obtained using *k-AdaPP*'s framework after searching through all the paths in Λ_t. We form this subset because the locations in the paths in set Υ_t can be good candidates for selecting the sampling points.

Algorithm 6.1: UCB based Sample Selection from a path

Data: Sampling Canditates ($\mu(\mathbf{P}) + \sigma^2(\mathbf{P})$), Remaining no. of samples (o), Sampled locations till time t ($\mathbf{S}_t$), Utility function $\mathcal{G}(\cdot,\cdot)$

Result: Greedy subset of locations that can be sampled ($\mathbf{S}_{T-t}$), Expected Reward ($\kappa^{\mathbf{P}}$)

`/* Initialization */`

Rewards before discounting $\Gamma = \mu(\mathbf{P}) + \sigma^2(\mathbf{P})$

Discount the rewards $\Gamma = \Gamma * \mathcal{G}(\mathbf{P}, {}^j\mathbf{x}) \; \forall \; {}^j\mathbf{x} \in \mathbf{S}_t$

$\mathbf{S}_{T-t} = \emptyset$

`/* Algorithm Loop */`

while $o > 0$ **do**

  Find the location of maximum reward $\mathbf{x} = \max(\Gamma)$

  Add $\mathbf{x}$ to $\mathbf{S}_{T-t}$

  Update no. of samples $o = o - 1$

  Update the rewards $\Gamma = \Gamma * \mathcal{G}(\mathbf{P}, \mathbf{x})$

  Update expected reward $\kappa^{\mathbf{P}} = \kappa^{\mathbf{P}} + \mu_*(\mathbf{x}) + \sigma^2_*(\mathbf{x})$

end

6.3.2 Selecting Sampling Locations from a Path

In an ideal scenario, we need to select the best sampling candidates that satisfy (6.1) using complete information about of $\mathcal{Y}(\cdot)$. However, we do not assume to have any prior information and the exploration happens while selecting the samples, therefore knowing the actual value of $\mathcal{Y}(\cdot)$ in advance is not possible. However, an estimate of $\mathcal{Y}(\cdot)$ can be obtained using the sparse GP's mean ($\mu(\cdot)$) and variance ($\sigma^2(\cdot)$), which are continuously updated using the data collected during exploration. In addition to this data collection, the exploration framework provides future candidate paths and the resulting variance as discussed in the previous section.

Given the estimated field and the candidate paths, the sampling problem can be divided into two layers of problems. The first layer is to select the best $O - o$ samples for each of the future paths given by the framework, where o

is the number of samples already collected. The second layer is to select one of the paths given all the O samples (exploitation) and the resulting variances (exploration) for each candidate path. The first layer of the problem follows the general formulation of making sequential decisions, whereas, the second problem represents an exploration-vs-exploitation dilemma. We first discuss the problem of finding $O - o$ sampling locations for each candidate path.

Given a future path $\mathbf{P}$ and the sparse GP's mean $\mu_*(\cdot)$ and variance $\sigma_*^2(\cdot)$, we define the spatial locations in the path $\mathbf{P}$ as the candidates for taking a sample and the sample value of each of these candidates can be identified as

$$\underset{{}^{i}\mathbf{x}\in\mathbf{P}}{\arg\max}\ \mu_*({}^{i}\mathbf{x}) + \sigma_*({}^{i}\mathbf{x}). \tag{6.3}$$

In simple terms, (6.3) represents the reward for taking a sample as the sum of mean and variance of the selected location, $\mu(\cdot) + \sigma^2(\cdot)$. With this formulation, a simple approach to find the $O - o$ sampling locations is to sort $\mu(P_i) + \sigma^2(P_i)$ in descending order and select the top $O - o$ candidates. Such a selection process can produce all samples from the same hot region, which is a valid solution but may be irrelevant for the scientific objective. Therefore, we use a utility function $\mathcal{G}(\cdot)$ to introduce the experimental objectives into our sample selection problem and adjust the sampling rewards on-the-fly. This utility function is used to control the sampling behaviour, such as minimum distance between the samples or sampling only when the sample value exceeds the user-set threshold.

Our approach to select the best sampling points from a given path is presented in Algorithm 6.1. At each sample selection step, the utility function is used to weight the rewards for the remaining sampling candidates. This is followed by selecting the next sampling location with the maximum weighted reward and this process continues until all the O sampling locations are selected for the given path. At the end, the total reward for a path is given by the sum of all weighted rewards for all the selected sampling locations.

Algorithm 6.2: SAM - Sampling and Adaptive Monitoring Framework

Data: Set of Paths from *k-AdaPP* (Υ_t), Total Reward for each path ($\kappa^{\mathbf{P}}$), Resulting Variance for the paths in Υ_t ($(\mathbf{V})$), Branching Threshold (ϵ)

Result: Selected path P_*

`/* Initialization */`

1 Find the maximum reward $\kappa = \max\{\kappa^{i\mathbf{P}} \; \forall \; {}^{i}\mathbf{P} \in \Upsilon_t\}$

2 Generate the subset of paths $\Upsilon'_t = \underset{P_i \forall \Upsilon_t}{\arg} \; \kappa^{i\mathbf{P}} > \kappa - \epsilon$

3 Generate the subset $\mathbf{V}'$ of resulting variance for paths in Υ'_t

4 **if** $|\Upsilon'_t| == 1$ **then**

5 Select the only path in Υ'_t

6 **else**

7 Find the path which has minimum remaining variance $P_* = \Upsilon'_t[\arg\min \mathbf{V}']$

8 **end**

6.3.3 Unifying Monitoring and Sampling tasks

In the previous sections, we have described the approach for generating candidate paths for exploration and selecting the best locations to sample from a given path. However, unifying these two components into one framework is not simple. The approach that we have used is given in Algorithm 6.2.

At each planning iteration, a set of candidate paths are available with information about the resulting variance and the total reward for collecting samples. This is where the exploration-vs-exploitation dilemma arises. In order to overcome it, we first sort the candidate paths given by Υ_t according to the total reward of the candidate paths $\kappa^{\mathbf{P}}$. We then form a subset Υ'_t and resulting variances $\mathbf{V}'$ for those paths where the total reward lies within a threshold ϵ of the maximum possible reward.

The idea of forming a smaller subset is similar to the Branch-and-Bound method used for efficiently narrowing down a search problem. If the size of set Υ'_t is 1, we select the only path in this set and take the first step according to it.

The set Υ'_t having only a single element represents that no other path in Υ_t has a reward within a threshold ϵ range of the maximum reward. On the other hand, if the set has more than one path, we select the path with the minimum resulting variance $\min \mathbf{V}'$. Such an approach of forming a subset based on the rewards and selecting paths using the resulting variance gives the balance required for the exploration-vs-exploitation dilemma.

6.4 Simulation Results

For our simulations, we used the real field Sea Surface Temperature (SST) data provided by the Jet Propulsion Laboratory [8]. We extracted the temperature data for an area of 200×200 km^2 each and mapped it to an area of 200×200 m^2. This mapping was done to conserve the features of the environmental field and have an area that can be explored within a practical value of T. We used the same simulation setup as explained in Chapter 3.

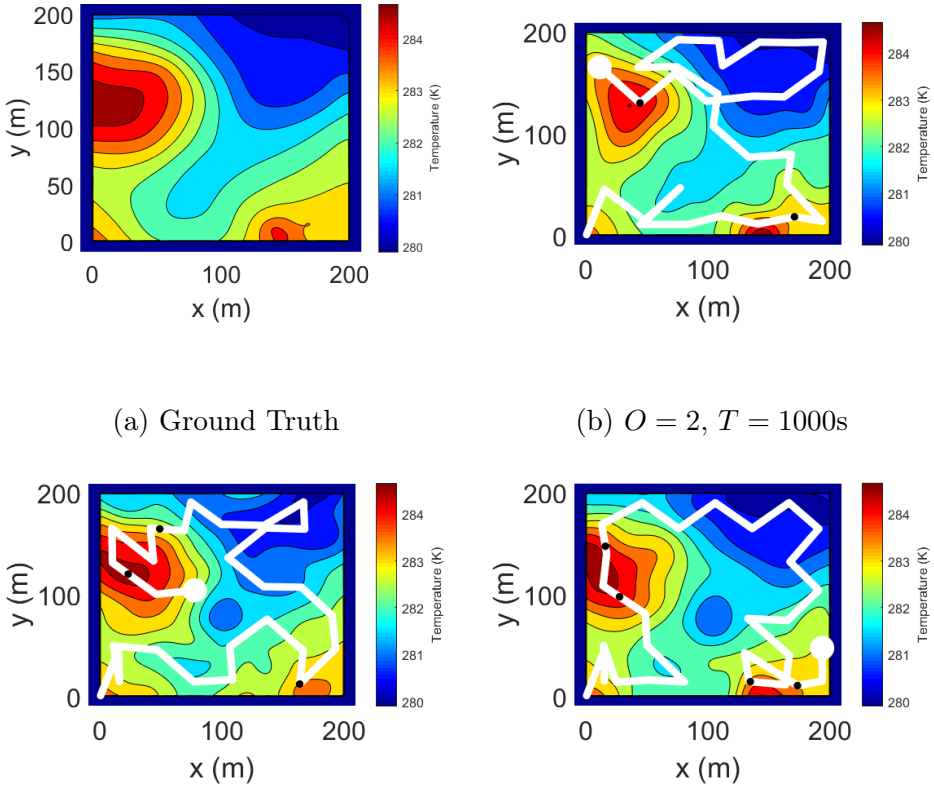
6.4.1 The Utility Function

We modeled the utility function such that the samples collected were not all from the same location, which is a general sampling strategy used used in the field. The utility function used in our simulation can be given as

$$\mathcal{G}({}^{i}\mathbf{x}, {}^{j}\mathbf{x}) = \min(\frac{||{}^{i}\mathbf{x} - {}^{j}\mathbf{x}||}{\sigma_l}, 1), \tag{6.4}$$

where σ_l is set to 50 m. The function $\mathcal{G}(\cdot, \cdot)$ penalizes the samples collected within 50 m radius of the already sampled location ${}^{j}\mathbf{x}$. The mission time is set to 1000 s and the vehicle is assumed to be traveling at constant speed of 1 m/s. We implemented our algorithm using MATLAB.

Fig. 6.2 shows the estimated fields and sampled locations using our algorithm for different valued of O given by $\{2, 3, 4\}$, with the value for ϵ set to 0.25. It can be observed that our algorithm was able to capture samples from the hots regions and provide a good approximation of the environmental field in each case.



(a) Ground Truth

(b) $O = 2$, $T = 1000$s

(c) $O = 3$, $T = 1000$s

(d) $O = 4$, $T = 1000$s

Figure 6.2: Simulation results for a 200×200 m^2 sea surface temperature field. (a) is the ground truth obtained using the MUR dataset [8]. (b)-(d) represents the field predicted using SAM for T = 1000 s and different values of O given by $\{2, 3, 4\}$. The white lines represent the paths taken by the robot and the black dots represent the locations where the samples were collected. The collected samples are all from the hot regions of the estimated field as well as the spatial difference between these samples controlled by the function $\mathcal{G}$ is significant. Moreover, the fields estimated for different values of O provide a good approximation of the ground truth.

Moreover, we used the root mean square error (RMSE) to measure the exploration performance and these results are present in Fig. 6.3. This figure shows similar transient drops in RMSE for different values of O, which demonstrates that our exploration strategy works efficiently.

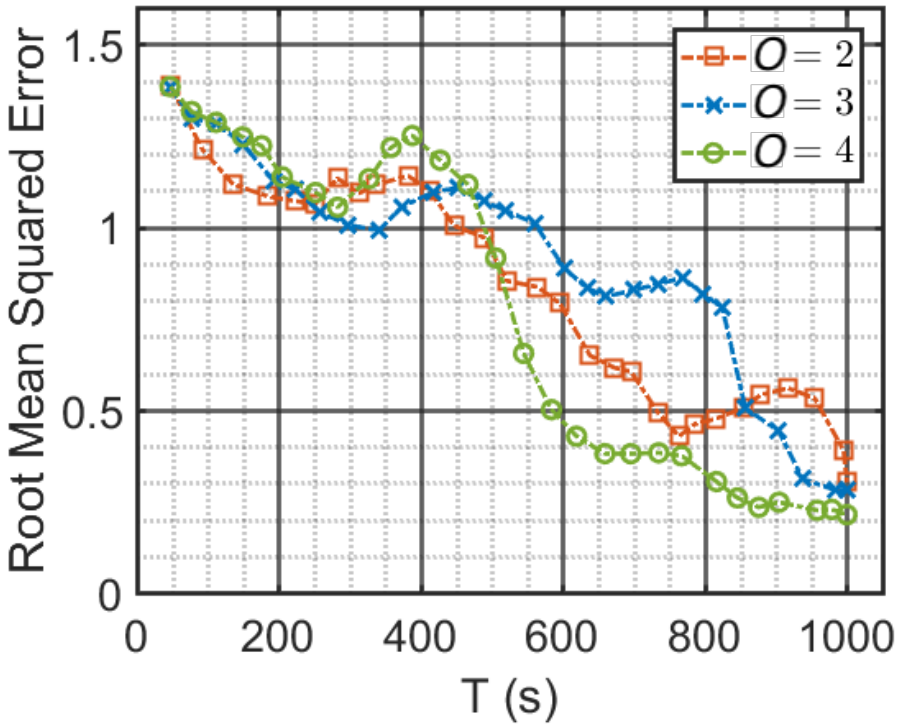
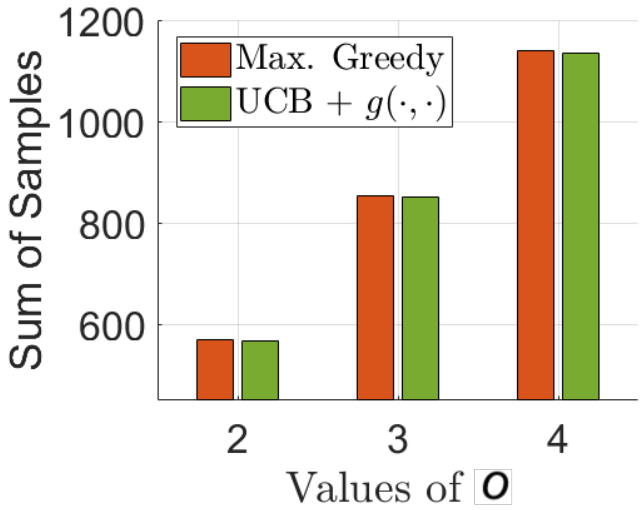


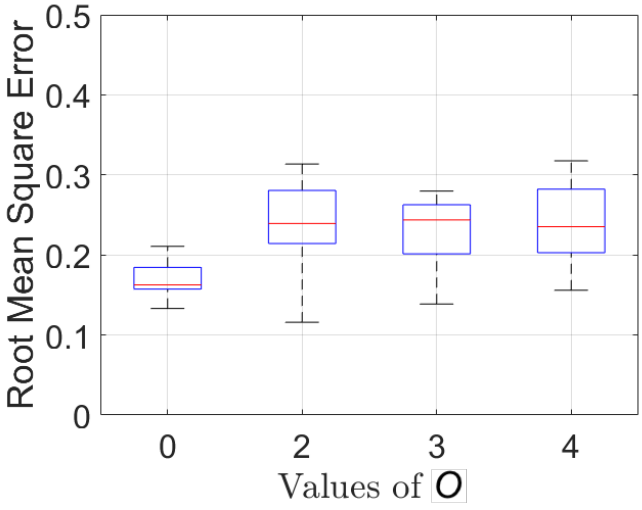
Figure 6.3: Root Mean Squared Error in estimating the ground truth for an overall mission time as $T = 1000$ s and different values of O as $\{2, 3, 4\}$. The RMSE trend for different values of O is comparable and thus it shows that the exploration framework in *SAM* is able to coordinate the robot well even with increasing the number of samples.

6.4.2 Comparison with Exploration or Exploitation only Frameworks

We further examined the exploration and exploitation characteristics of our framework *SAM*. For benchmarking the exploitation performance, we simulated a greedy algorithm that can collect samples without any discounting from the utility function $\mathcal{G}(\cdot, \cdot)$. The sampling location is also not constrained by the locations in the path of the robot. Therefore, this algorithm would result in collecting all the samples from maximum of the the temperature field given in Fig.



(a)



(b)

Figure 6.4: (a) Sum of samples collected using the greedy algorithm and mean sum of samples collected using our framework *SAM*. (b) Root Mean Squared Error in predicting the ground truth for different values of O as $\{2, 3, 4\}$. Both of these results are over 10 simulation runs.

6.2(a). The sum of samples for this algorithm will be the maximum value of the sum of samples that could be collected from the field. Similarly, We measured the exploration performance *SAM* by using our exploration-only framework *k-AdaPP* as a benchmark. An interesting observation is that our framework *SAM* can be made a exploration-only framework by removing the constraint on collecting samples, which is equating O to zero.

We simulated the greedy algorithm, *k-AdaPP* and *SAM* for each value of O and repeated these simulations 10 times. The results of these simulations are present in Fig. 6.4. The RMSEs to measure the exploration performance are present as box plots, where $O = 0$ represents the performance of the of the exploration-only framework *k-AdaPP*. The overall RMSE for *k-AdaPP* is low as compared to the fields estimated using *SAM*, however, the error difference between these two frameworks is not large. Moreover, the mean RMSE for each value of O in $[2, 3, 4]$ is similar. This provides more evidence to our observations from Fig. 6.3 that our framework provides good estimates of the field, even on increasing number of samples. The exploitation performance of our framework is shown in Fig. 6.4(a). It is clear that our framework performs as good as the greedy algorithm. These results are encouraging as the the samples collected from the greedy algorithm represents a maximum sum of samples and our frameworks scores similar sum values even with the constraints from the utility function $\mathcal{G}(\cdot, \cdot)$.

6.5 Summary

In this chapter, we presented a framework for simultaneously sampling and estimating a scalar environmental field using a single robot with the constraints on mission time. The environmental field was modeled with a sparse GP framework, SPGP, and the sampling decision process was handled using a combination of UCB and Branch-and-Bound approach.

At each decision iteration, we generate candidate paths using our *k-AdaPP* framework and select the samples for each candidate path based on a science

objective fused with UCB. The Branch-and-Bound approach helps in narrowing down the candidate paths to a reduced set of paths that provide good sampling locations. The final path is selected from this reduced set based on which path will result in lowest uncertainty at the end of mission. We simulated our framework using real world temperature data and the results show that our algorithm is able to provide a good approximation of the environmental field, while simultaneously collecting samples from the hots regions of the estimated field. This chapter provides details about our simultaneously sampling and monitoring framework, which is capable of integrating scientific sampling behaviours.

Chapter 7

Data-driven System Identification

Robots are now commonly used for environmental monitoring. These robots are equipped with environmental sensors such as water quality or weather sensors. However, the robots generally have actuators to move, which can also cause disturbance in the environmental field that it is sensing. It is thus important to quantify this disturbance and a first step in this direction is to estimate the dynamics of the robot in the environment accurately. In this chapter, we present a method to estimate the robot's dynamics model and use the motion data of one our AUV to show that our method outperforms the conventional system identification technique.

7.1 Methods for Estimating Vehicle Dynamics

Traditional approaches in modelling AUVs are based on physics models derived using first principles [76] with certain assumptions to make it linear model for easier estimation of system parameters [77]. However, AUVs with vehicle design not similar to a conventional AUV such as Iver or REMUS cannot use these linear system model. AUVs with un-conventional designs are now becoming more common and some of them are being used for environmental monitoring [78, 79, 80, 81, 2]. Therefore, it is important to have a method to learn the dynamics model of such robots.

Learning a vehicle's dynamics model can be seen as learning a function that can estimate the output parameters based on the input parameters such as velocity and control inputs. Therefore, a function approximation approach can

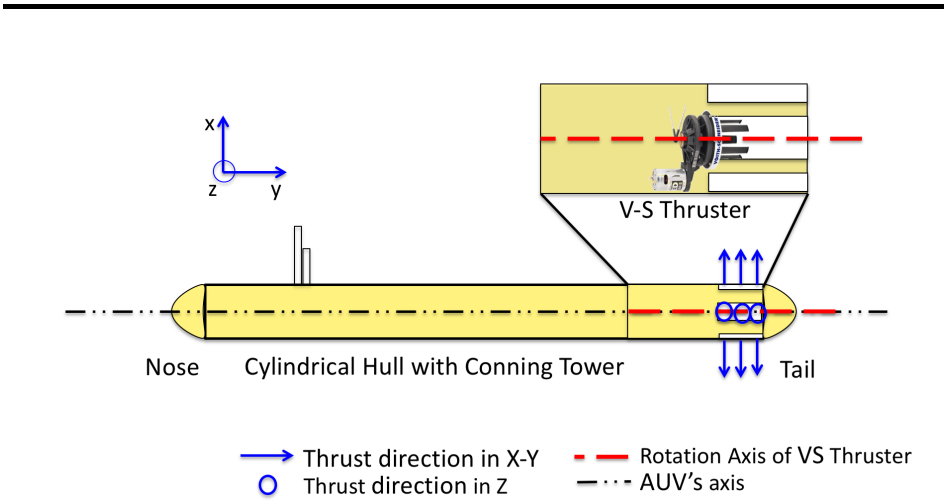


Figure 7.1: The schematic diagram of our modular AUV, STARFISH, with a Voith-Schneider (VS) thruster. This thruster is installed close to the tail and thus can be used as an alternative to conventional fin design. The blue arrows close to the tail of the AUV show the direction of thrust from the VS thruster. These arrows are perpendicular to the rotation axis of the AUV given by the dotted red-line.

be used to learn a good dynamics model of the robot. It has been shown that feed-forward neural networks have the capability of approximating any continuous function [86]. Moreover, the function approximated using neural networks can learn complex mappings between the input and output parameters. This learning is done using only examples of input and output parameters and it does not require prior information. Therefore, neural networks are a good candidate for learning the vehicle's dynamics model.

We make use of a neural network structure, known as Multi-Layer Perceptron (MLP), with a non-linear activation of its hidden units using rectifiers [90] to approximate an AUV's dynamics. In addition, we explain the methods used for optimizing our MLP's learning process. We used a conventional shape AUV with vector thrusters for our experiments. The details about the vector thruster and the state space model of our AUV are present in Section 7.2. We explain our MLP network and its optimization in Section 7.3. The details about our experiments and the performance comparison between the models are present in

Section 7.4.

7.2 Linearized Model of AUV with Vector Thruster

7.2.1 Voith-Schneider Propeller as Vector Thruster

Typical design of an AUV consists of a cylindrical hull with a horizontal thruster and four protruding control fins. The integrated motion of these four fins control the AUV's roll, pitch, yaw and depth. However, this design has limited maneuverability as the motion is dependent on the control surfaces around the AUV. For example, such AUVs under nominal speeds generally have large turning radii. In addition, the protruding fins increase the AUV's drag and are mechanically the weakest part of its body. An alternative to using these fins is vector thrusters. These thrusters can provide forces perpendicular to vehicle's body and make the turning radii smaller. One of the thrusters that has such capability is Voith-Schneider Thruster.

In sea vessels like ships or ferries, complex maneuvering such as on-the-spot turning is made possible by the use of Voith-Schneider (V-S) propulsion mechanism [111]. In general, this propulsion system consists of a cycloidal rotor that provides thrust in the direction perpendicular to its rotation axis. This thruster mechanism can be used to control the AUV's yaw and pitch and can act as an alternative to AUV's fins. In order to substitute the function efficiently, we align the rotation axis of the V-S propeller to coincide with the AUV's roll axis. Furthermore, this thruster is positioned close to the AUV's tail to replicate the effect of the four fins as shown in Fig. 7.1. This design makes the vehicle's motion independent of control surfaces and adds capabilities like thrust vectoring and on-the-spot turning. However, the non-linear behavior of the AUV increases by adding this thruster module and thus the system dynamics becomes more complex.

7.2.2 State Space Equation Model

Generally, the dynamics of an AUV is described using differential equations of motion [76]. The model equations are defined in two different coordinate frames: earth-fixed or North-East-Down (NED) frame and body-fixed frame. Six velocity components, $\upsilon = [u, v, w, p, q, r]'$ (surge, sway, heave, roll, pitch, yaw) are used to define dynamics in the body-fixed frame, where $[\cdot]'$ represents a transpose of the matrix. Similarly, the Euler angles $[\delta, \zeta, \psi]$ and the distance between NED and body-fixed frame in NED coordinate system $[d_1, d_2, d_3]$ is given as $\Theta = [d_1, d_2, d_3, \delta, \zeta, \psi]'$. These two vectors are related through Euler Angle transformation as

$$\dot{\Theta} = J\upsilon. \tag{7.1}$$

As described in [76], the nonlinear vehicle dynamics of a robotic system operating in fluids can be expressed in a compact form as:

$$I\dot{\upsilon} + \mathcal{C}(\upsilon)\upsilon + \mathcal{D}(\upsilon)\upsilon + \Omega(\Theta) = \mathcal{B}(\upsilon)\mathbf{u}, \tag{7.2}$$

where I represents the total inertia, $\mathcal{C}(\upsilon)$ represents the coriolis and centripetal forces, $\mathcal{D}(\upsilon)$ represents the hydrodynamic damping, $\Omega(\Theta)$ is the vector of restoring forces and moments, $\mathcal{B}(\upsilon)$ is the control matrix and $\mathbf{u}$ is the control inputs given to the system. The individual components in (7.2) are highly nonlinear and difficult to estimate via field experiments. Therefore, it is not practical to use (7.2) for estimating the dynamics of an AUV. A more detailed explanation of this problem is present in [76]. Therefore, an alternative solution is to separate the model into non-interacting or lightly interacting subsystems. The widely accepted solution is to divide the model into the following three categories: speed subsystem, steering subsystem and diving subsystem [77]. However, it is not feasible to divide the complex AUV designed into such subsystems, for example, the pitch and yaw dynamics are coupled for an AUV using vector thruster. Another approach for system identification for such AUVs is the state space model, which can be used to obtain a simplified approximation of vehicle's dynamics. This model can be

represented as

$$f = \mathbf{A}\ {}^{\upsilon,\Theta}\mathbf{q} + \mathbf{B}\mathbf{u}, \tag{7.3}$$

where f is a system parameter that is being modelled, ${}^{\upsilon,\Theta}\mathbf{e}$ is the vector containing the AUV's state variables and $\mathbf{A}$ and $\mathbf{B}$ are matrices of the state space model. A state space model is capable of capturing all possible linear dependencies on state variables and control inputs. Interestingly, the three subsystems mentioned in [77] can also be represented using (7.3). For example, the yaw acceleration in steering subsystem is dependent on yaw rate and rudder input. Therefore, in such cases, all the array elements of matrices $\mathbf{A}$ and $\mathbf{B}$ that are not corresponding to yaw rate and rudder input will be zero. This implies that the matrices $\mathbf{A}$ and $\mathbf{B}$ contain elements that have physical significance like components of hydrodynamic added mass or the effects of stern input. Therefore, it can be concluded that the state space representation is very similar to a linear physics model and it can be considered as a good baseline for performance comparison.

The coefficient matrices $\mathbf{A}$ and $\mathbf{B}$ in (7.3) define the accuracy of the model and they are generally estimated using least square fit on the input data. As the state space model is a linear approximation of the AUV's dynamics rather than a theoretical derivation, its accuracy will depend mostly on the quality of motion data. If the data used to estimate the coefficient matrices do not capture sufficient dynamics, it will be difficult to obtain an accurate dynamics model. Therefore, the state space model given by (7.3) offers a good linear representation of the AUV but it does not guarantee an accurate dynamics model as it is affected by the quality of motion data.

7.3 Multilayer Perceptron as Function Approximators

A Multi-Layer Perceptron is a feed-forward artificial neural network that learns the mapping between the input data and the output data. The conventional structure of an MLP is a fully connected network with three layers: an input

layer having all the input features, a hidden layer activated by an activation function and an output layer giving the final result. We have adopted the same network structure for our model. The output f' of our MLP can be expressed as

$$f' = \hat{\omega} \times \Psi(\omega \, {}^{v,\theta,\mathbf{u}}\mathbf{q} + \delta) + \hat{\delta}, \tag{7.4}$$

where $\hat{\omega}$ and $\hat{\delta}$ are the weights and biases for the hidden layer, ω and δ are the weights and biases for input layer, $\Psi(\cdot)$ is the activation function and ${}^{v,\theta,\mathbf{u}}\mathbf{e}$ is the input feature vector containing the AUV's state variables and control input.

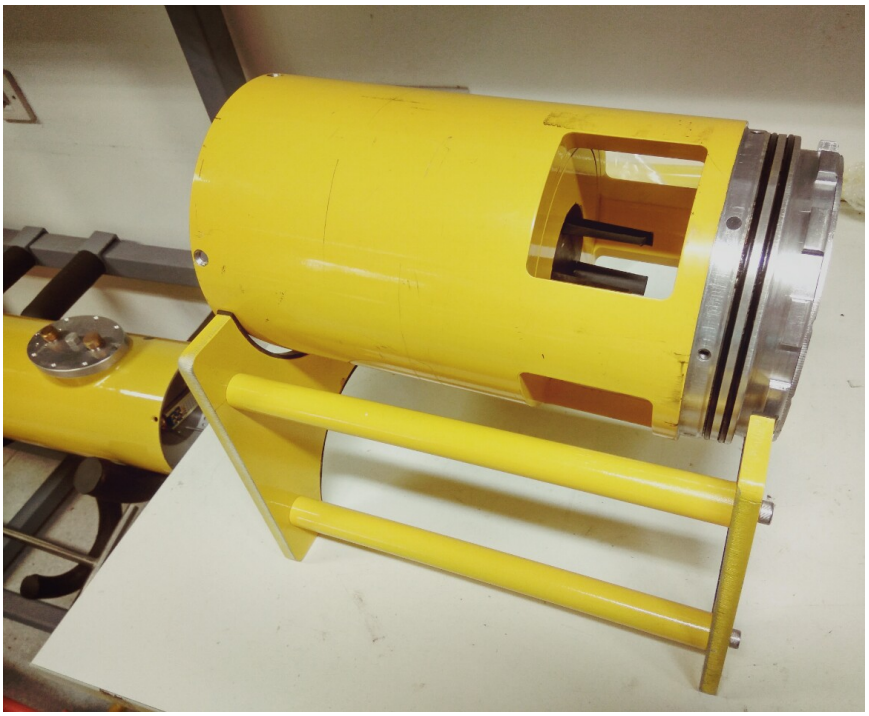
MLPs have the capability of universal function approximation because of the nonlinear transformation in its hidden layer. This nonlinear transformation is given by the activation function $\Psi(\cdot)$. This transformation of the scaled and shifted input features can either activate or deactivate each hidden unit. Therefore, some of the hidden units get activated for certain regions of the input data and learn their representation in the final output. This results in projecting the entire input dataset into a space where it becomes linearly separable with respect to each hidden unit. Therefore, the estimated output is obtained using only the hidden units activated by a particular input.

We understand that such characteristics would be useful in learning an accurate dynamics model for AUVs. For example, we know that in a linear physics model, yaw acceleration is dependent on yaw velocity and rudder deflection [77]. However, this representation is a simplified version of (7.2), which shows that yaw acceleration is dependent on other state variables as well. Therefore, we opted to design an MLP to estimate the dynamics model as it is capable of learning such underlying representations through the nonlinear transformations.

The activation function $\Psi(\cdot)$ is one of the key features of a neural network. The standard options for an activation function of MLP are sigmoid and tanh. However, recent advancements in Deep Learning have been driven by the use of rectifiers as the activation function. A simple rectifier activation function, $\Psi(\cdot)$, is given as

$$\Psi(\cdot) = \max(0, \cdot). \tag{7.5}$$

(a)



(b)

Figure 7.2: (a) Our modular AUV, STARFISH, with the (b) Voith-Schneider Module. The motion data was collected using module.

An important characteristic of such activation function is that its output is proportional to its input for all positive input values, and zero for all negative input values. This gives rectifiers a much larger active region than the tanh function. Also, such activation function helps in efficient gradient propagation during training, and does not suffer from the vanishing gradient problem [112].

7.4 Experiments

7.4.1 AUV's Motion Data

In order to test the performance of our network against the baseline linear model, we collected the motion data using our modular AUV called STARFISH. It has a thruster for horizontal propulsion and a V-S propeller for providing directional thrust as shown in Fig. 7.2. The horizontal propulsion and V-S propeller together control the yaw, pitch and depth of STARFISH. In order to suppress its roll dynamics, STARFISH has an internal rolling compensator mechanism [113]. We took the AUV to a local reservoir and programmed it to do complex manoeuvres. These complex manoeuvres included 360° on-the-spot turning. These manoeuvres were repeated for different thrust values to excite the AUV's dynamics and we recorded the data for a total of 12 minutes of such motions.

The recorded data has 13 features: orientation (δ, ζ, ψ), linear and angular velocities (u, v, w, p, q, r), two servo positions of V-S propeller's control rod, V-S propeller's rotational rpm and horizontal thruster's rpm. Some of the sensors had low sampling frequency and thus the data was interpolated to obtain a sampling rate of 10 Hz. We used a cubic polynomial fit on each subset of 10 adjacent points for this interpolation. Sensors with sampling frequency higher than 10 Hz were down sampled to achieve a fixed sampling rate across the entire dataset.

The total dataset consists of $7,314$ data points out of which $4,994$ data points were randomly selected for training, $1,070$ for validation and an equal number of data points for testing.

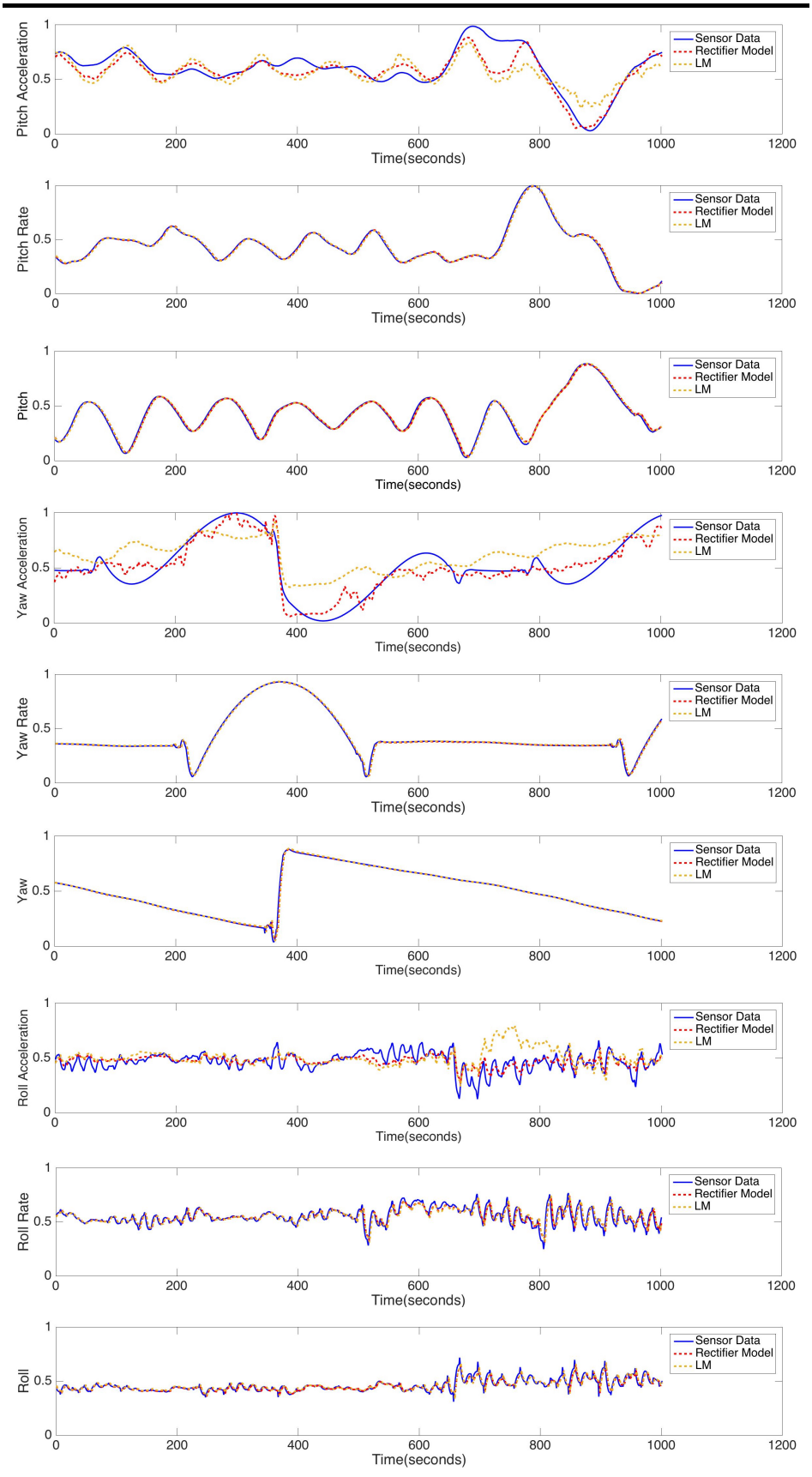


Figure 7.3: Performance comparison between our MLP network and the linear model in estimating AUV's dynamics.

7.4.2 Optimization of our State Space Model and MLP

The coefficient matrices of our baseline state space model were learned using regression analysis. This analysis was done on the training dataset and we used Levenberg-Marquardt algorithm [114] for learning the matrices. For each regression, a system parameter was selected and it was put as the output variable f given by (7.3). This was followed by the regression analysis and the best fit coefficients were learned. Generally, the output variable f is a system parameter that cannot be directly measured using any sensor but has an important role in the design of a controller for AUVs.

Our MLP model has a single hidden layer and a rectifier as an activation function. We used Gradient Descent (GD) optimizer to minimize the residual sum of squares (RSS) between the model's prediction f' and the observed dynamics. We also used RSS as a metric to determine MLP's performance on the validation and test dataset. In addition, the GD optimizer used a decaying learning rate β given as

$$\beta = {}^{0}\beta e^{-i}, \tag{7.6}$$

where ${}^{0}\beta$ is the initial learning rate, i is the optimizer's iteration number and e is the mathematical constant used in natural logarithms. We also scaled all the input features ${}^{v,\theta,\mathbf{u}}\mathbf{q}$ between 0 and 1 to maintain consistency across all the features for the entire dataset. We used the patience interval [115] method for stopping our training process. This patience parameter is used to control the number of iterations to go further in training and look for a better performance. Therefore, during training whenever a low RSS value is recorded, we extend the number of iterations by adding this patience parameter and look for a lower RSS value.

The optimization of neural networks also depend on the initialization of weights and biases, and the learning rate. The standard method for initializing is to select weights and biases randomly from a zero-mean Gaussian distribution [116]. Selecting a learning rate is critical for obtaining a good performance from

neural networks. A small learning rate β and a random initialization can result in the optimizer getting stuck in a local minima. On the other hand, a higher learning rate may result in the optimizer completely missing the global optima.

We used a two-phase initialization process to solve the problem of selecting the learning rate and the initial values of weights and biases. In the first phase, the weights and biases are initialized using a zero-mean Gaussian and trained with a relatively average learning rate β'. The value for β' is obtained from some preliminary tests using the training dataset. This value is a the of a low and high learning rate relevant to the training dataset. The training process is initiated using the decaying learning rate given by (7.6) until it is terminated by the number of training iteration exceeding the patience interval.

Before entering into the second phase, the weights and biases corresponding to the best performance from the first phase are restored. This restoration is required because the first phase was terminated because the network was not able to perform better within the patience interval. This means that the last iteration's weights and biases do not correspond to the best performance or the lowest RSS score. At the beginning of the second phase, the weights and biases are initialized using the restored values. After this second initialization, the training is performed again using a decaying learning rate with its initial value β'', which is lower than the previous learning rate β' . This type of two-phase learning process ensures that the initial weights and biases come close enough to a global minima and then critical updates are made during the second run to find the best performance.

7.4.3 Performance

We trained the baseline model and our rectifier network using the training set and used RSS as a performance metric. We used Google's open source library TensorFlow to implement both the models. The MLP model used $2,500$ hidden units and a mini-batch size of $1,000$ samples for training. The learning rate β' was set to 10^{-4} and β'' to 10^{-6}. These learning rate values were decided

System Parameter	RSS for **MLP**	RSS for **State Space**
Yaw Acceleration	4.2	19.2
Pitch Acceleration	5.2	11.0
Roll Acceleration	1.8	6.5

Table 7.1: Residual Sum of Scores (RSS) for acceleration parameters obtained using MLP and State Space model.

after some initial experimental runs using the training dataset. The patience interval was set to $10,000$ iterations. The rectifier MLP model has 13 input features as descibed in Section 7.4.1. Both the state space and MLP models are trained to predict the system parameter one time-step ahead. In real time, this one time-step prediction is equivalent to estimating the value 0.2 s ahead of the current timestamp. The test dataset performance for both the models is presented in Fig. 7.3 and Table 7.1 gives the details about the RSS value for the system parameters which were not present as an input variable.

It can be clearly observed from Fig. 7.3 that our MLP network and state space model gave promising performance in predicting orientation and angular velocities. However, MLP significantly outperformed state space models in predicting acceleration state variables. These are also the parameters which were not present as input features to our models. Therefore, no prior information was available about their current state when both models were estimating the variable's future value. This difference in performance is a good evidence that MLPs are capable of learning the underlying non-linear dynamics whereas, state space models are only able to provide rough estimates.

The Table 7.1 gives more details about the performance of both models in estimating the acceleration parameters of our AUV. The RSS values provide another verification that the MLP model learned a better representation of acceleration parameters as compared to the state space model. This means that state space models can only perform well when sufficient information about the output variable is already present as an input. However, the MLP model is able

to estimate unknown as well as known dynamics accurately and therefore, can be considered ideal for modelling complex AUV dynamics.

7.5 Summary

In this chapter, we defined a rectifier activated MLP network for learning AUV dynamics. We used the state space model as a baseline method. These models were tested for predicting system parameters for an AUV with a vector thruster. Our results show that the performance of MLP network is either better or at least as good as our baseline model. The baseline model gives a decent performance only when the output variable is present as a state input. Whereas, the MLP model accurately predicts all the system parameters irrespective of their presence as an input feature. Therefore, the MLP model is shown to be a better choice over a linear dynamics model, and capable of modelling complex AUV dynamics. Lastly, we also discussed briefly the methods for optimizing our model's learning process. The main contribution detailed in this chapter is the design of a MLP model to estimate the system parameter that cannot be measured directly.

Chapter 8

Conclusions and Future Work

In this chapter, we provide a conclusion for this thesis and the future directions that can be pursued based on the results present in this work.

8.1 Conclusions and Discussions

This work provides a comprehensive approach for monitoring and sampling environmental fields using robots. We started with the problem of estimating scalar environmental fields, such as chlorophyll concentration or dissolved oxygen. The environmental fields generally have spatio-temporal characteristics and thus estimating fields with such dynamic behaviour is difficult. However, the temporal changes in the fields generally occur over large temporal intervals. This means that the environmental fields can be assumed to be temporally constant for a short period of time. We used this assumption by constraining the total mission time for estimating the environmental fields using robots.

The modelling of spatial phenomenons can be done using Gaussian Processes (GP), a method commonly used in the field of Geostatistics. However, learning a GP model for a large dataset is computationally expensive and thus it was not ideal for real-time applications. Another model that can be used to approximate the environmental fields is sparse GP, which learns a sparse representation of the full GP model and it can be learned at a much faster rate. Therefore, all our field estimation tasks used sparse GPs to estimate the spatial distribution of the environmental fields. Specifically, we used a spase GP variant called as Sparse GP using Pseudo Inputs or SPGP.

We discussed two frameworks for using a single robot to estimate a scalar environmental field. We placed limits on the mission time for collecting the data and these constraints were introduced for assuming a temporally static field. The main difference between our two single-robot frameworks was the amount of information that was gained using GP model and integrated into the planning step. One approach used only the variance map estimated by the GP model, whereas, the other approach used the kernel function as well as the variance map to continuously update the variance as per the future paths. We termed the framework using kernel information as *k-AdaPP*.

We simulated our frameworks using real world temperature data and the results showed that both our frameworks provided a better approximation of the field as compared to traditional lawn mower paths. We further analyzed various components in our frameworks, such as the spatial decomposition algorithm. We showed using this decomposition algorithm our framework was able to perform better. Finally, we compared our framework with other methods that are used for collecting data and showed that our framework's mean performance over 10 simulation runs outperformed all the other methods. The performance improvement using our framework was considerable. It provided a substantial evidence that our framework for single robots can provide good estimates of the environmental fields using GP's information.

The overall performance of the single robot frameworks is limited by the time a robot can use to collect data. This means reducing the total amount of mission time will affect the framework's overall performance. This problem can be solved by using a team of robots. However, extending the single-robot approach to a team of robots is not simple. We suggested another framework for estimating the environmental fields using a team of robots. Similar to the single robot framework, we used the kernel information of the sparse GP model to coordinate the team of robots and collect informative data. This is also the first framework where we included the computation time as part of the mission time to provide a more field-ready framework. We termed this framework as *m-AdaPP*.

We tested the performance of our framework in simulations as well as in the field. In our simulations, we reduced the total mission time on increasing the number of robots. We showed through these simulations that our framework was able to coordinate the team of robots well. The performance of a team of two robots with half the time was similar to the performance of a single robots given the full amount of time. This similarity in performance could be only possible if the framework coordinated the motion of the robots efficiently. Moreover, we also compared the performance of this framework with conventional field estimation methods. This performance comparison was done via field experiments. We developed a team of two robots and used these robots to estimate the Dissolved Oxygen levels in a local reservoir. The results from the field experiments clearly showed that the fields estimated using our framework gave a low estimation error as compared to the fields estimated using lawn mower patterns.

Our frameworks, *k-AdaPP* for single robot and *m-AdaPP* for a team of robots, were shown to provide a good estimate of the environmental fields. Once we established that our framework performed well in reservoirs, we examined the biological relevance of the fields estimated using these framework. We used the *m-AdaPP* framework to provide an estimate of the Dissolved Oxygen levels and collected samples from the estimated hot and cold regions. We repeated this field experiment over large and short temporal delays.

The collected samples were analyzed for the quality of DNA recovered from each of the sample. Our results showed that the quality score for all our experiments was good. This means that our frameworks can not only be used for estimating the environmental fields but it can be also used for studying microorganisms with a more informed sampling approach. We used this preliminary set of data to explore the differences between the microorganisms existing in hot and cold regions. Our results showed that there is a clear difference between the microorganisms existing in the two regions. This is an interesting result and the cause for this difference can be further examined with a larger number of samples.

In the field experiments for examining the biological relevance, the samples were collected after the robots had finished estimating the environmental field. This is a general approach in most of the scientific studies. Such a sequential approach also introduces a temporal delay between sensing a location and collecting a sample from it. Therefore, we presented a framework for simultaneously sampling and estimating a scalar environmental field using a single robot. We handled the decision problem in this framework by using a combination of Upper Confidence Bound and Branch-and-Bound approach. We also provided a function that can be modelled by scientists to put additional constraints on selecting the samples. Our simulations were done using real world temperature data and the results demonstrated that our framework was able to provide a good approximation of the environmental field, while simultaneously collecting samples from the hot regions of the estimated field. We termed this framework as *SAM*

In this work, we have provided a complete set of frameworks that can be used to address various requirements of a general environmental study. All our frameworks are based on using robots for collecting informative data. However, the robots are generally equipped with actuators to move and these actuators or the motion of the robots itself can cause a disturbance to the parameter it is sensing. It is necessary to quantify this disturbance and we took the first step in this direction by suggesting a method for estimating the robot's motion accurately. We used a neural network structure, also known as MLP, to develop a data-driven system identification method. We bench-marked its performance against a linear model to estimate the complex dynamics of an AUVs with vector thruster. Our results represented that our MLP method outperformed the linear approach for all the variables that could not be directly measured using the on-board sensors.

We developed following frameworks as part of this work:

- **k-AdaPP**: A single-robot framework for estimating the scalar environmental fields using Sparse GPs. The path selection was based on the kernel model learned using the collected data. This framework was shown to

outperform standard monitoring approaches in simulation.

- **m-AdaPP**: A multi-robot framework for managing a team of robots to estimate the scalar environmental fields. This framework was validated in simulations as well in field. It was shown to manage the team of robots efficiently and perform better as compared to lawn mower paths.

- **SAM**: Single-robot framework for simultaneously monitoring and physical sample collection from an environmental field. It can incorporate a science based objective and adapt it's sampling behaviour based on this objective. We showed this framework could provide a good estimate of the environmental field while collecting the samples based on the scientific objective.

8.2 Future Work

This section suggests a few directions that can be pursued as continuation to the work in this thesis.

- **Long Term Monitoring:** Our current frameworks perform the field estimation task with a constraint on the total amount of time. We will identify the approaches to relax this constraint and develop a framework that can be used for long term monitoring.

- **Distributed Framework for a Team of Robots:** Increasing the team size for our centralized framework can increase the computation time significantly. We will devise a computationally efficient and distributed data collection framework and test its performance against our centralized framework.

- **Shifting from Estimation to Searching:** We will re-structure our problem statement to shift the goal of the framework from just estimating the environmental fields to searching for areas of high or low concentrations.

- **Generalizing the Frameworks as Optimizers:** We will derive the framework as a generalized optimizers and identify new areas of applications such as fast convergence for convex optimization problems.

- **Examining the Disturbance from Robot Motion:** We will examine the disturbances of environmental parameters caused by the movement of robots using our dynamics model estimator.

References

[1] D. Ribas, N. Palomeras, P. Ridao, M. Carreras, and A. Mallios, "Girona 500 auv: From survey to intervention," *IEEE/ASME Transactions on mechatronics*, vol. 17, no. 1, pp. 46–53, 2012.

[2] T. Salumäe, R. Raag, J. Rebane, A. Ernits, G. Toming, M. Ratas, and M. Kruusmaa, "Design principle of a biomimetic underwater robot u-cat," in *Oceans-St. John's, 2014*. IEEE, 2014, pp. 1–5.

[3] M. Dunbabin, J. Roberts, K. Usher, G. Winstanley, and P. Corke, "A hybrid auv design for shallow water reef navigation," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE, 2005, pp. 2105–2110.

[4] N. Cao, K. H. Low, and J. M. Dolan, "Multi-robot informative path planning for active sensing of environmental phenomena: A tale of two algorithms," in *AAMAS 2013*, 2013, pp. 7–14.

[5] JPL MUR MEaSUREs Project, "Ghrsst level 4 mur global foundation sea surface temperature analysis," 2010, dataset accessed [2018-01-30] at http://dx.doi.org/10.5067/GHGMR-4FJ01.

[6] T. B. Koay, A. Raste, Y. H. Tay, Y. Wu, A. Mahadevan, S. P. Tan, J. Lim, M. Chitre, and C. N. Ong, "Interactive monitoring in reservoirs using nuswan–preliminary field results," *Water Practice and Technology*, vol. 12, no. 4, pp. 806–817, 2017.

[7] T. B. Koay, M. Chitre, and C. N. Ong, "Using swan to monitor water quality in reservoirs," *Innovation in water - Research highlights by PUB*, vol. 7, p. 15, 2015.

[8] N/A, "Jpl mur measures project, ghrsst level 4 mur global foundation sea surface temperature analysis," vol. 2, 2010.

[9] N. B. Grimm, S. H. Faeth, N. E. Golubiewski, C. L. Redman, J. Wu, X. Bai, and J. M. Briggs, "Global change and the ecology of cities," *science*, vol. 319, no. 5864, pp. 756–760, 2008.

[10] C. Tortajada, "Water management in singapore," *Water Resources Development*, vol. 22, no. 2, pp. 227–240, 2006.

[11] R. B. Jackson, S. R. Carpenter, C. N. Dahm, D. M. McKnight, R. J. Naiman, S. L. Postel, and S. W. Running, "Water in a changing world," *Ecological applications*, vol. 11, no. 4, pp. 1027–1045, 2001.

[12] M. E. McClain, E. W. Boyer, C. L. Dent, S. E. Gergel, N. B. Grimm, P. M. Groffman, S. C. Hart, J. W. Harvey, C. A. Johnston, E. Mayorga *et al.*, "Biogeochemical hot spots and hot moments at the interface of terrestrial and aquatic ecosystems," *Ecosystems*, vol. 6, no. 4, pp. 301–312, 2003.

[13] J.-L. Kuang, L.-N. Huang, L.-X. Chen, Z.-S. Hua, S.-J. Li, M. Hu, J.-T. Li, and W.-S. Shu, "Contemporary environmental variation determines microbial diversity patterns in acid mine drainage," *The ISME journal*, vol. 7, no. 5, pp. 1038–1050, 2013.

[14] S. D. Gaines and M. W. Denny, "The largest, smallest, highest, lowest, longest, and shortest: extremes in ecology," *Ecology*, vol. 74, no. 6, pp. 1677–1692, 1993.

[15] J. Yu, M. Schwager, and D. Rus, "Correlated orienteering problem and its application to persistent monitoring tasks," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1106–1118, 2016.

[16] G. S. Sukhatme, A. Dhariwal, B. Zhang, C. Oberg, B. Stauffer, and D. A. Caron, "Design and development of a wireless robotic networked aquatic microbial observing system," *Environmental Engineering Science*, vol. 24, no. 2, pp. 205–215, 2007.

[17] K. H. Low, J. M. Dolan, and P. Khosla, "Active markov information-theoretic path planning for robotic environmental sensing," in *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*. International Foundation for Autonomous Agents and Multiagent Systems, 2011, pp. 753–760.

[18] J. Das, K. Rajany, S. Frolovy, F. Pyy, J. Ryany, D. A. Caronz, and G. S. Sukhatme, "Towards marine bloom trajectory prediction for auv mission planning," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 4784–4790.

[19] J. Das, F. Py, J. B. Harvey, J. P. Ryan, A. Gellene, R. Graham, D. A. Caron, K. Rajan, and G. S. Sukhatme, "Data-driven robotic sampling for marine ecosystem monitoring," *The International Journal of Robotics Research*, vol. 34, no. 12, pp. 1435–1452, 2015.

[20] S. Manjanna, A. Q. Li, R. N. Smith, I. Rekleitis, and G. Dudek, "Adaptive exploration and sampling by heterogeneous robotic team," in *Robotics and Automation (ICRA), 2018 IEEE International Conference on*. IEEE, 2018.

[21] M. Dunbabin and L. Marques, "Robots for environmental monitoring: Significant advancements and applications," *IEEE Robotics & Automation Magazine*, vol. 19, no. 1, pp. 24–39, 2012.

[22] N. D. Le and J. V. Zidek, *Statistical analysis of environmental space-time processes*. Springer Science & Business Media, 2006.

[23] R. Webster and M. Oliver, *Geostatistics for Environmental Scientists*. John Wiley & Sons Inc, 2007.

[24] D. Higdon, "A process-convolution approach to modelling temperatures in the north atlantic ocean," *Environmental and Ecological Statistics*, vol. 5, no. 2, pp. 173–190, 1998.

[25] S. Krause, J. Lewandowski, N. B. Grimm, D. M. Hannah, G. Pinay, K. McDonald, E. Martí, A. Argerich, L. Pfister, J. Klaus *et al.*, "Ecohydrological interfaces as hot spots of ecosystem processes," *Water Resources Research*, vol. 53, no. 8, pp. 6359–6376, 2017.

[26] E. Frichot, S. D. Schoville, G. Bouchard, and O. François, "Testing for associations between loci and environmental gradients using latent factor mixed models," *Molecular biology and evolution*, vol. 30, no. 7, pp. 1687–1699, 2013.

[27] M. M. Palta, J. G. Ehrenfeld, and P. M. Groffman, "âĂIJhotspotsâĂİ and âĂIJhot momentsâĂİ of denitrification in urban brownfield wetlands," *Ecosystems*, vol. 17, no. 7, pp. 1121–1137, 2014.

[28] G. Zhu, S. Wang, W. Wang, Y. Wang, L. Zhou, B. Jiang, H. J. O. Den Camp, N. Risgaard-Petersen, L. Schwark, Y. Peng *et al.*, "Hotspots of anaerobic ammonium oxidation at land–freshwater interfaces," *Nature Geoscience*, vol. 6, no. 2, p. 103, 2013.

[29] A. Darrouzet-Nardi and W. D. Bowman, "Hot spots of inorganic nitrogen availability in an alpine-subalpine ecosystem, colorado front range," *Ecosystems*, vol. 14, no. 5, pp. 848–863, 2011.

[30] D. A. Sutherland and M. A. O'Neill, "Hydrographic and dissolved oxygen variability in a seasonal pacific northwest estuary," *Estuarine, Coastal and Shelf Science*, vol. 172, pp. 47–59, 2016.

[31] Y. Zhang, Z. Wu, M. Liu, J. He, K. Shi, Y. Zhou, M. Wang, and X. Liu, "Dissolved oxygen stratification and response to thermal structure and long-term climate change in a large and deep subtropical reservoir (lake qiandaohu, china)," *Water research*, vol. 75, pp. 249–258, 2015.

[32] E. F. Wood, J. K. Roundy, T. J. Troy, L. Van Beek, M. F. Bierkens, E. Blyth, A. de Roo, P. Döll, M. Ek, J. Famiglietti *et al.*, "Hyperresolution

global land surface modeling: Meeting a grand challenge for monitoring earth's terrestrial water," *Water Resources Research*, vol. 47, no. 5, 2011.

[33] G. Zhu, S. Wang, L. Zhou, Y. Wang, S. Zhao, C. Xia, W. Wang, R. Zhou, C. Wang, M. S. Jetten *et al.*, "Ubiquitous anaerobic ammonium oxidation in inland waters of china: an overlooked nitrous oxide mitigation process," *Scientific reports*, vol. 5, p. 17306, 2015.

[34] K. A. Capps and A. S. Flecker, "Invasive fishes generate biogeochemical hotspots in a nutrient-limited system," *PLoS One*, vol. 8, no. 1, p. e54093, 2013.

[35] D. Roemmich, O. Boebel, Y. Desaubies, H. Freeland, K. Kim, B. King, P.-Y. Le Traon, R. Molinari, B. W. Owens, S. Riser *et al.*, "Argo: The global array of profiling floats," 2001.

[36] A. Mignot, H. Claustre, J. Uitz, A. Poteau, F. D'Ortenzio, and X. Xing, "Understanding the seasonal dynamics of phytoplankton biomass and the deep chlorophyll maximum in oligotrophic environments: A bio-argo float investigation," *Global Biogeochemical Cycles*, vol. 28, no. 8, pp. 856–876, 2014.

[37] E. Siswanto, J. Ishizaka, A. Morimoto, K. Tanaka, K. Okamura, A. Kristijono, and T. Saino, "Ocean physical and biogeochemical responses to the passage of typhoon meari in the east china sea observed from argo float and multiplatform satellites," *Geophysical Research Letters*, vol. 35, no. 15, 2008.

[38] S. Hosoda, T. Suga, N. Shikama, and K. Mizuno, "Global surface layer salinity change detected by argo and its implication for hydrological cycle intensification," *Journal of Oceanography*, vol. 65, no. 4, p. 579, 2009.

[39] E. V. Stanev, Y. He, J. Staneva, and E. Yakushev, "Mixing in the black sea detected from the temporal and spatial variability of oxygen and sulfide–

argo float observations and numerical modelling," *Biogeosciences*, vol. 11, no. 20, pp. 5707–5732, 2014.

[40] G. Hitz, A. Gotovos, M.-É. Garneau, C. Pradalier, A. Krause, R. Y. Siegwart *et al.*, "Fully autonomous focused exploration for robotic environmental monitoring," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 2658–2664.

[41] A. Pascoal, P. Oliveira, C. Silvestre, L. Sebastião, M. Rufino, V. Barroso, J. Gomes, G. Ayela, P. Coince, M. Cardew *et al.*, "Robotic ocean vehicles for marine science applications: the european asimov project," in *Oceans 2000 MTS/IEEE Conference and Exhibition*, vol. 1. IEEE, 2000, pp. 409–415.

[42] Y. Zhang, J. P. Ryan, J. G. Bellingham, J. B. Harvey, and R. S. McEwen, "Autonomous detection and sampling of water types and fronts in a coastal upwelling system by an autonomous underwater vehicle," *Limnology and Oceanography: Methods*, vol. 10, no. 11, pp. 934–951, 2012.

[43] D. A. Caron, B. Stauffer, S. Moorthi, A. Singh, M. Batalin, E. A. Graham, M. Hansen, W. J. Kaiser, J. Das, A. Pereira *et al.*, "Macro-to fine-scale spatial and temporal distributions and dynamics of phytoplankton and their environmental driving forces in a small montane lake in southern california, usa," *Limnology and Oceanography*, vol. 53, no. 5part2, pp. 2333–2349, 2008.

[44] R. Camilli, C. M. Reddy, D. R. Yoerger, B. A. Van Mooy, M. V. Jakuba, J. C. Kinsey, C. P. McIntyre, S. P. Sylva, and J. V. Maloney, "Tracking hydrocarbon plume transport and biodegradation at deepwater horizon," *Science*, vol. 330, no. 6001, pp. 201–204, 2010.

[45] W. E. Hart and R. Murray, "Review of sensor placement strategies for contamination warning systems in drinking water distribution systems," *Journal of Water Resources Planning and Management*, vol. 136, no. 6, pp. 611–619, 2010.

[46] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies," *Journal of Machine Learning Research*, vol. 9, no. Feb, pp. 235–284, 2008.

[47] A. Singh, A. Krause, C. Guestrin, and W. J. Kaiser, "Efficient informative sensing using multiple robots," *Journal of Artificial Intelligence Research*, vol. 34, pp. 707–755, 2009.

[48] B. Zhang and G. S. Sukhatme, "Adaptive sampling for estimating a scalar field using a robotic boat and a sensor network," in *Robotics and Automation, 2007 IEEE International Conference on.* IEEE, 2007, pp. 3673–3680.

[49] K. H. Low, J. M. Dolan, and P. Khosla, "Adaptive multi-robot wide-area exploration and mapping," in *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 1.* International Foundation for Autonomous Agents and Multiagent Systems, 2008, pp. 23–30.

[50] J. Binney, A. Krause, and G. S. Sukhatme, "Informative path planning for an autonomous underwater vehicle," in *Robotics and automation (ICRA), 2010 IEEE international conference on.* IEEE, 2010, pp. 4791–4796.

[51] S. Kemna, J. G. Rogers, C. Nieto-Granda, S. Young, and G. S. Sukhatme, "Multi-robot coordination through dynamic voronoi partitioning for informative adaptive sampling in communication-constrained environments," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on.* IEEE, 2017, pp. 2124–2130.

[52] K.-C. Ma, L. Liu, and G. S. Sukhatme, "Informative planning and online learning with sparse gaussian processes," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on.* IEEE, 2017, pp. 4292–4298.

[53] G. Hitz, E. Galceran, M.-È. Garneau, F. Pomerleau, and R. Siegwart, "Adaptive continuous-space informative path planning for online environ-

mental monitoring," *Journal of Field Robotics*, vol. 34, no. 8, pp. 1427–1449, 2017.

[54] R. Mishra, M. Chitre, and S. Swarup, "Online informative path planning using sparse gaussian processes," in *OCEANS 2018 MTS/IEEE Conference and Exhibition*. IEEE, 2018.

[55] G. Hollinger and S. Singh, "Proofs and experiments in scalable, near-optimal search by multiple robots," *Proceedings of Robotics: Science and Systems IV, Zurich, Switzerland*, vol. 1, 2008.

[56] A. Meliou, A. Krause, C. Guestrin, and J. M. Hellerstein, "Nonmyopic informative path planning in spatio-temporal models," in *AAAI*, vol. 10, no. 4, 2007, pp. 16–7.

[57] K. H. Low, J. Chen, J. M. Dolan, S. Chien, and D. R. Thompson, "Decentralized active robotic exploration and mapping for probabilistic field classification in environmental sensing," in *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*. International Foundation for Autonomous Agents and Multiagent Systems, 2012, pp. 105–112.

[58] D. Hydraulics, "Delft3d-flow user manual," *Delft, the Netherlands*, 2006.

[59] A. F. Shchepetkin and J. C. McWilliams, "The regional oceanic modeling system (roms): a split-explicit, free-surface, topography-following-coordinate oceanic model," *Ocean Modelling*, vol. 9, no. 4, pp. 347–404, 2005.

[60] R. N. Smith, A. Pereira, Y. Chao, P. P. Li, D. A. Caron, B. H. Jones, and G. S. Sukhatme, "Autonomous underwater vehicle trajectory design coupled with predictive ocean models: A case study," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 4770–4777.

[61] "YSI | Water Quality Sampling and Monitoring Meters and Instruments for dissolved oxygen, pH, turbidity." [Online]. Available: https://www.ysi.com/

[62] E. Snelson and Z. Ghahramani, "Sparse gaussian processes using pseudo-inputs," in *Advances in Neural Information Processing Systems*, 2006, pp. 1257–1264.

[63] C. E. Rasmussen, "Gaussian processes in machine learning," in *Advanced lectures on machine learning*. Springer, 2004, pp. 63–71.

[64] M. L. Stein, *Interpolation of spatial data: some theory for kriging*. Springer Science & Business Media, 2012.

[65] W. Oh and B. Lindquist, "Image thresholding by indicator kriging," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 7, pp. 590–602, 1999.

[66] T. Hengl, G. B. Heuvelink, and A. Stein, "A generic framework for spatial prediction of soil variables based on regression-kriging," *Geoderma*, vol. 120, no. 1-2, pp. 75–93, 2004.

[67] M. Seeger, C. Williams, and N. Lawrence, "Fast forward selection to speed up sparse gaussian process regression," in *Artificial Intelligence and Statistics 9*, no. EPFL-CONF-161318, 2003.

[68] D. J. MacKay, "Introduction to gaussian processes," *NATO ASI Series F Computer and Systems Sciences*, vol. 168, 1998.

[69] Y. Girdhar and G. Dudek, "Optimal online data sampling or how to hire the best secretaries," in *2009 Canadian Conference on Computer and Robot Vision*. IEEE, 2009, pp. 292–298.

[70] M. Bateni, M. Hajiaghayi, and M. Zadimoghaddam, "Submodular secretary problem and extensions," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*. Springer, 2010, pp. 39–52.

[71] S. M. Smith and S. E. Dunn, "The ocean voyager ii: An auv designed for coastal oceanography," in *Autonomous Underwater Vehicle Technology,*

1994. AUV'94., Proceedings of the 1994 Symposium on. IEEE, 1994, pp. 139–147.

[72] H. Kondo, T. Maki, T. Ura, and T. Sakamaki, "Observation of breakwaters and their rock mound by auv" tri-dog 1" at kamaishi bay," in *Oceans 2005-Europe*, vol. 1. IEEE, 2005, pp. 585–590.

[73] T. T. J. Prestero, "Verification of a six-degree of freedom simulation model for the remus autonomous underwater vehicle," Ph.D. dissertation, Massachusetts institute of technology, 2001.

[74] R. Panish, "Dynamic control capabilities and developments of the bluefin robotics auv fleet," in *Proceedings of the 16th International Symposium on Unmanned Untethered Submersible Technology (UUST)*, 2009, pp. 1–8.

[75] R. Camilli, C. M. Reddy, D. R. Yoerger, B. A. Van Mooy, M. V. Jakuba, J. C. Kinsey, C. P. McIntyre, S. P. Sylva, and J. V. Maloney, "Tracking hydrocarbon plume transport and biodegradation at deepwater horizon," *Science*, p. 1195223, 2010.

[76] T. I. Fossen, *Guidance and control of ocean vehicles.* John Wiley & Sons Inc, 1994.

[77] B. Jalving, "The ndre-auv flight control system," *IEEE Journal of Oceanic Engineering*, vol. 19, no. 4, pp. 497–501, 1994.

[78] M. Listak, G. Martin, D. Pugal, A. Aabloo, and M. Kruusmaa, "Design of a semiautonomous biomimetic underwater vehicle for environmental monitoring," in *Computational Intelligence in Robotics and Automation, 2005. CIRA 2005. Proceedings. 2005 IEEE International Symposium on.* IEEE, 2005, pp. 9–14.

[79] H. Yu, A. Shen, and L. Peng, "A new autonomous underwater robotic fish designed for water quality monitoring," in *Modelling, Identification & Control (ICMIC), 2012 Proceedings of International Conference on.* IEEE, 2012, pp. 561–566.

[80] R. Siddall and M. Kovač, "Launching the aquamav: bioinspired design for aerial–aquatic robotic platforms," *Bioinspiration & biomimetics*, vol. 9, no. 3, p. 031001, 2014.

[81] A. Ravalli, C. Rossi, and G. Marrazza, "Bio-inspired fish robot based on chemical sensors," *Sensors and Actuators B: Chemical*, vol. 239, pp. 325–329, 2017.

[82] P. Abbeel, V. Ganapathi, and A. Y. Ng, "Learning vehicular dynamics, with application to modeling helicopters," in *Advances in Neural Information Processing Systems*, 2006, pp. 1–8.

[83] M. Polkinghorne, G. Roberts, R. Burns, and D. Winwood, "The implementation of fixed rulebase fuzzy logic to the control of small surface ships," *Control Engineering Practice*, vol. 3, no. 3, pp. 321–328, 1995.

[84] R. Sutton, G. Roberts, and S. Dearden, "Design study of a fuzzy controller for ship roll stabilisation," *Electronics & Communication Engineering Journal*, vol. 1, no. 4, pp. 159–166, 1989.

[85] Y. Yang, C. Zhou, and J. Ren, "Model reference adaptive robust fuzzy control for ship steering autopilot with uncertain nonlinear systems," *Applied Soft Computing*, vol. 3, no. 4, pp. 305–316, 2003.

[86] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

[87] K. P. Tee and S. S. Ge, "Control of fully actuated ocean surface vessels using a class of feedforward approximators," *IEEE Transactions on Control Systems Technology*, vol. 14, no. 4, pp. 750–756, 2006.

[88] D. E. Rumelhart, G. E. Hinton, R. J. Williams *et al.*, "Learning representations by back-propagating errors," *Cognitive modeling*, vol. 5, no. 3, p. 1, 1988.

[89] K. Jarrett, K. Kavukcuoglu, Y. LeCun *et al.*, "What is the best multi-stage architecture for object recognition?" in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 2146–2153.

[90] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[91] A. Punjani and P. Abbeel, "Deep learning helicopter dynamics models," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 3223–3230.

[92] S. C. Chapra, *Surface water-quality modeling*. Waveland, 2008.

[93] T. G. Sanders and D. D. Adrian, "Sampling frequency for river quality monitoring," *Water Resources Research*, vol. 14, no. 4, pp. 569–576, 1978.

[94] J. C. Loftis and R. C. Ward, "Water quality monitoring some practical sampling frequency considerations," *Environmental Management*, vol. 4, no. 6, pp. 521–526, 1980.

[95] B. Chen, P. Pandey, and D. Pompili, "An adaptive sampling solution using autonomous underwater vehicles," *IFAC Proceedings Volumes*, vol. 45, no. 27, pp. 352–356, 2012.

[96] S. M. Petillo, "Autonomous & adaptive oceanographic feature tracking on board autonomous underwater vehicles," WOODS HOLE OCEANOGRAPHIC INSTITUTION MA, Tech. Rep., 2015.

[97] K. R. Newman, M.-H. Cormier, J. K. Weissel, N. W. Driscoll, M. Kastner, E. A. Solomon, G. Robertson, J. C. Hill, H. Singh, R. Camilli *et al.*, "Active methane venting observed at giant pockmarks along the us mid-atlantic shelf break," *Earth and Planetary Science Letters*, vol. 267, no. 1-2, pp. 341–352, 2008.

[98] B. Ewing, L. Hillier, M. C. Wendl, and P. Green, "Base-calling of automated sequencer traces usingphred. i. accuracy assessment," *Genome research*, vol. 8, no. 3, pp. 175–185, 1998.

[99] B. Ewing and P. Green, "Base-calling of automated sequencer traces using phred. ii. error probabilities," *Genome research*, vol. 8, no. 3, pp. 186–194, 1998.

[100] P. R. Kannel, S. Lee, Y.-S. Lee, S. R. Kanel, and S. P. Khan, "Application of water quality indices and dissolved oxygen as indicators for river water classification and urban impact assessment," *Environmental monitoring and assessment*, vol. 132, no. 1-3, pp. 93–110, 2007.

[101] J. C. Davis, "Minimal dissolved oxygen requirements of aquatic life with emphasis on canadian species: a review," *Journal of the Fisheries Board of Canada*, vol. 32, no. 12, pp. 2295–2332, 1975.

[102] C. J. Post, M. P. Cope, P. D. Gerard, N. M. Masto, J. R. Vine, R. Y. Stiglitz, J. O. Hallstrom, J. C. Newman, and E. A. Mikhailova, "Monitoring spatial and temporal variation of dissolved oxygen and water temperature in the savannah river using a sensor network," *Environmental monitoring and assessment*, vol. 190, no. 5, p. 272, 2018.

[103] C. Quast, E. Pruesse, P. Yilmaz, J. Gerken, T. Schweer, P. Yarza, J. Peplies, and F. O. Glöckner, "The silva ribosomal rna gene database project: improved data processing and web-based tools," *Nucleic acids research*, vol. 41, no. D1, pp. D590–D596, 2012.

[104] B. J. Callahan, P. J. McMurdie, M. J. Rosen, A. W. Han, A. J. A. Johnson, and S. P. Holmes, "Dada2: high-resolution sample inference from illumina amplicon data," *Nature methods*, vol. 13, no. 7, p. 581, 2016.

[105] M. J. Anderson and T. J. Willis, "Canonical Analysis of principal coordinates: a useful method of constrained ordination for ecology," *Ecology*, vol. 84, no. 2, pp. 511–525, 2003.

[106] D. A. Berry and B. Fristedt, "Bandit problems: sequential allocation of experiments (monographs on statistics and applied probability)," *London: Chapman and Hall*, vol. 5, pp. 71–87, 1985.

[107] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine learning*, vol. 47, no. 2-3, pp. 235–256, 2002.

[108] J. H. Holland *et al.*, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

[109] N. Srinivas, A. Krause, S. Kakade, and M. Seeger, "Gaussian process optimization in the bandit setting: no regret and experimental design," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*. Omnipress, 2010, pp. 1015–1022.

[110] A. Garivier and E. Moulines, "On upper-confidence bound policies for switching bandit problems," in *International Conference on Algorithmic Learning Theory*. Springer, 2011, pp. 174–188.

[111] B. Jürgens and W. Fork, *The fascination of the Voith-Schneider-Propeller: History and engineering*. Koehler, 2002.

[112] J. F. Kolen and S. C. Kremer, "Gradient flow in recurrent nets: The difficulty of learning longterm dependencies," 2001.

[113] E. Y. Hong and M. Chitre, "Roll control of an autonomous underwater vehicle using an internal rolling mass," in *Field and service robotics*. Springer, 2015, pp. 229–242.

[114] J. J. Moré, "The levenberg-marquardt algorithm: implementation and theory," in *Numerical analysis*. Springer, 1978, pp. 105–116.

[115] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 437–478.

[116] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 9–48.

List of Publications

- R. Mishra & M. Chitre (2016, November). Modelling of an AUV with Voith-Schneider vector thruster. In Autonomous Underwater Vehicles (AUV), 2016 IEEE/OES (pp. 355-359). IEEE. [Published]
- R. Mishra, M. Chitre & S. Swarup (2018, May). Online Informative Path Planning using Sparse Gaussian Processes. In OCEANS 2018, Kobe. IEEE. [Published]
- R. Mishra, M. Chitre & S. Swarup (2018, November). Informed Sampling and Adaptive Monitoring using Sparse Gaussian Processes. In Autonomous Underwater Vehicles (AUV), 2018 IEEE/OES. IEEE. [Published]
- R. Mishra, M. Chitre, & S. Swarup (2018, October). Multi-Robot Online Path Planning using Sparse Gaussian Processes. In IROS MPCP Workshop 2018, Madrid. IEEE. [Presented]
- B. Kalyan, R. Mishra, H. Vishnu & M. Chitre (2018, September). An Information Theoretic Approach to Polymetallic Nodule Exploration using AUVs. In Underwater Mining Conference 2018. [Presented]
- R. Mishra, A. Bandla, T. B. Koay, M. Chitre, & S. Swarup. Examining the Biological Relevance of the Environmental Fields estimated using Adaptive Monitoring Frameworks. [Accepted]
- R. Mishra, M. Chitre, & S. Swarup. Sparse Gaussian Process based Adaptive Point Planning for Single-Robot System. [In-Preparation]
- R. Mishra, T. B. Koay, M. Chitre, & S. Swarup. Using Kernel Information and Residual Variance for Multi-Robot Adaptive Exploration. [In-Preparation]

Appendix A

Sample preparation, DNA extraction and amplicon sequencing

Biomass was captured on 0.2 μm polycarbonate filters by passing water samples (approx. 0.2 L) using gentle vacuum (5 mmHg in). Polycarbonate filters were subsequently stored in deep freezers (-80° C) until DNA extraction. Genomic DNA was extracted from the polycarbonate filters using a DNA/RNA co-extraction kit (Norgen Biotek Corp.). The V4-V5 hypervariable regions of the 16SrDNA gene were then amplified using the 515Y (`GTGYCAGCMGCCGCGGTAA`) and 926R (`CCGYCAATTYMTTTRAGTTT`) primers. DNA quality and quantity were assessed using agarose gel electrophoresis and fluorescence spectroscopy (Qubit, Thermo Fischer). Samples were by Mr. Ooi Qi En while amplicon libraries were prepared by Mr. Chua Si Hao under the supervision of Dr. Aditya Bandla. Amplicon libraries were then tagged with Nextera (Illiumina Inc.) indices and adaptors using standard protocol and sequenced using the Miseq (Illumina; v3 chemistry) to generate 300 bp paired-end reads at the sequencing facility at the Singapore Center for Environmental Life Sciences Engineering (SCELSE).

16SrDNA amplicon sequence analysis

Raw demultiplexed reads were depleted off primers using cutadapt v2.3. Amplicon sequence variants were then resolved using the dada2 R package [104] using a customized in-house workflow. Briefly, low-quality tails of the paired-end reads were trimmed to fixed lengths and then those with maximum expected errors of ≤ 3 were retained for subsequent steps using the `filterAndTrim` function. Error profiles were generated using the `learnErrors` function with default settings.

Dereplicated reads were then denoised using a two-pass strategy designed to detect singletons using the dada function. Denoised reads were merged using the `mergePairs` function to yield sequence variants. Chimeras were then removed using the `removeBimeraDenovo` function with the pooled option. Chimera-free sequence variants were then filtered to retain only those variants with length 374 bp $\pm$ 10 bp. Finally, taxonomy was assigned using the `assignTaxonomy` function with the SILVA v132 database [103] with species labels being inferred using an exact sequence matching strategy implemented in the `addSpecies` function. Amplicon sequences were analyzed by Dr. Aditya Bandla.