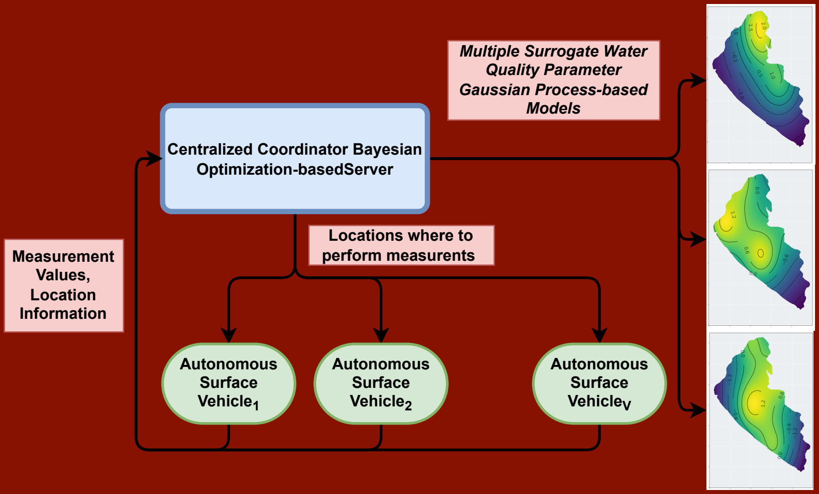
Tesis Doctoral
Doctorado Internacional en Ingeniería Electrónica

Intelligent Online Water Quality Monitoring through multiple Autonomous Surface Vehicles using the Bayesian Optimization Framework


Multiple Surrogate Water Quality Parameter Gaussian Process-based Models
Centralized Coordinator Bayesian Optimization-basedServer
Locations where to perform measurents
Measurement Values, Location Information
Autonomous Surface Vehicle$_1$
Autonomous Surface Vehicle$_2$
Autonomous Surface Vehicle$_V$

Autor: Federico Daniel Peralta Samaniego
Directores: Daniel Gutiérrez Reina
Sergio Luis Toral Marín

Ingeniería Electrónica
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2022

Tesis Doctoral
Doctorado Internacional en Ingeniería Electrónica

Intelligent Online Water Quality Monitoring through multiple Autonomous Surface Vehicles using the Bayesian Optimization Framework

Autor:
Federico Daniel Peralta Samaniego

Directores:
Daniel Gutiérrez Reina
Profesor Titular
Sergio Luis Toral Marín
Profesor Catedrático

Ingeniería Electrónica
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

2022

Tesis Doctoral: Intelligent Online Water Quality Monitoring through multiple Autonomous Surface Vehicles using the Bayesian Optimization Framework

Autor: Federico Daniel Peralta Samaniego
Directores: Daniel Gutiérrez Reina
Sergio Luis Toral Marín

El tribunal nombrado para juzgar la Tesis arriba indicada, compuesto por los siguientes doctores:

Presidente:

Vocales:

Secretario:

acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:

*A las personas
que no dudan en hacer un
esfuerzo para mejorar al mundo*

Agradecimientos

Escribir un libro de tesis doctoral no es una acción individual, pero el esfuerzo de un grupo de gente. De igual manera, mejorar el mundo también lo es. Agradezco a mi familia, en especial a mi padre y madre, por enseñarme esto. Agradezco a mis profesores Daniel Gutierrez Reina, Sergio Toral Marín, por demostrarme que esto es cierto, además de agradecerlos por la enseñanza academica y profesional que fue impartida por ellos desde el primer momento de haberlos conocido. Agradezco a mis amigos y compañeros, por motivarme a seguir creyendo que mejorar el mundo es un esfuerzo colectivo, en donde todos tenemos nuestro espacio y nuestras oportunidades de seguir evolucionando como civilización, por y para el mundo.

Federico Daniel Peralta Samaniego
Sevilla, 2022

Acknowledgements

The writing of a doctoral thesis book is not an individual action, but the effort of a group of people. Hopefully, so is improving the world. All of this, makes me want to thank my family, especially my father and mother, for teaching me this. Likewise, I want to thank my professors Daniel Gutierrez Reina, Sergio Toral Marín, for showing me that this is true, besides thanking them for the academic and professional education that was given from the first moment that I met them. I thank my friends and my fellow students for motivating me to keep believing that improving the world is a collective effort, where each one of us has their spot and opportunities to keep evolving as civilization, by and for the world. Ad astra per aspera.

Federico Daniel Peralta Samaniego
Sevilla, 2022

Resumen

La monitorización de los recursos hídricos y de su calidad es una actividad que está cobrando mayor importancia a lo largo de los años. Es necesario desarrollar sistemas de vigilancia eficientes e inteligentes aprovechando las tecnologías de vanguardia, como los agentes robóticos. La utilización de vehículos autónomos de superficie equipados con sensores de calidad del agua es un enfoque prometedor para medir continuamente los parámetros físico-químicos relacionados con la calidad del agua. Sin embargo, la mayoría de los trabajos actuales relacionados con este tema no tienen en cuenta la creciente disponibilidad y asequibilidad de estos sistemas. Por lo tanto, los enfoques actuales no se generalizan bien para tener en cuenta múltiples objetivos y la participación de múltiples agentes. La presente tesis proporciona un sistema de diseño que considera el uso de varios vehículos autónomos de superficie equipados con múltiples sensores de calidad del agua, de manera que se realiza un modelado en tiempo real de las masas de agua. Además, las mediciones se realizan considerando un sistema de partición de regiones Voronoi utilizando un marco de optimización bayesiana subyacente con múltiples objetivos. Los resultados muestran que el sistema puede obtener de forma robusta modelos muy precisos a pesar de la limitada información disponible y la restricción de autonomía energética de los vehículos. También se ha realizado una comparación con los enfoques basados en la cobertura y el patrullaje, y el sistema propuesto supera a estos enfoques fuera de línea en una media de 8,2% y 14,4%, en lo que respecta al error de modelado. El rendimiento de este enfoque también se ve reforzado por su robustez y escalabilidad en comparación con las misiones de monitorización fuera de línea.

Abstract

Monitoring water resources and their quality is an activity that is gaining more importance over the years. Efficient and intelligent monitoring systems must be developed taking advantage of cutting-edge technologies like robotic agents. The utilization of autonomous surface vehicles equipped with water quality sensors is a promising approach to continuously measure physico-chemical parameters related to water quality. However, most of the current related works do not acknowledge the increasing availability and affordability of these systems. Therefore, current approaches do not generalize well to account for multiple objectives and the involvement of multiple agents. The present work provides a bottom-up-designed system that considers the usage of multiple agents equipped with multiple water quality sensors so that online modeling of water bodies is done. Furthermore, the measurements are done considering a Voronoi Region Partitioning system using an underlying Bayesian optimization framework with multiple objectives. The results show that the system can robustly obtain very accurate models despite the limited available information and the constraint on vehicle energy autonomy. Comparisons with coverage and patrolling-based offline approaches have also been made and the proposed system outperforms these approaches on average by 8,2% and 14,4% with respect to the modeling error. The performance of this approach is also enhanced by its robustness and scalability compared to offline monitoring missions.

Short Contents

Contents

Notation

Acronyms

AAV	Autonomous Aerial Vehicle
AF	Acquisition Function
AGV	Autonomous Ground Vehicle
ASV	Autonomous Surface Vehicle
AUV	Autonomous Underwater Vehicle
AV	Autonomous Vehicle
BO	Bayesian optimization
CC	Centralized Coordinator
DRL	Deep Reinforcement Learning
EI	Expected Improvement
GA	Genetic Algorithm
GNC	Guidance, Navigation & Control
GP	Gaussian Process
GPP	Global Path Planning
GQI	Good Quality Indicator
IPP	Informative Path Planning
LPP	Local Path Planning
MFE	Multi-Function Estimation
MiP	Mission Planning
MOO	Multi-Objective Optimization
MRS	Multi-Robot System
NSGA-II	Non-dominated Sorting Genetic Algorithm II
PESMOC	Predictive Entropy Search for Multi-Objective BO with Constraints
PI	Probability of Improvement
POMDP	Partially Observable Markov Decision Process
PP	Path Planning

PSD	Positive Semidefinite
PSO	Particle Swarm Optimization
RBF	Radial Basis Function
RQ	Rational Quadratic
RRT*	Rapidly-Exploring Random Tree star
SE	Squared Exponential
SEI	Scaled Expected Improvement
SF	Shekel function
TG	Trajectory Generation
TSP	Traveling Salesman Problem
WQP	Water Quality Parameter

Notation

$\mathbb{R}^n$	Real Number Hyper-space
$\mathcal{M}$	Matrix Model Map
$\sigma(x)$	Standard Deviation value of a GP Regression
$p(y\|x)$	probability of y given x
$\alpha(\cdot)$	Acquisition Function
ℓ	Length-scale hyperaparameter for SE, RQ and Matérn Kernels
ε	Unknown noise value
$\exp\cdot$	Exponential function
$\Gamma(\nu)$	Gamma Function
λ	Proportion of length-scale value
$\log$	Logarithmic function
$\mathbb{E}$	Expected value
$\mathbb{V}[x]$	Normalizing function for SEI AF
μS/cm	microSiemens per centimeter
$\mu(x)$	Mean value of a GP Regression
$\mathcal{M}_{i,j}$	Element of the map model
$\mathcal{N}(\cdot,\cdot)$	Normal distribution function
$\mathcal{X}$	Feasible Search Space
$\phi(x)$	Probabilistic Density Function
$\Phi(Z)$	Cumulative Distribution Function
σ_n^2	Expected measurement noise variance
$\vec{v}$	Velocity of an ASV
ξ	Exploration bias hyperparameter
$_{posterior}$	subscript for posterior component of a GP
$_{prior}\ ,\ _{p}$	subscript for prior component of a GP
$\|\|\cdot\|\|$	Norm function
$\mathcal{L}$	Likelihood of a distribution
A	Location of peaks in a SF
c	Inverse powers of a peak in a SF
d	side length of $\mathcal{M}_{i,j}$
D	Distance traveled by an ASV
$d(\cdot)$	Euclidean distance between different x
e	Average current draw for ASV movement
$f(x)$	Unknown target Function
$\mathcal{F}(x)$	Objective Function
$\hat{f}(x)$	Approximation of target function
$\bar{f}_s(x)$	Mean value of the real data
x^B	Optimized (Maximum/Minimum) measurement location
$g_1(x)$	Feasibility check constraint function
I	Identity Matrix
K	Kernel Matrix
$k(x,x')$	kernel evaluation

K_v	Modified Bessel Function for value ν
L	Lower Cholesky Decomposition
σ_0	Linear Kernel expression
α	Scale-mixture hyperaparameter for RQ Kernel
ν	Smoothing factor hyperaparameter for Matérn Kernel
γ	Scale factor hyperaparameter for the linear Kernel
P	Order hyperaparameter for Lineal Kernel
p	Periodicity hyperaparameter for Sine Squared Kernel
l	Limited maximum distance to next measurement location
M	Number of measurement locations
m	Measurement ID
p_m	location of mth measurement
p_v	Current location of vth ASV
R^2	Coefficient of Determination
R_v	Monitoring Region of vth ASV
S	Number of available sensors
s	Sensor ID
T_{energy}	Total available energy
v	Vehicle ID
V	Number of Vehicles
x , $x*$	Location within the feasible space
x^* , p_{m+1}	Next measurement location
Z	Normalized optimum response
arg max	Argument that maximizes a function
arg min	Argument that minimizes a function
H+	potential of Hydrogen
mg/L	miligrams per Liter
MSE	Mean Square Error
mV	miliVolts
NTU	Nephelometric Turbidity Unit
ºC	Celsius

1 Introduction

When you explain a 'Why?' you have to be in some framework that you allow something to be true.

Richard Feynman, 1983

1.1 Background and Motivation

Water is a vital component of life as we know it. Every living being on Earth depends on it. The biological functions of water include supporting the cellular structure (life), and its chemical composition allows it to react easily with other molecules, generally creating more stable structures [1]. Water can be found almost everywhere on the surface of the Earth, and humans have been using it for personal consumption, as well as to create things and improve our quality of life. Water bodies are volumes of water that can be found on the surface of the Earth; this includes and is not limited to lakes, lagoons, salt lagoons, oceans, seas, rivers, basins, streams, and pools. A water body can also be used as a water reservoir, tourist site, fishing site, or even a wildlife bed, among other uses, so its health is of the highest importance. Limnology is a science study of aquatic systems that was born to fulfill the task of knowing the behavior of such water bodies and is based on observing and determining the chemical, physical, and biological characteristics of a water environment or ecosystem. The levels of hydrogen (pH), turbidity, oxygen, and temperature are chemical, physical, and biological variables that are generally the most common "Water Quality Parameters" (WQPs) that quantitatively express how healthy and fresh water bodies are. Collecting useful and standardized information about these variables helps improve decision making regarding the maintenance and cleaning of water bodies. Data gathering or collection is a related field of study that plans and organizes the acquisition and storage of information. In the case of water quality, it is generally necessary to continuously obtain new data due to the dynamics of water bodies and the continuous changes that they undergo. This is called monitoring, which describes a series of recurring tasks performed with the goal of obtaining and updating data and information.

Of course, maintaining the health and freshness of the waters must be a desirable practice. However, water does not always maintain an adequate level of health, which can be the product of contamination, misuse, or other bad practices. Therefore, water resources cannot provide the healthy and fresh state desired. The most common example found in nature is known as eutrophication, which is the fast unbalanced enrichment of nutrients, especially nitrogen and phosphorus, which causes alterations in water, its quality, and produces uncontrolled *algae* growth [2]. This algae is not a plant, but a type of toxic bacteria of the phylum of cyanobacteria. Cyanobacteria are toxic not only for humans but also for animals and plants as well, and they thrive in eutrophicated waters killing plants and animals as they can massively reproduce and overpopulate wet ecosystems.

Located between the cities of Ypacari, Aregua, and San Bernardino, the Ypacarai Lake (Paraguay) is a lake that can be used as an example (Figure 1.1). Officially, since 2013, its waters has blooms of cyanobacteria caused by eutrophication [3], especially during the hot months of summer. Alarms have since been raised and humans were no longer allowed to use the lake as a tourist spot due to the rising leves of cyanobacteria [4]. This issue has also occurred in other locations around the world, such as Lake Titicaca (Perú), the Mississippi-Atchafalaya River system (USA-Mexico) [2] and Mar Menor (Spain) [5].

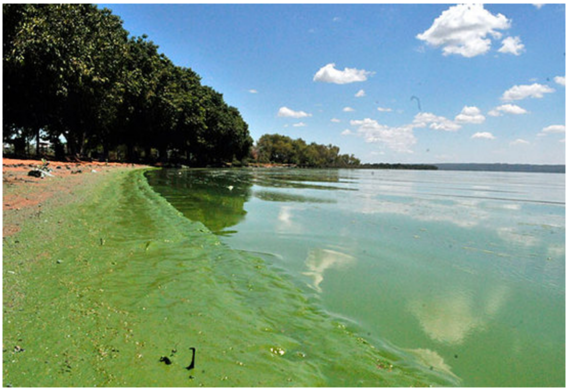


Figure 1.1 Blue-green algae proliferation in the Ypacarai Lake as pictured by a local news media [4].

It is crucial to eradicate algae blooms and restore and maintain the freshness and quality of their waters [6]. Eradicating this toxic algae in large-scale water body scenarios such as Ypacarai Lake (60 km^2) is not an easy task, but an obvious first step toward this is to monitor the state of WQPs. The use of sensors and water sampling can track and provide spatial distribution maps of WQPs. The resulting modeling helps and enables the development of efficient strategies to maintain WQPs in an adequate state, because water quality models will help locate contamination spots, as well as unnatural behaviors of the water, as water quality levels can be determined by considering the mentioned characteristics through chemical and physical indicators or parameters, monitoring the water quality of a particular water body is a systematic, useful, efficient, and standardized mission that is a field of study of particular interest.

Classical approaches for monitoring water quality consist mainly of manual measurement campaigns and measuring at fixed stations. These missions were designed as such due to the limited availability of technology during the early developing stages of monitoring, when WQPs were described, their levels were established, and, through mostly mechanical or manual processes, samples of water determined the WQP of a complete water body. Evidently, manual sampling requires humans to perform the sampling, which includes traveling to certain locations and back, and also to measure the WQP values. Monitoring stations, on the other hand, can automatically obtain the measures but are not designed to measure in generic locations mainly for maintenance and transport reasons. In some scenarios, such as monitoring the WQP of Ypacarai Lake, the missions mentioned expose the human operator to the dangers of toxic waters and unhealthy environments while obtaining samples, etc. They also do not provide a reliable global state of the measured WQP, an issue that is accentuated when applied to large-scale water resources, leading to wrong global estimates and affecting the desired outcome of decisions and strategies. Thus, these applied missions can present several disadvantages that can be addressed with the usage of Autonomous Surface Vehicles (ASVs).

Generally speaking, autonomous vehicles can be described as mobile robots (land, sea, and air) that deliberate the desired movement based solely on their sensors and guidance systems (i.e., they do not require any sort of driver or teleoperation control). Autonomous Aerial Vehicles (AAVs) are being used more and more because drones are economically affordable, easy to produce, transport, and deploy, and more importantly because drones serve as an abstract multidisciplinary research line, but nowadays flying AAVs need specific permissions that are not always easily available for research and development. Water-traveling AVs are also being developed, but they do not share all the same properties as AAVs, not even in between them, since there are several considerations that a vehicle must accomplish in order to travel in open-sea, lakes, through rivers (Autonomous Surface Vehicles) or underwater (Autonomous Underwater Vehicles, AUVs). However different AVs may seem, they all have one characteristic in common: autonomy.

Autonomy is defined as a way to describe systems that are: *capable of operating in the real-world environment without any form of external control for extended periods of time*. Therefore, AVs use their systems to properly obtain a desired result from a pre-defined mission, this mission can vary from transportation to monitoring services, and the AVs are, in no doubt, restricted by the vehicle's state. Some authors decide to trust the mission not to one, but to a group of vehicles [7, 8], conferring a higher level of security to the team, this decision affects the economic costs of the project but can greatly improve the expected output. Others decide to use only one vehicle but with more complex math models [9, 10], procedures and algorithms, which improve the output but heavily depend on the vehicle's capabilities of processing external (and internal) information. A more elaborate proposal includes both of these aims, taking advantage of the best of each method: the distributed information gathering and processing that uses complex mathematics [11]. This thesis uses ASVs as executors of complex algorithms aiming at a fully-autonomous monitoring of water quality parameters.

It has been observed that ASVs can significantly reduce recurrent mission costs if designed for automated WQP measurement [9]. In the current literature, a large number of environmental monitoring systems can be found that have ASVs as central executors, based on intelligent patrolling [10] to information path planning approaches [12]. ASVs, when used for monitoring, have significant advantages compared to manual or fixed monitoring systems. Compared to stations, ASVs have the advantage of being autonomously able to move to any location within an enclosed water body, while fixed stations can only measure in one place. Compared to manual sampling campaigns, if the environment is hazardous, that is, has high levels of toxicity, an ASV has no problem being in direct contact with the water, while human operators must be extra careful. Additionally, not every location can be reached with manned boats and ships, while ASVs can be designed with dimensions that can easily move on the surface of any water body.

ASVs have on-board systems that are responsible for their efficacy. Among these systems, there exists a system that can be thought of as the one that gives a major constraint, the energy system, which provides the energy autonomy to the vehicle [13]. This system controls and limits the amount of work that the ASV can do before stopping and is a major performance factor [7, 8]. There exist two practical ways to overcome this issue, one is to monitor the waters using a multi-robot system, i.e., two or more ASVs in charge of obtaining water quality measurements of the mentioned lake, and the second is to include a form of limitation of movement during the mission for each ASV The latter strategy needs to be included within the monitoring decision-making, i.e., the ASV needs to consider its energy in order to decide a measurement location that does not harm the efficiency of the system. This thesis develops a system that includes this constraint in terms of traveling distance limitation for each of the ASVs.

Regarding the usage of multiple ASVs, the improvement is given as long as two aspects are addressed: i) an intelligent system needs to distribute the work among the available ASVs and ii) for data gathering efficiency, redundancy must be avoided. Each ASV needs to explore the environment considering its current state, the state of other agents, and the state of the environment itself. Contrary to the latter statement, most proposed systems focus on naive or pure exploration of water bodies [14]. Only a few of them manage to intelligently monitor an aquatic environment [15, 16]. Furthermore, considering these few works, only a small quantity of them consider the measures and the resulting maps to decide on measurement locations [17, 18]. The vast majority of these systems only take into account a single parameter; for example, only one of the many WQ parameters is used for decision making, despite the fact that the system is usually prepared for performing measures of different parameters at the same time. In this thesis, the system considers multiple WQPs, as well as the current state of other intervening ASVs, to decide on measurement locations.

The consideration of multiple WQPs enhances monitoring because it has been observed that aimlessly patrolling or covering water environments needs and excessive, usually non-viable, amount of working time to achieve results. This is because many works that use ASVs focus on monitoring small scenarios and performing many WQP measurements on every zone of the water body. Of course, the larger the scale environment, the higher the cost, and the application of these techniques does not provide reliable water quality models that can be updated or used for future purposes. The cost of performing measurements is related to the costs of reaching the location. Consequently, most of the constraints are related to energy consumption and reaching locations.

The objective of monitoring is to obtain useful information about the state of the environment. Therefore, performing an offline monitoring mission, i.e., measuring locations without the aim of obtaining approximate models, is not a plausible mission for large-scale scenarios. In fact, many of the WQP monitoring systems do not account for measurement values to perform intelligent monitoring, i.e., data are not used in the process of determining next sampling locations. Thus, they are only stored for post-analysis, effectively performing inefficient monitoring. Using the obtained data can affect positively, because online monitoring is not only more efficient, but also possible using current technologies. The objective of this thesis is to design a system that can monitor(measure) multiple WQPs producing reliable WQP (surrogate) models. While offline methods are easy to implement and execute, the usage of one or multiple is not fully optimized. Therefore, this thesis focuses on a sequential decision-making strategy for monitoring.

This thesis bases its framework on Bayesian optimization (BO), which is a sequential decision-making strategy based on surrogate models and acquisition functions [19]. In the classical BO approach, Gaussian Processes (GPs) are used to approximate models of unknown black-box functions due to their suitable behavior using limited amount of data input [20]. The proposed approach in this thesis takes advantage of the fact that current WQ measuring systems are capable of quickly performing measurements of multiple WQPs at the same time. Consequently, the problem generalizes well to a multi-objective optimization setting, where the optimal modeling of multiple WPQs is the mission to be accomplished.

BO was chosen due to its model-based design, as it is useful to infer models of the WQ parameters of water bodies during the monitoring procedure. Additionally, BO provides an intelligent way of obtaining measurement locations if the Acquisition Function (AF), or function that evaluates the utility of measuring on a certain location, is appropriately selected towards reducing model uncertainty. In that sense, it is necessary to appropriately select the components. The surrogate model should be optimized to work for WQPs, while the acquisition function should be biased towards exploration to achieve monitoring. Moreover, the acquisition function should be adapted to account for the specific constraint of this problem, which limits the total travel distance of each ASV.

BO is a sequential procedure that starts with a prior surrogate model, and given a set of observations, a posterior model can be obtained through marginalization and conditioning. In this thesis, the surrogate models predict the behaviors of real water resource parameters using GPs, which are normally employed as surrogate models since they can determine stochastic behaviors producing stochastic responses. In the BO framework, the surrogate model supersedes a real behavior. Therefore, it is necessary that the surrogate model behaves in a similar way that of the real function to be modeled. In that sense, a natural way of designing appropriate surrogate models corresponds to designing covariation functions for the inputs, so that a relationship between inputs can be described. In this thesis, we describe a study on covariance functions and their usage for WQP modeling. Regarding the acquisition or utility functions, it is used to calculate a maximum over the current model so as to achieve maximum information benefit in the next sample. In the task at hand, this maximum value is used to determine the next water quality measurements performed by the ASV. Notice that the acquisition function is very important in the proposed monitoring work because it defines the movements of the ASV. Thus, classical acquisition functions used in BO cannot be directly applied to the monitoring problem due to the mobility restrictions of an ASV. In this work, we proposed several adaptations of the classical acquisition functions that are more suitable for monitoring tasks.

To undergo the process of eutrophication, water bodies must suffer from several atypical changes. These values can be obtained beforehand (or during eutrophication) in order to prevent (or recover from) water pollution and prevent algae blooms. In this sense, multiple WQP values need to be obtained from the current state of water bodies. Therefore, an important factor is the simultaneous measurement of several WQPs, so an efficient multi-function estimation system is proposed, as well as a Multi-Objective Optimization (MOO) strategy solved with GA. It is used to combine different utility functions because it is an optimal decision-making strategy whenever there are several objectives, as in this case study. Furthermore, in this work, we make use of different tools and techniques to make monitoring with an ASV possible, such as constraint handling and adaptations of AFs.

Note that data acquisition should be cooperative, i.e., several ASVs are available and need to jointly obtain measurements. Therefore, an intelligent multi-ASV system is proposed, where the ASVs are in charge of efficiently performing measurements using a Voronoi Partition system to effectively create online coverage regions. The Voronoi Partition or Voronoi Diagram (VD) is a mathematical expression that defines regions according to a set of generators [21]. If the positions of the ASVs serve as generators, each region has an ASV assigned to it, and the BO technique can be utilized seamlessly for each ASV to choose new measurement locations within each of the defined regions.

1.2 Scope

Monitoring WQPs can be interpreted as a recurrent task aimed at obtaining useful information about the target environment. In this thesis, a single mission is presented, as it will produce the current state of various WQPs of a water body using several ASVs for performing measurements. All ASVs are supposedly equipped with the same systems, which implies that these agents are homogeneous and will present the same behavior and constraints. For the selection of the different components, several experiments will be performed so that the system proposed in this thesis and its results are replicable and applicable to any water body.

1.3 Objectives

The previous sections served to summarize the background, general task, vehicle usage, and scope of this thesis. The General and Specific Objectives of this thesis are described below, with keywords highlighted in bold. The General Objective is as follows.

Design and implementation of a **multi-autonomous surface vehicle** monitoring system aimed at obtaining models of **multiple-water quality parameters** using a **Bayesian optimization** framework with a **Voronoi partitioning** system.

The specific objectives are designed to separate the general objective into specific components that the system needs to ensure efficacy, efficiency, and robustness. They are are described as follows.

- Propose a Bayesian optimization-based Monitoring Mission for a multi-vehicle system in unknown aquatic environments.
- Perform intelligent autonomous surface vehicle deployments in order to obtain a set of real or simulated approximated water quality parameter maps of a water body using Gaussian Processes as Surrogate Models, using a fleet of surface vehicles.
- Evaluate and compare the proposed method with common exploration/monitoring methods.

1.4 Thesis Contributions

The main contributions of this thesis can be summarized as follows.

- Application of Bayesian optimization for sequential decision making in a Water Quality Monitoring scenario, where the general BO equation is used to obtain measurement locations that will be useful for generating WQP models according to Gaussian Process stochastic models based on kernels or covariance functions.
- Hyperparameterization and comparison of classical kernel functions as main components of stochastic surrogate models for water quality distribution maps, validated through the usage of expected WQP behaviors in simulations using benchmark functions.
- Proposition of three adaptations of acquisition functions based on classical techniques for monitoring missions using an ASV with energy constraints, so that data can be acquired taking into account the limitations of mobility of the ASVs in charge of monitoring.
- A centralized Voronoi-based coordination system for multiple ASVs based on active region partitioning and data sharing, to create robust operations, seamless cooperation, and achieve better results in less time.
- A generalized multi-parameter measuring system for environmental monitoring based on Bayesian optimization with fusions of functions, considering centralized communication of the different ASVs, asynchronously.
- A multi-model acquisition system through multiple ASVs for online environmental monitoring. The system considers not only the existence of muti-WQPs but also multi-ASVs.
- Performance evaluation of the system considering the coefficient of determination of the obtained models with benchmark functions. Comparison of the proposed system with related monitoring works.

1.5 Document Organization

The rest of this book is organized according to a series of chapters that contain the following topics: Chapter 2 presents the related work and starts by defining the basic concepts of real components of the system according to the current state of the art, including the definitions of autonomous surface vehicles. This chapter describes the usage of other mathematical components of the system, such as Bayesian optimization and multi-objective optimization. Finally, current efforts for Monitoring Applications are also described in the chapter, addressing not only general water quality monitoring, but also monitoring research and implementation for highly eutrophized water like Ypacarai Lake.

Next, Chapter 3 describes the main architecture and each part and component of the system. This includes considerations, assumptions, and constraints according to the scope defined previously. Bayesian optimization is properly described, as well as its main components: Gaussian Processes and Acquisition Functions, their mathematical definitions, limitations, and solving methods/strategies. It includes examples of covariance functions used in the work. Voronoi regions are described in this chapter, as well as Multi-Function Estimation and Multi-Objective Optimization strategies. These components are fully described in this chapter, but their implementation is described in Chapter 4, where the usage of numerous pre-built systems is explained. The embbeded systems that implement the system are also described in this chapter.

The results are shown in Chapter 5. It includes complete results in detail by component, as well as summarized results. Several tables can be found that show the numerical performance values obtained, as well as a discussion of the results section. Comparison with other methods has also been done, the implementation and results are also found in this chapter, accompanied by a discussion of the obtained results. Finally, Chapter 6 concludes the work showing the learned insights, the overall results of the system, and future work.

2 Literature Review

Galaxies make stars, stars make worlds.

Neil deGrasse Tyson, 2020

2.1 Autonomous Vehicles

Every year, more robots perform their tasks and objectives with fewer margins of error; this is the result of countless research and advances in each of the components that generate what is known as an autonomous system. These components can be grouped into three main systems according to Liu et al. [22]: Navigation, Guidance and Control systems, known as GNC systems. These components answer some critical questions [23]: **"Where am I?"**, **"Where do I want to go and how?"**, and **"What can I do to get there?"**. Robots that can answer these questions can act on their own in order to accomplish their goals, and therefore be perceived to have a higher level of autonomy. Strategically, AVs research can be done aiming to answer these questions and develop systems that provide a solution to theses problems, specifically the ones that involve navigation, guidance and control. Searching for a solution to the latter issue, Control Systems have been in development since much before the birth of robotics, these control systems are used to determine appropriate control forces to be generated by the actuators, according to Liu et al. [22], and are already very well designed because of its necessity as a way to control any machine, including autonomous ones. Control Systems are in charge of the *movement*, which is a fairly well studied subject. A common situation is that these machines are in the need of constantly identifying their states and the states of their surroundings, this identification is accomplished by a Navigation System, which uses sensors and mathematical models to perform an *observation* of the situation and/or calculate the position of itself in respect of the environment (e.g. answers the question "Where am I?".) Research on *movement* and *observation* has been extensively developed in this field, but *planning* is a relatively new area and is the missing piece in the Autonomy puzzle [22]. Planning is what Guidance Systems are responsible for, and even though in the past, the useful signals produced by Navigation Systems and the ones that Control Systems receive did not have a computer as an intermediary, nowadays these systems work and cooperate seamlessly due to the implementation and use of Autonomous Systems, which are processing and producing the necessary signals. Clearly, current research on AVs is mainly focused on developing optimal deployment, intelligent motion, or smart guidance system, these terms correspond to *planning* [24, 25, 16].

2.1.1 Autonomous Systems

The deployment of vehicles are always mission-dependent; that is, when the deployment performance is defined, it always refers to the level of compliance of the mission with respect to the actual movements of the vehicles. This premise limits the vision of the average researcher to the development of *Path Planning* (PP) subsystems (or components). On the other hand, most of high performance Autonomous Systems also manage to include an appropriate Mission Planning (MiP) component that modifies the way of searching next goals or locations. Another way to improve the global effectiveness of the autonomous system is to focus on Trajectory Generation (TG) after the PP procedure. TG ensures that the control system positions the AV in known locations. Based on the work by Liu et al. [22], Autonomous Systems can be defined with a set of the 3 components introduced in this paragraph (Figure 2.1.) It is also worth noting that the system has an input signal “AV and Environment Information” and an output signal “Desired Velocity”, both of which were previously noted as output and input from the other systems that comprise a GNC or Autonomous System.

When it comes to the problem of developing GNC systems, one of the first decisions involves choosing the right method to solve global optimization problems. Once the domain of the environment is defined, the system must find the global minimum of a function $f(x)$, which can be a defined function or a function not analytically representable. In either way, a solution must be found and there are two general types of approaches: stochastic and deterministic [20]. Whenever research papers present a new way of solving a particular problem or a review of methods and techniques found in the state of the art, it is usual for the algorithms to be compared with algorithms of the same type (stochastic or deterministic), and this can be considered a good practice because it is assumed that the other approach may underperform with respect to the objective of the research. It is known that stochastic methods will identify the global minimum of a function in an infinite time with probability of one, and that deterministic methods ensure that with exact computations and arbitrary long times, an approximation of the global minimum will be found after a finite time.

All in all, Guidance Systems are designed and implemented, with a focus on the mission of the project under development. It is necessary to define and exemplify the components of the GSs in order to deeply comprehend the Guidance System design paradigm and, therefore, propose and produce better guidance systems.

- Mission Planning (MiP): referring to global high-level decisions that need to be made, Liu et al. [22] define that a MiP component should obtain all possible data (tasks, environment, operators, etc.) and define missions accordingly. These missions can

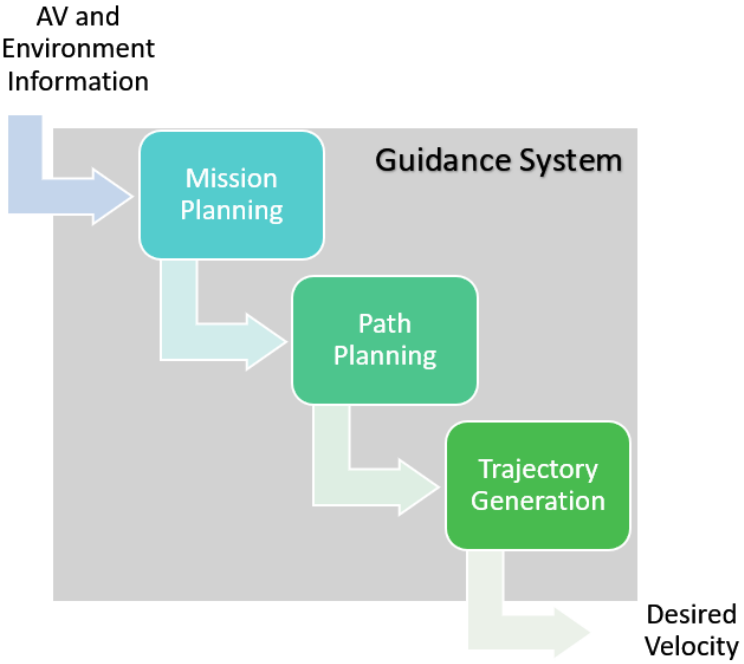


Figure 2.1 Autonomous System components: Mission Planning, Path Planning and Trajectory Generation.

be to obtain a target point or proceed to perform a task. Zhen et al. [26] describe MiP as a Waypoint (or target) Generation module for a multi-UAV system, Sampedro et al. [27] differ in these definitions, defining a Global Mission Planner (which acts like a normal MiP by providing waypoints) and an Agent Mission Planner that is in charge of executing the tasks made by the previous planner. In the survey by Mac et al. [28], mission planning can arguably be defined as the "cognition" step in the autonomous process they define.

MiP component is among the more abstract components of AVs not only because they need to provide context to the lower levels of Autonomous Systems but also they need to transform the abstract ideas of Mission and Vision Statements into feasible actions.

- Path Planning: The process of obtaining a valid route between two points is called Path Planning and is the crucial component of the GS due to the fact that to safely guide one vehicle to a point, a route must be planned. Zhou et al. [23] define this stage as the *route planning* procedure and also as a macro-modality constraint problem, and the vehicle is treated as a particle in the environment. Path Planning takes in all the data from NS and the MiP component, and obtains a series of waypoints to visit in order to reach to a desired location. Shin and Chae [29] elaborate a performance review of some of the best known path planning algorithms, such as A* or GA, using a small grid-based experimental configuration and a simulated discrete particle as vehicle. Other works elaborate similar performance reviews but take into account models of the vehicle dynamics to test the algorithms, for example, Chaari et al. [30] using a simulated Autonomous Ground Vehicle (AGV) while Zammit and Van Kampen [31] used a simulated AAV as vehicle for comparison. These works define various types of path planning methods, they can be Global Path Planning (GPP) or Local Path Planning (LPP), the first approach takes into account only information found in the past (be it from the environment, or even from the vehicle itself), it can be considered as a first step towards a high-performance planning or a previous step for LPP, which is the approach that considers continuously updated information and obtains routes that can change through time if required. Though Path Planning is sufficient to produce valid routes for vehicles, in some occasions additional computation is needed in order to provide safe routes.

- Trajectory Generation: Whenever a more complex scenario is presented, this component is necessary to ensure that the vehicle follows the planned or projected path. More complex scenario means more complex dynamics and/or more rules to be followed to reach a destination. Panden et al. [32] denote that the TG component is suitable whenever the environment or constraints are dynamic, this is true in most real-life scenarios since environment hazards such as winds, waves, etc. are present.

In spite of the vast amount of definition for these systems, all researchers in this field define these systems in the same way as stated in this thesis, differing only in word selection. For the sake of clarity, in this thesis, the terms from different works will be adapted to the previous definition whenever necessary.

2.1.2 Autonomy

Figure 2.2 is designed around the definition of *autonomy*, taking into account the components of the Guidance Systems. The less the AV decides its actions, the lower the autonomy level, therefore, the less it comprises all possible components of the Guidance Systems. However, the process shown in Figure 2.1 can be misleading because it shows that each component depends on previous ones, and while this is true, if a work attempts to work on more general components, it is assumed that more specific components are also developed, but this is not true. Some authors focus their work on a specific component like Path Planning and not Trajectory Generation, while others develop more components to ensure a higher level of autonomy. To emphasize the definition of *autonomy*, it is necessary to differentiate this term from *automation*, the first refers to the capacity to deliberate a course of action by itself, while the latter most often implies the process of performing physical or logical actions without any help.

Zhou et al. [23] proposed eight levels of automation, in which each designed Guidance System can be assigned to each of the levels. However, it must be noted that these levels include cooperation between vehicles (i.e.: sharing information, etc.) from level 6+ and it is safe to say that a vehicle can be fully autonomous while not being able to cooperate with others. Abraham et al. [33] define seven levels of automation for AVs and the levels differ in definition but correlate in the desired output with the previous cited work. Wickens [34]

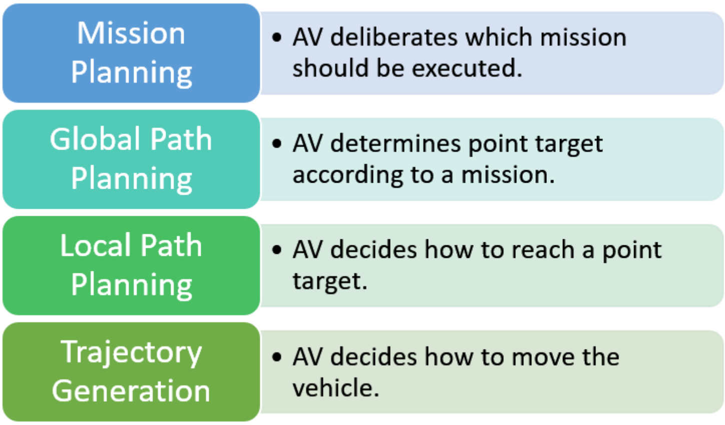

Mission Planning
• AV deliberates which mission should be executed.
Global Path Planning
• AV determines point target according to a mission.
Local Path Planning
• AV decides how to reach a point target.
Trajectory Generation
• AV decides how to move the vehicle.

Figure 2.2 Automation capabilities.

made a third automation definition that is based on stages rather than levels and defines 4 global situations: Attention filtering, Diagnosis, Action Choice, and Action Execution. These definitions were consolidated in Table 2.1 to have the least possible number of levels of Guidance Systems but sufficient to differentiate the perceived *quality* of automation.

Table 2.1 Automation Level.

Automation Level	General Definition	Level (Zhou et al. [23])	Level (Abraham et al. [33])	Stage (Wickens [34])
1	Human Guidance	1-2	1	1
2	LPP	3	2-4	2
3	GPP & LPP	4-5	5	-
4	PP & TG	6-7	6	3
5	MiP, PP & TG	8	7	4

Guidance Systems design must exploit the vehicle's capabilities, properties and applications, i.e.: for maritime rescues purposes, the vehicle must prioritize time and power management, for unknown terrain mapping, the vehicle must cover the entire environment in detail, and for precision agriculture, the vehicle must carefully select the paths to drive. In Table 2.2, several examples of work in this field are shown. The approach that relates to this thesis is the work of Morere et al. [19], where an AAV deliberates the next goal position according to the level of uncertainty of the environment and aims to decrease the uncertainty in high-gradient zones.

Table 2.2 Automation level of some vehicles.

Guidance System	Perceived Automation Level	Main Focus	Vehicle Type
Morere et al. [19]	4	MiP, PP, TG	AAV
Tokekar et al. [35]	4	GPP, LPP	AGV & AAV
Liu et al. [36]	3	LPP	ASV

2.2 Global Path Planning

Some applications require redundancy, dangerous activities, coverage, and emergence. These are defined as applications that can be achieved by a *Multi-Robot System* (MRS), which is more robust, flexible, and scalable than a single robot system (isolated vehicles) according to Bayindir [37]. MRS have a special place in the field of mobile robotics because they deal with the organization of *many*, organization that should be comprehensible from a general point of view. The last statement by no means implies a general supervisor or coordinator, MRS can be centralized or decentralized [37]. They are capable of providing a solution whether a central unit is providing the tasks or not.

Many models of social robots and AVs are based on natural behaviors or the human interpretations of them, these models are focused on defining rules and instructions for a single vehicle, they can intentionally cooperate or compete in a social environment but in the first approach of development, then on a second stage of development, the rules are expanded and include interactions between vehicles in order to achieve the social requirements that are in the scope of the research. Addressing, for example, the coverage problem, it is common to have a vehicle that is capable to cover a pre-defined area and later develop the communication and cooperation system in order to divide the areas to cover when a MRS is present. Table 2.3 shows a number of examples of MRS, grouped by application. A *coverage* application aims to monitor a geographical area, *swarming* is the action of collaborating between simple robots to accomplish a goal, *transport* robots are vehicles that coherently act and communicate to move one large object.

From the examples shown in Table 2.3, the only application that cannot be accomplished by a single vehicle is swarming, there are numerous works that aim to patrol, transport or even cover an area, but for evident reasons, they take more time and more prone to fail the task at hand. In any case, it is important to consider the research that fits the scope of this thesis, which includes all *coverage* application researches and works. The thesis by Kapoutsis [38] provides a multi-ASV approach to cover an unknown environment. It focuses on dividing the environment using a cyclic algorithm taking into account the starting position of the vehicles.

While enclosed environments can be optimally covered in a finite time, open environments are difficult to cover because in some of the worst-case scenarios the supposed

Table 2.3 Examples of coverage applications.

Application	Vehicle Type	Main Alg.	References
Coverage			
	AGV	Flocking	[25]
	ASV	DARP	[38]
Swarming			
	AAV	Forward, Coherence, Avoidance	[39]
Transport			
	AGV	Formation, Local and Global PP	[40]

previous information can lead to dead-ends or local minimum. This problem can be avoided using the Kapoutsis method proposed [38] or by continually updating the information about the environment, more specifically, updating the *posterior* model of the environment with the *prior* model and a new *evidence* taken from the environment. Although this method can be more computationally intense than the first one, it provides better coverage whenever the information of the environment is more important than the coverage of the environment itself.

This thesis describes a method that aims to obtain approximate models of unknown environments in a shorter total time than current methods, admitting uncertainty. This will be accomplished by a group of Autonomous Surface Vehicles that distributively work to obtain information about the environment and mathematical models that take into account uncertainty and probability. The automation level will reach a level of 5 (Table 2.1), considering that this thesis addressed the Mission Planning component, and uses multiple systems to perform Path Planning and Trajectory Generation. This is one of the main contributions of this thesis, as initially described in the previous chapter.

2.3 Modeling and monitoring applications

Regarding GPP for monitoring, there exist conventional monitoring methods that have been implemented due to their simplicity, like the lawnmower method. It consists of sweeping the space using a predefined distance between measurement locations (in the case of discrete measuring) and a preestablished route morphology. In general, using these techniques yield a slow but detailed monitoring. Therefore, these are the preferred methods when there are neither time-related restrictions nor expensive exploration costs. L-Cover, T-Cover, and Z-cover [16] are some common methods that fall into the lawnmower category. These methods are extremely useful in situations where the work or search space is very narrow, such as with rivers and streams; however, they have less impact observational wise as the water body widens due to energy availability concerns. In addition, these methods are not designed to do online learning or monitoring. Various coverage, patrolling, and monitoring works can be found that use unmanned vehicles to accomplish the task of offline monitoring. Some of them [41, 42] develop control systems for ASVs and are tested in real life scenarios.

Monitoring applications with the usage of autonomous vehicles is an active research field [43] [44]. Both aerial and aquatic vehicles (surface and underwater) equipped with environment sensors have been proposed to measure parameters. This is the case of the work in [45] that proposes a Q-learning fuzzy logic for flying ad-hoc networks of aerial vehicles. Fuzzy logic is a type of algorithm that accounts for unknown behaviors and results for actions with probabilistic overlapping outputs. Another work [46] develops path planning strategies for aerial vehicles for autonomous search and target location in a risky environment, while optimizing both search and survival through the selection of a balanced Pareto-point front and using customized reward functions. A recent study [47] designs a Multi-Objective Optimization for Drone Delivery that manages to minimize delivery cost using the ε-constraint method and the percentage of unsuccessful delivered packages while maximizing on-time delivery reward. The multi-objective strategy is also proposed in [48], where a coverage problem is addressed using multiple aerial vehicles for efficient communication networks. In relation to monitoring, in [49], the authors describe a path generation algorithm that minimizes the time of rescue action using a Genetic Algorithm (GA) technique. The work considers several dynamic and time constraints in order to provide efficient routes for reaching a target destination. One approach [50] considers multiple aerial vehicles for agricultural monitoring using a distributed swarm control technique. Definitely, multi-objective problems as well as multi-agent systems have been proposed for accomplishing monitoring missions with autonomous vehicles.

An in-depth survey focusing on robot path planning [51] classified some of the referred works and others into strategies and approaches that completed the coverage mission. The cited applications in the survey range from mapping and surveillance to coastal coverage. It is important to note that monitoring tasks are not exclusively based on the usage of surface vehicles, since multiple works use AAVs to perform the same type of mission. In [52], the authors presented an algorithm that monitors an environment using an energy-constrained AAV that covers an area together with an AGV. There exists a related work that presents energy-related constraints.

A specific GPP implementation related to monitoring corresponds to Informative Path Planning (IPP). IPP are algorithms designed to a path that maximizes obtain the gain between a starting and a final location [12], and they are very useful when the goals are already defined. IPP algorithms have been used for surveying underwater algae farms [12], using Gaussian Processes as surrogate models and information gain-based functions to select waypoints or locations. Another IPP implementation [53] uses AAVs to map unknown environments via ray-casting for mazes and building explorations using a Rapidly-Exploring Random Tree star (RRT*) inspired algorithm. RRT* is a good path planner since it is a fast a reliable algorithm [54]. Another example that uses ASVs for environmental monitoring can be found in [24], where orienteering-based approaches are proposed.

Some works have focused on developing monitoring systems for Ypacarai Lake [14]. For example, the International Hydroinformation Center (CIH)[55] of Itaipú focuses on continuously monitoring fixed locations within the Lake as well as its main sources of inflow and outflow. Furthermore, the Multidisciplinary Technology Research Center provides results of sampling measurements carried out in 14 different locations. Considering the usage of ASVs, in [14], the authors propose algorithms that explore the lake with an ASV using evolutionary algorithms based on the Traveling Salesman Problem (TSP), which provides a global path that manages to explore the surface of the Ypacarai Lake. The main algorithm in the mentioned work evolves the visiting order of 60 waypoints located on the shore of the lake so that coverage is maximized considering Eulerian Circuits. The mentioned work was further improved and, compared to Hamiltonian circuits in [15], the evolutionary-based approach manages to obtain the best visiting order so that the maximum possible area coverage is performed. Additional improvements were proposed in [56], where two objectives (Exploration and Exploitation) are accomplished but not simultaneously. More recently, in [10], a Deep Reinforcement Learning (DRL) approach was designed so that a single ASV patrolled the lake. Homogeneus and non-homogeneous patrolling were designed and solved through Deep Learning Techniques in other works [11]. The authors compared their proposed algorithm with GA techniques in [57]. Another related work used a Particle Swarm Optimization-based (PSO) algorithm for monitoring and exploration [58], where multiple vehicles were considered and an optimization is done through the vehicles. Their next work [59] considers not only the PSO algorithm but also the results of a Gaussian Process regression to search for peaks and valleys of the measured WQP. Furthermore, they compared PSO algorithms in their most recent work [60], which is accompanied by a work in progress in [61], which applies the most recent variations of PSO to the monitoring problem for peaks (or maximum) optimization. The only multi-objective related work performed for Yparacai Lake corresponds to [62], where a set of ASVs are in charge of patrolling the lake considering multiple objective maps.

Regarding BO-based approaches, recent monitoring proposals using AAV include [19, 63, 64]. In [19], the authors use sequential BO to obtain a height map using an AAV, where the decision is made using a Partially Observable Markov Decision Process (POMDP). The results obtained indicate that the Bayesian approach is useful whenever the number of measurements is limited. In [63], continuous 3D trajectories are generated to map an area using Gaussian Processes. Similarly, a path planning model is proposed in [64], where an interactive multiple model algorithm is used to update the belief space. Large-scale pollution and luminosity monitoring simulations were performed in [65] using continuous BO approaches, obtaining a significant error reduction compared to other methods. A Bayesian Exploration-Exploitation approach for online planning was presented in [66], using a POMDP utility function. A recent work [67] shows the development of save navigation procedures for mobile robots using BO to predict high movement vibrations in a defined space. A summary of monitoring approaches can be found in Table 2.4, including the monitoring algorithm and whether it is online (uses new information to decide on the next actions of the agents).

Although the mentioned works are promising, the proposed method presents some important advantages with respect to the application of real monitoring task applications. The main benefits include: i) obtaining a mathematical surrogate model of several real water quality parameter models, which can be easily used for future purposes, ii) intelligent determination of measuring locations using adapted acquisition functions that consider the mobility restrictions of autonomous vehicles, and iii) an important reduction of the exploration cost of a water resource, which is crucial for efficient large-scale monitoring.

Table 2.4 Summary of path planning approaches for monitoring, coverage and autonomous decisions.

Ref. (Year)	Specific Objective	Monitoring Algorithm	Online
[19] (2017)	Height reconstruction	BO path planning	☑
[16] (2019)	Coverage	Lawnmower Z-,T- Cover	□
[47] (2019)	Drone Delivery	Multi-Objective O.	□
[49] (2020)	Rescue action	GA for LPP	□
[56] (2019)	WQP Monitoring	TSP-based GA 2x planning	☑
[63] (2020)	Environment reconstruction	GP based path planning	☑
[58] (2021)	WQP Monitoring	PSO monitoring	☑
[62] (2022)	Water patrolling	GA-based MOO	☑

To this point, the contributions of this thesis mentioned in the previous chapter are clearer and are identified as the current gaps in the state-of-the-art that needed to be addressed.

Furthermore, Multi-Objective Optimization systems using fixed sensor stations have recently been proposed for different environmental missions, such as monitoring ventilation system [68] or even water quality monitoring [69]. Both mentioned works pursue the objective of optimal positioning sensors in fixed locations. While the former [68] uses a GA for obtaining the optimal locations according to several functions, the latter [69] utilizes Transinformation Entropy, which measures mutual information, to be minimized by an enhanced GA for Multi-objective Optimization. These methods consider multiple objectives and solve the problem by taking into account each objective independently. For coverage problems taking into account multiple vehicles, in [70], the authors propose a coupled weighted multifunction to decide the positions of UAVs for optimal coverage deployments. These works are based on the selection of locations on GA.

Multi-Objective Bayesian optimization has been applied to many optimization works [71, 72] in several research fields. The sequential BO proposed in existing work is useful and can be adapted to accomplish monitoring tasks. For example, in [71], a Predictive Entropy Search for Multi-Objective BO with Constraints (PESMOC) is proposed using a 4-step process. This method takes into account the Pareto-front generated by a set of predicted mean distributions. The general PES AF is used for each objective function. In addition, the authors propose both **coupled** and **decoupled** methods to fuse the different AFs. The first fusion technique consists of adding the different AFs in order to obtain a global AF. However, the second method, which consists of selecting the maximum of the different AFs, has been proven to generate better results. On the other hand, in [72], the authors used a two-stage BO for global optimization based on the evaluation of multiple AFs using the same surrogate model and a selection criterion, obtaining a reward-based AF selection system. However, in the mentioned work [72], the system is capable of evaluating the objective function without accounting for the distance between the locations of the measurements, so that, ultimately, multiple AFs (one for each objective) were used only to increase the total number of measurements.

In [73], multiple Dubins vehicles (capable of following only soft curved paths, i.e., no hard turns or rotations around their axes) were used to perform coverage in a 200 km^2 lake. The vehicles can be heterogeneous as well, implying different physics, dynamics, and control for each agent. For example, in [74], the authors developed a coordinated system using an AUV with an ASV to detect pollution in water environments. Another study [75] proposes a multi-robot path planning algorithm whenever connectivity constraints are present. The work in [76] shows that AAVs can be used to inspect rice farms using PSO algorithms. Next, the work in [8] shows a dynamic graph-based formulation for underwater vehicles so that ocean temperature can be measured with multiple vehicles. Other underwater vehicles usages include a work for monitoring underwater algae bloom [21]. Information entropy can also be considered for monitoring, in the case of the work in [77], an IPP algorithm was developed that considered entropy to accomplish a mission of plant phenotyping. Table 2.5 shows a brief summary of monitoring approaches that use multiple vehicles to achieve their objectives.

Table 2.5 Brief summary of monitoring approaches using multiple autonomous vehicles.

Ref. (Year)	Specific Objective	Monitoring Algorithm	Vehicle
[21] (2017)	Algae Bloom Monitoring	Decentralized adaptive informative sampling	Underwater Vehicle
[8] (2018)	Ocean temperature environmental monitoring	Dynamic graph-based routing	Underwater Vehicle
[77] (2018)	Plant phenotyping	Entropy-based Informative Path P.	Ground Vehicle
[73] (2018)	Complete coverage of known environments	Traveling salesman problem k-TSP -formulation	ASV
[75] (2019)	Exploration under Connectivity Constraints	Bipartite Graph Formulation	Generic Robot

It is not very common to fin a work that uses multiple agents focusing on multiple objectives, simultaneously. Moreover, if the objective is for monitoring, the number of works is more limited. Table 2.6 shows some works that are designed for monitoring that considers multi-agent systems or multi-objective problems. It can be seen that this is not a common approach to address both simultaneously, in fact, the last two works mentioned are works that were developed for the specific problem of Ypacarai Lake water quality monitoring.

Finally, the proposed method differentiates from the related works as it is the first, to the best knowledge of the author, that addresses the multi-objective data acquisition for environmental monitoring using multiple ASVs. The system is capable of minimizing the uncertainty, which minimizes the surrogate error, of multiple water quality models simultaneously. Furthermore, we manage to implement Multi Objective Optimization

Table 2.6 Related monitoring works indicating whether the works consider or not multiple agents and multiple objectives.

Ref. (Year)	Path Planning Approach	M-Agent	M-Obj.
[73] (2018)	Dubins Coverage with Area/Route Clustering	☑	□
[78] (2018)	Human Operated	□	☑
[75] (2019)	Multi-Robot Informative Path Planning	☑	□
[45] (2020)	Q-learning design	☑	□
[63] (2020)	Informative 3D PP for Terrain Mapping	□	□
[58] (2021)	Gaussian Process-based PSO	☑	☑
[62] (2022)	GA-based MOO	☑	☑

approaches for environmental monitoring, whereas recent algorithms are not tested in this scenario. Compared to fixed station works, the proposed method can efficiently select measurement locations when the functions are unknown, whereas fixed monitoring station methods are based on monitoring environments with a priori information. With respect to other ASV-based environmental monitoring systems, the proposed system is actively learning, exploring the environment taking into account multiple information, due to the fact that it is based on Bayesian optimization for multiple functions. The related works included in this section provide theoretical bases for the design and implementation of our proposed method. Although most of the works shown used GP as surrogate models, none of them implement BO procedures to obtain information. This work also takes into account energy constraints, so that the performance evaluations can provide insight for real-life experiments.

Our method also differentiates from the related work, as it relies on the contribution of several input data, rather than only one objective. Generally, the underlying mission is only exploration [14, 16]. But this work simultaneously performs both monitoring and model obtaining. Our method quickly performs active learning, obtaining approximate models of various WQ parameters simultaneously. In the next chapters, the problem is stated, and the assumptions and the different components are appropriately defined.

3 Methodology

Work is never over.

Daft Punk, 2001

In order to achieve the desired level of autonomy mentioned, the monitoring system must decide the location of the measurements according to the data already available. This process can be done planning one step at a time or multiple steps ahead. Although offline methods will perform better when the available information meets a certain level of accuracy and relevancy, the online method is preferred when the system is first introduced into a new environment. Moreover, the natural starting point is to opt for an online monitoring system. In that sense, several key elements of both the system and the environment must be properly described.

In this setting, the system sequentially chooses new “evaluation points” regarding the current state of the environment, updates its belief according to newly collected data, and repeats the process until a certain criterion is met or a constraint is violated. Figure 3.1 shows a simple flow chart that shows the abstract behavior of the system. This online approach closely resembles the BO framework, which can also be related to Active Learning. Referring to the work of Morere et al. [19], Bayesian optimization is a sequential method that can ensure lowering the uncertainty levels of the measured or evaluated function. BO considers a surrogate model and a utility function to select measurement locations in the real world accordingly [79]. This surrogate model can, of course, be a good or bad approximation, but eventually the evidence will provide a confidence measure on whether or not the model fits the real environment. Selecting an appropriate model can be tough and time consuming, but also optimal for applications where the environment is unknown or only has a limited amount of prior information available. To organize the acquired data and present the BO framework, some preliminary definitions are considered.

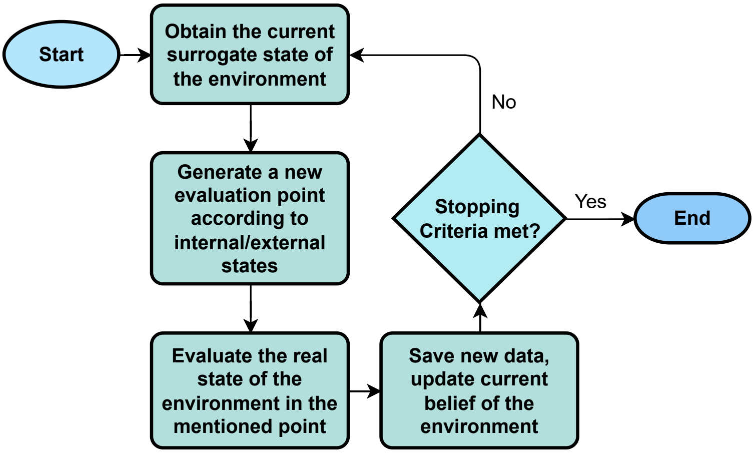

Start
Obtain the current surrogate state of the environment
Generate a new evaluation point according to internal/external states
Evaluate the real state of the environment in the mentioned point
Save new data, update current belief of the environment
Stopping Criteria met?
No
Yes
End

Figure 3.1 Abstract general flowchart of the steps that needs to be done in the monitoring mission.

3.1 Preliminaries

The monitoring system designed in this thesis has the objective of efficiently obtaining water quality models of a given water body. A water body is defined as any significant accumulation of water on the surface of the Earth and can be a reference to seas, salt lagoons, freshwater lakes, rivers, or small lagoons. In this context, each of many water quality parameters is mathematically defined as an unknown black box target function $f(x)$, where the argument x corresponds to a location in the search space (water body). Evidently, each target function is unknown to the observer, and additionally, their true behaviors could never be obtained. Consequently, these functions are replaced by surrogate models designed to approximate the real behaviors. The values of the target functions are available to the system, but the system performs noisy reads such that:

$$\hat{f}(x) = f(x) + \varepsilon \tag{3.1}$$

with ε as the noise bounded to the measurement. With that, the expected output of the system will be based on these noisy measurements rather than on the true values. Additionally, for computation purposes, the real model will be replaced with a surrogate model that is set to behave like the real one. A comprehensive definition of the surrogate model is available in later sections, but, loosely speaking, the surrogate model can be thought of as a model of $\hat{f}(x)\,\forall\, x \in \mathcal{X}$.

3.1.1 Map model and measurement locations.

$\mathcal{X}$ is a subspace of the $\mathbb{R}^n$ space. It is expected that x is a geodetic coordinate, that is, a latitude, longitude, and altitude data, because the agents in charge of performing the measurements are on the surface of a water body. However, since the agents can only travel on the surface, the space of actions reduces to Latitude and Longitude. Moreover, since the dimensions of Earth will always be much larger than the dimensions of the water body to be monitored, the space $\mathcal{X}$ reduces to a 2D projection of the water body. Then, the modeled map can be described in more detail, since only the rectangle that encloses the water body is needed. This rectangle can be thought of as a matrix $\mathcal{M}$, which can be called a map model. Therefore, the monitoring region is considered as a matrix $\mathcal{M}$ with dimensions of $m \times n$, i.e, a monochrome image with pixel dimensions $m \times n$, i.e., each pixel corresponds to a square of side d that has a black or white color that represents the occupancy state of the square. This square represents a compromise solution in which the environment is sufficiently detailed but large enough to allow large-scale modeling. Each element of the map model $\mathcal{M}_{i,j}$ has a value defined by an occupancy state (1 if the element corresponds to a occupied square space $d \times d$ in the space, 0 if not).

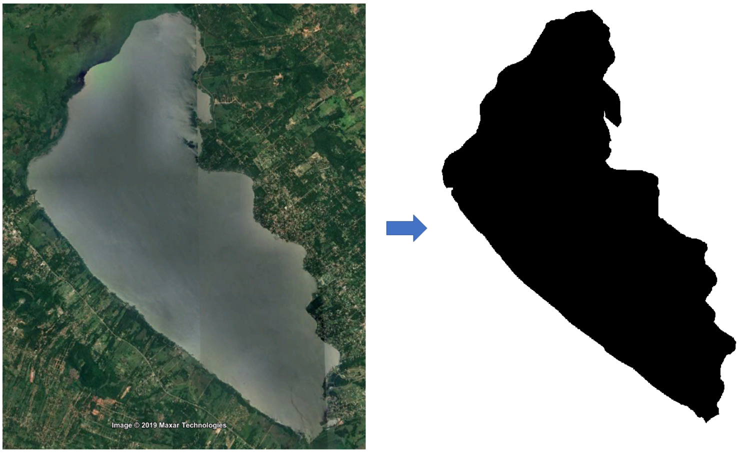


Figure 3.2 Example of a map model. Originally, the real image is drawn from a Satellite Resource. Then, it is translated to a binary image according to the navigability inside a square of a defined side d.

Figure 3.2 shows an example of the map model of a water body used in this work; in the image, the color white represents obstacles (including prohibited zones, natural obstacles, land, etc.), while the color black represents a square of free obstacles. Definitely, the environment studied is discrete due to the usage of the map model $\mathcal{M}$, so that a vehicle can only be located in a position related to an element of the map. Finally, the set of navigable elements is defined as follows.

$$\mathcal{X} \subseteq \mathcal{M} \subseteq \mathbb{R}^2 \tag{3.2}$$

A general way to represent the map is to also include every non-navigable location in the $\mathcal{X}$ with the addition of a constraint function that yields the occupancy state. The constraint function will satisfy the following inequality for all feasible locations: $g_1(x) \leq 0$. Additionally, any location outside of the map model $\mathcal{M}$ will yield undefined values, i.e., the edges of the map model define the boundaries of the region in which an agent can be found. However, in what follows, unless defined otherwise, a generic location $x \in \mathcal{X}$ will be considered to satisfy the constraint.

An important consideration is that the studied water environments are always non-convex sets. The convexity of a set can be described by graphical observations. If, in the space of actions (map model), there exists any pair of feasible locations that cannot be connected (through a segment line) without selecting unfeasible locations, the set has a non-convex property. Regarding the fact that any water body will generally have a non-standard shape, it is better to consider that the water body will always be a non-convex set, including the example shown in Figure 3.2. Consequently, local path planners are needed in this scenario to ensure that each available ASV can travel safely from one location to another.

3.1.2 Water Quality Parameter Model Maps. Measurement values

Observing Eq. (3.1), x, as well as $f(x)$, are now mathematically defined, but $f(x)$ and the measurements $\hat{f}(x)$ must also be defined in terms of the monitoring application, as x was in the preceding paragraphs. Environmental water quality is evaluated according to a set of physicochemical and biological parameters of the water. In the case of water bodies, the parameters must meet a defined criterion so that they can harbor life and maintain a balanced healthy ecosystem. These parameters can be divided into three subclasses, physical, chemical, and biological indicators. Current available water quality sensors include both physical and chemical sensors, but not biological ones. Therefore, the available Water Quality Parameters or Indicators relevant to this thesis are included in Table 3.1. Note that the values for Good Quality Indicators are not defined for drinkable water but for freshwater [80] capable of sustaining life.

Table 3.1 Water Quality Parameters (ordered by their importance according to [80]) available for measuring indicating their abbreviation, whether they are a chemical (C.) or physical (P.) indicator, unit of measurement, typical range, and typical Good Quality Indicator (GQI) value for freshwaters.

Parameter	Abvr.	Type	Unit	Typical Range	GQI
Dissolved Oxygen	DO	C.	%	$0 \sim 100$	$75-125$
Potential of Hydrogen	pH	C.	H+	$0 \sim 14$	$6-8.5$
Nitrate	NO3	C.	mg/L	0 - 15	≤ 5
Temperature	T	P.	ºC	$0 \sim 100$	—
Turbidity	Tu.	P.	NTU	$0 \sim 4{,}000$	≤ 20
Conductivity	Cond.	P.	μS/cm	$0 \sim 60{,}000$	$200-800$
Oxidation-Reduction Potential	ORP	C.	mV	$0 \sim 2{,}000$	$300-500$

A Water quality parameter model is defined using the function $f(x)$, a value or evaluation of the model at the location x is always available for measurement. The model represents the current behavior of a particular WQP and is considered continuous and multimodal. These characteristics are the expected characteristics of water quality parameter maps.

WQPs are not necessarily time-invariant, implying that their stochastic behavior is not stationary since the environment is continuously changing. In that sense, a study on WQP modeling must also be done to obtain time frames in which the measurements are valid and can yield good results.

3.1.3 Autonomous Surface Vehicles: Agents and coordination

The agents in charge of performing the measurements are ASV. The vth ASV, considering v the id number of an ASV in a fleet of V ASVs, i.e., $v = [1, 2, 3, \dots, V]$, must travel on the surface of a water body from location to location, loiter around the measurement point, i.e. stopping and managing to stand still, perform the measurement (and save/send/store), and obtain the next measurement location according to the system. All these actions must be performed by their on-board system. Therefore, it is necessary to define the specific on-board systems of the ASVs.

Guidance, Navigation & Control (GNC)

As discussed in the previous chapters, all ASVs are admittedly capable of traveling through the surface of any water body and, in this case, can also perform Water Quality measurements. The GNC system on board is capable of determining the location of an ASV can be determined by assuming a certain error. The Navigation subsystem provides a location value of the vehicle within the real world; however, since the positioning system is not perfect, the positioning error leads to guidance inaccuracies. These inaccuracies in the guidance include considering that an ASV is not at the estimated location. To efficiently overcome this issue, the map model to be obtained can account for this and define that the square side length d, previously defined, is greater than the expected error output d_{err} of the location estimation. Then, the assumption is that the WQPs measured at the real location x are the same for all nearby locations that are d units away, in terms of latitude and/or longitude, including the assigned measurement location $x \in \mathcal{M}$.

The Guidance subsystem will plan paths according to the current location and the goal location. Taking into account the previous definition of this subsystem and its components, in this thesis, each ASV is considered to have a dedicated Local Path Planning component, which uses an already defined goal to obtain a collision-free path that can effectively and efficiently move the ASV from one location to another. If there are no obstacles in the segment, a direct route is planned; otherwise, the guidance system plans a path using RRT*, as it can provide quick and good routes. The control subsystem is the last subsystem of the GNC, and it actively modifies the motor and other actuators states to reach the desired locations. However, since the navigation subsystem has an error, the goal location may be different from the final location reached. This error also accounts for water and air currents, stopping mechanisms, etc. Consequently, the Mission Planning and Global Path planning components are the only non-essential components that are left to be defined, but will be in the next chapter.

ASV constraints

Regarding energy and autonomy, considering that the WQP sensor system is powered by its own battery and that the onboard electronics consume relatively very little power, battery usage, and time consumption are considered constraints, defining the maximum distance that each ASV can travel. The battery of the WQP sensor system can run for long sessions, noticeably longer than the expected mission time, while the onboard electronics draw current from the main battery, but at a significantly lower rate than the motors. Moreover, the mission planning and global path planning components do not run on each ASV, as they run on a server-side system.

The usage of batteries per unit of movement cannot be exacted, but an approximation can be made considering the movement factors, the available battery, the motors utilized, their acceleration capabilities, and the desired velocity of the ASV. Considering that on average an ASV draws e current per unit of distance driven, the average velocity of the ASV is $||\vec{v}||$, and the total distance driven is D, the energy used is:

$$T_{energy} = \frac{e}{||\vec{v}||} \times D \tag{3.3}$$

Finally, with a defined maximum energy use $T_{energy_{max}}$, usually in terms of Amperes hours [Ah], the maximum distance D_{max} can be obtained using the latter equation. This value could be used to express when the mission has ended. Normally, this maximum energy usage will be less than the energy available to ensure that an ASV can return to the shore of the water body after it ends its mission.

Centralized Coordination

The centralized coordinator (CC) is responsible for running the system. It stores the data obtained by each vehicle in a database and is responsible for providing an optimal measurement location for each vehicle. It is centralized because it serves as a global decision-making component and coordinates the movement of multiple available ASVs. The paradigm is centralized coordination and decentralized execution, which yields effective coordinate solutions (surrogate models that include information obtained by all vehicles) and efficient asynchronous measurement of water quality parameters through multiple vehicles.

Water Quality Parameter Sensor System

Each ASV has a system dedicated to measuring multiple water quality parameters (pH, DO, Tu., etc.). When an ASV reaches a desired measurement location, it stops and takes measurements of all available WQPs. There are S different sensors, each one is accessed through its id number s, considering $s = [1,2,3,\ldots,S]$. To take measurements, each ASV must stop and instruct to perform a still measurement, preventing the so-called continuous monitoring, i.e., the vehicle cannot constantly obtain new information. Consequently, measurements are performed sequentially and, at the end of the mission, a discrete number M of measurements that have been performed is available. In that sense, p_m, with $m = [1,2,3,\ldots,M]$, is a location x, where a measurement has been performed.

Since multiple WQPs are measured simultaneously, it is necessary to describe multiple target functions $f_s(x)$, and, consequently, multiple measurements $\hat{f}_s(p_m)$ at location p_m. For simplicity, each measured WQP is referred to as $f_s(x)$ instead of their real name, e.g., the DO (Table 3.1) function for a water body is $f_1(x)$, etc. Then, the mission of the proposed system pivots to selecting measurement locations p_m so that each surrogate model $\hat{f}_s(p_m)$ is as accurate as possible to the real behavior, which in turn can be described in terms of a Multi-Objective Optimization.

The specific required on-board system is described in Figure 3.3. It comprehends the 4 systems discussed in this section. The input values are directed to specific subsystems so that autonomy can be efficiently achieved. The output values are directed towards changing the location of the ASV or to inform the coordinator about measurements performed. In Figure 3.4, an overview of the behavior of the system can be observed. It is a simplified version that shows only what matters to the mission and objective of this thesis. In this thesis, the insights, detailed behaviors, and components of each member of the diagram will be thoroughly described.

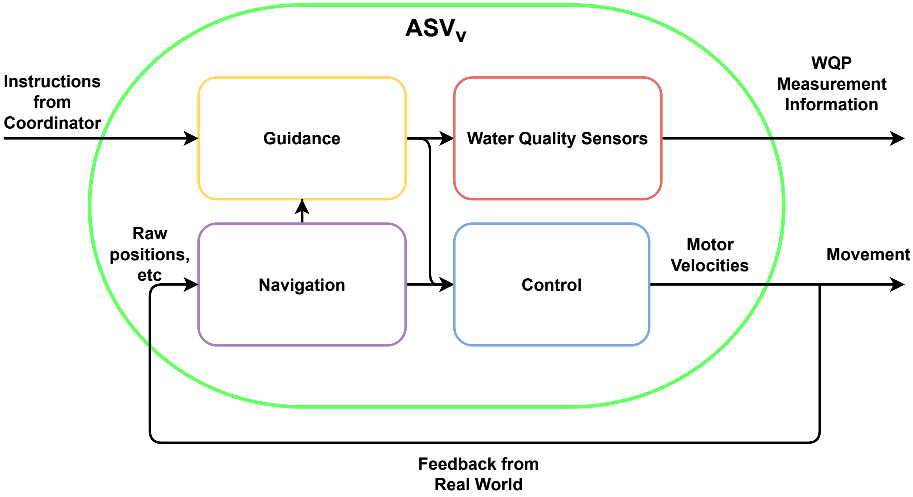


Figure 3.3 Simplified on-board systems of each ASV. In this thesis, these systems are considered to be available and fully functional.

3.1.4 Objective Functions

Multiple WQPs must be obtained simultaneously, and this mission can be seen as a Multi-Objective Optimization equation, which typically has the form of:

$$\begin{gathered}\text{optimize } \{\sum_{s=1}^{s=S} \mathcal{F}_s(x)\} \\ \textbf{s.t.} \;\; x \in \mathcal{X}\end{gathered} \tag{3.4}$$

where $\mathcal{F}_s(x)$ is one of the S functions to be *optimized* (these functions are not the unknown functions $f_s(x)$), i.e., $\mathcal{F}_1(x), \mathcal{F}_2(x), \ldots, \mathcal{F}_S(x)$. Generally, there is no solution that simultaneously optimizes all functions, since the objectives are counterbalanced, that is, the value that optimizes $\mathcal{F}_s(x)$ is not equal to the value for optimizing any other function $\mathcal{F}_s(x)$.

Each function $\mathcal{F}_s(x)$, rather than each WQP function, corresponds to a summarized error value of the surrogate model obtained of a real WQP s in a particular water body. This thesis considers each function $\mathcal{F}_s(x)$ as a function that evaluates the average difference

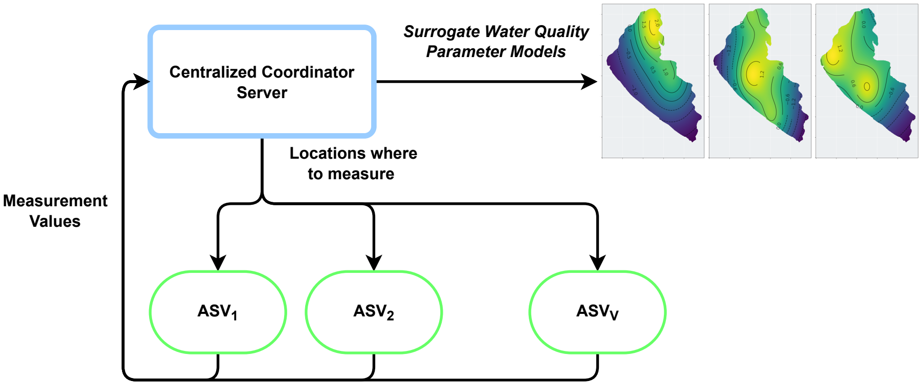

Centralized Coordinator Server
Surrogate Water Quality Parameter Models
Locations where to measure
Measurement Values
ASV$_1$
ASV$_2$
ASV$_V$

Figure 3.4 System Overview comprehending a Centralized Coordinator, ASVs, which in turn has GNC and WQP Sensor systems, and the expected output (reliable surrogate WQP Models).

between a real or objective function $f_s(x)$ and the surrogate or approximate regression $\hat{f}_s(x)$, and this value is considered a performance metric.

Performance Metrics

To estimate the model error, the mean squared error (MSE) is usually taken as the loss function. MSE measures the quality of a surrogate model w.r.t. the real model in terms of squared differences between them. MSE yields an absolute error value, which cannot be used directly to evaluate multiple MSEs simultaneously because, as shown in Table 3.1, the real values are not normalized and therefore an average value is not directly available. Consequently, another metric will be used to account for this normalization.

The Coefficient of Determination is a regression analysis estimator that evaluates how well independent variables can explain the dependent variable. Moreover, this measure, often named R^2 or R-squared, also depends on the square difference, as does MSE, and additionally depends on the variance of the data, which acts as a normalization factor. Then, a single average of R^2 values can be obtained for multiple different WQP models. R^2 has the form of:

$$\mathrm{R}_s^2(f_s(x), \hat{f}_s(x)) = 1 - \frac{\sum(f_s(x_i) - \hat{f}_s(x_i))^2}{\sum(f_s(x_i) - \bar{f}_s(x))^2} \tag{3.5}$$

In this equation, $\bar{f}_s(x)$ corresponds to the mean value of the real data. Since the equation evaluates the proportion between the sum of squared errors (i.e., the residual sum of squares) and the total sum of squares, the value can vary between $(-\infty, 1.0]$, because the proportion, which acts as a variation of the MSE (which improves towards zero), is subtracted from the value of one. The system will be useful when the average R^2 of multiple surrogate models is closer to one.

Multi Objective Optimization

Since R^2 is the selected performance metric, a maximization is obtained for multiple target functions, which implies that the optimization procedure (Eq. (3.4)) is to obtain the highest values for each R^2. Furthermore, the maximization of each $\mathcal{F}_s(x) = \mathrm{R}^2_s(f_s(x), \hat{f}_s(x))$ is not done directly, but by fitting the surrogate models according to the measurements that were recently evaluated. In that sense and with regard to multi-objective optimization, after fitting the surrogate models, a linear scalarizing a priori method is applied, with equal weights, so that the solution corresponds to an average of the R^2 values. Definitely, the optimization procedure, obtaining better surrogate models, is done indirectly (through the proposed monitoring system), whereas the performance evaluation is done using the obtained surrogate models and a comparison of them with the real models.

3.2 Bayesian Optimization Framework for Monitoring

With the environment appropriately defined both mathematically and according to the water monitoring application, the next step is to formally define the system, which applies the Bayesian framework to obtain measurement locations. For simplicity and without loss of generality, the present and future sections will consider only one WQP and one ASV, such that a bottom-up architecture can be appropriately defined afterward.

Originally, the idea of BO focuses on optimizing (finding maximum/minimum values) an unknown function, which is not only costly to evaluate, but its first and second derivatives are also unknown to the decision maker and the system [79, 65]. This optimization is done sequentially, through the usage of utility functions based not on the real unknown function but on a surrogate model that yields an approximate function, so that in real terms the optimization is done to the surrogate model instead of the real one, while the surrogate model is sequentially expected to resemble the real model. Hance, BO has the general form of:

$$x^* = \underset{x \in \mathcal{X}}{\arg\min}\, f(x) \tag{3.6}$$

which implies that the optimization goal is to find the minimum location for the black-box function, i.e., x^* is the measurement location that optimizes the WQP function $f(x)$.

Considering a proposed measurement location x^* and the fact that $f(x)$ is unknown, the real function is replaced by the regression produced by the surrogate model $\hat{f}(x)$. Then, to update the belief, the function is evaluated (WQP is measured) at this location x^*. Now, the system can incorporate the newly obtained information, update the surrogate model, and, accordingly, propose a new location for measurement x^*. This is why BO is a sequential method and, since it depends on surrogate models, it also uses utility functions that may depend on stochasticity. The right-hand side of Eq. (3.6) can be replaced with these components, considering that the utility function $\alpha(\cdot)$ depends directly on the surrogate model.

3.2.1 Surrogate Model

The BO paradigm is to define a location that optimizes an unknown function under a lack of information or uncertain beliefs. For that matter, surrogate models are used to infer the behaviors of the true black-box functions. These models can be both deterministic and stochastic. The latter is the most widely adopted because uncertainties can be used efficiently to balance exploration/exploitation through specific utility functions. Stochastic models yield not only the predicted mean value but also the related uncertainty. In the literature, $\hat{f}(x)$ corresponds to the mean value of the stochastic model but is more frequently referred to as $\mu(x)$, while uncertainty is referred to as $\sigma(x)$. Gaussian Processes are often used as probabilistic surrogate models, since they include prior beliefs (such as expected behavior and covariance) and fit the regression through posterior updates (using evidence). Moreover, GPs are relatively easy to understand because of their derivation from normal distributions. GPs are formally described in the next section (Section 3.3).

3.2.2 Utility Function

Utility functions often use the GP regression to define optimal evaluation locations. Generally, these functions consider both the mean and the uncertainty to acquire a new location to measure. It is evident that this proposed location is x^*, and that the complete implementation of the BO procedure is redefined from Eq. (3.6) as follows.

$$x^* = \underset{x \in \mathcal{X}}{\arg\max}\ \alpha(\mu(x), \sigma(x)) \tag{3.7}$$

considering that the mean and uncertainty are obtained using a fitted GP. Moreover, the procedure is now to maximize the function, mainly because, generally speaking, these functions are designed that way. Note that if we consider a weighting factor for both arguments of the function $\alpha(\cdot)$, the acquired measured location x^* can be based exclusively on the mean or uncertainty. When considering only the uncertainty, the procedure is often named Active Learning, in which the idea is to measure the most unknown location to learn the model as accurately as possible [81]. In movement-free environments, this procedure would be preferred, since the mission is to obtain fair approximations of the unknown environment. However, since vehicles must travel from location to location, the system can also make efficient use of the mean value to obtain the measurement locations. Generally, the specific function is named Acquisition Function. An extensive study on Acquisition Functions can be found in Section 3.4.

3.3 Gaussian Processes as Surrogate Models

Gaussian Processes are a generalization of the Gaussian probability distribution [20]. Based on multivariate Gaussian distributions, GPs can fit data with ease (some examples include [19, 67, 12]) through the definition of its 2 main components: the mean function $\mu(x)$ and the covariance function $k(x,x')$, both of which are expectations of values according to a function, more formally:

$$\begin{aligned} \mu(x) &= \mathbb{E}[f(x)] \\ k(x,x') &= \mathbb{E}[(f(x)-\mu(x))(f(x')-\mu(x'))] \end{aligned} \tag{3.8}$$

Note that these functions are defined according to the real process $f(x)$, but since the process is unknown and its evaluations can be noisy, it is safe (and almost necessary) to define the target function also as stochastic, with this an approximation of $f(x)$ can be obtained with the defined mean and covariance functions, according to [20]:

$$f(x) \sim \mathcal{GP}(\mu(x), k(x,x')) \tag{3.9}$$

The equation above implies that $\hat{f}(x) = \mathcal{GP}(\mu(x), k(x,x'))$, so the regression of the surrogate model in this definition yields a stochastic-dependent variable. In conclusion, GPs can be described as a way to obtain a Gaussian distribution for each input variable. In the case of this thesis, this means that a GP can obtain the expected value of a WQP (and its standard deviation) for each location inside the water body.

One of the main advantages of GPs is that they support the fit of the model through modifications (marginalization and conditioning) of their parameters that will maintain their closed behavior, i.e., the result of applying these particular modifications yields another GP [82]. Since the confidence interval encloses a volume, the 2D GP regression is often represented by two 2D heat maps showing the mean and standard deviation functions of the model. The dependence between the input locations is modeled by the covariance function, which will be defined in the next subsection. Subsequently, the process of obtaining a regression is described in a suitable way.

3.3.1 Covariance Functions

Covariance functions, or *kernels*, are the functions typically used for pattern analysis that are designed to provide a measure of similarity between two random input variables. The relationships values can be grouped into three different ones, but but since the desired GP regression should be able to describe a 2-D map of a supposedly continuous spread of a water quality measure, the studied kernels should be as smooth as possible and present positively correlated values for similar inputs, i.e., close-by locations present similar values of WQPs.

Kernels describe every possible covariance between the inputs, and, usually, to ensure handling, this description is modeled as a matrix K, whose elements $K_{i,j} = k(x_i, x_j)$, this matrix is called a *Covariance, Gram or Kernel Matrix.* For GP kernels, this matrix is always positive semidefinite, which implies a closed behavior, always forming a square matrix. Therefore, to obtain an inferred distribution, the kernel matrix can be augmented to include covariance values between the test locations $x*$ with themselves and the covariances of them with the known data input x using:

$$K = \begin{bmatrix} K & K_* \\ K_*^T & K_{**} \end{bmatrix} = \begin{bmatrix} k(x,x) & k(x,x*) \\ k(x*,x) & k(x*,x*) \end{bmatrix} \tag{3.10}$$

where $x*$, should not be confused with the next measurement location x^*. Observing 3.10, it is evident that the augmented matrix K is symmetric considering that a kernel function will always be defined as symmetric $k(x,x') = k(x',x)$; this property will be useful to obtain the inverse later. Any covariance function that is symmetric can be an appropriate kernel function for a GP if they create a Positive Semidefinite (PSD) matrix. It is important to define that *train inputs* corresponds to the locations in which a measurement has been performed, i.e., there are *train target* values, while *test inputs* correspond to locations in which we would like to infer a *test target* or value. Next, some of the most well-known kernels are described in the following paragraphs.

Another property of most kernels corresponds to stationarity; a kernel is said to be stationary when it is a function of $x - x'$. Thus, any translation changes in the input space do not affect the kernel output. Furthermore, if the kernel is a function only of $||x - x'||$, it is also isotropic (invariant to all rigid motions) [20]. Most of the kernels are isotropic, but they need to be selected according to the expected behavior of the black-box function.

Example of covariance functions

- **Constant kernel:** A standard covariance function denoted by ${\sigma_0}^2$, which determines the average distance of a target value from the mean of the function. Usually σ_0^2 appears in every kernel, but some authors prefer to treat them as different kernels, implying that operations between kernels are possible.

- **Squared Exponential (SE):** The kernel most widely used in this field, also known as RBF (Radial Basis Function). It is a function that captures covariance within the range of [0, 1]. This kernel has a hyperparameter ℓ that corresponds to a *length scale* that describes the smoothness of the function. Provides a measure of how far apart two different inputs can be to affect or influence each other. The higher the value of ℓ, the slower the rate of change. Moreover, larger values provide more extrapolation limits. It has the form of

$$k_{SE}(x,x') = \exp\left(-\frac{||x-x'||^2}{2\ell^2}\right) \tag{3.11}$$

  The SE kernel is very useful for synthetic data and not very noisy real behaviors, since the smoothness provided by the SE can fit virtually any curve/function. It has proven to be very useful for *first approach* modeling in many scenarios [19, 65, 67].

- **Rational Quadratic (RQ):** A kernel that represents an infinite sum of RBFs with different characteristic length scales. As the infinite sum can be correlated to a scale mixture, the RQ kernel expects the objective function to behave smoothly across many length scales.

$$k_{RQ}(x,x') = \left(1 + \frac{||x-x'||^2}{2\alpha\ell^2}\right)^{-\alpha} \tag{3.12}$$

  The hyperparameters are ℓ, which behaves exactly like SE, and α, which corresponds to the scale mixture parameter. As α tends to ∞, RQ is approximately equal to SE.

- **Matérn:** Contrary to the kernels mentioned above, the Matérn kernel is capable of fitting less homogeneous data. It has two positive hyperparameters: ν and ℓ, and it has the form of

  $$k_{Matern}(x,x') = \left(2^{\nu-1}\Gamma(\nu)\right)^{-1}\left(\frac{||x-x'||^2\sqrt{2\nu}}{\ell}\right)^{\nu} K_{\nu}\left(\frac{||x-x'||^2\sqrt{2\nu}}{\ell}\right) \quad (3.13)$$

  The first hyperparameter defines the smoothness of the function (i.e., the greater the value, the smoother, up to $\nu \to \infty$ where the Matérn function becomes the SE), while the latter defines the length scale of the kernel. This covariance function also incorporates two functions $\Gamma(v)$ and K_{ν} that correspond to the Gamma Function and Modified Bessel Function of second order, respectively. In several works [20, 79], ℓ is usually defined as an integer, and ν as a half-integer: $\ell + 1/2$. Despite the complexity of the expression, the Matérn kernel is also a stationary kernel valid for GPs. The Matérn kernel was designed to overcome the smoothness issues provided by other kernels, such as RQ or SE. The "aggresive" smoothness of RQ or SE prevents a good fit of real behaviors, since it is more common to find particularly noisy behaviors in nature; in that sense Matérn is generally preferred when the target function is known to be very noisy.

- **Linear:** is used whenever the target function changes at a constant rate, according to the input. Evidently, it differs from previous kernels since it is not stationary. It depends on a scale factor γ, an order P, and the dot product between inputs:

  $$k(x,x') = \frac{(x \cdot x')^P}{\gamma^2} \quad (3.14)$$

  Dot product kernels are invariant to rotations about the origin but not to translations. These kernels are PSD according to [20] and are valid for GPs. It is used mainly for temporal series.

- **Exponential Sine Squared:** is a covariance function that is very useful when the target function is known to be periodic. In this implementation, a squared sine function is used, where the argument includes the hyperparameter p, which defines the periodicity of the kernel. The length scale ℓ hyperparameter is also used to provide smoothness to the function.

  $$k(x,x') = \exp\left(-\frac{2\sin^2(\pi||x-x'||/p)}{\ell^2}\right) \quad (3.15)$$

  This kernel is very similar to the SE, but the sine function makes it a periodic non-stationary kernel. This particular kernel is useful for seasonal analysis.

A summary of the most widely used covariance functions is presented in Table 3.2, showing their respective hyperparameters defined in previous paragraphs and whether they are stationary.

Table 3.2 Examples of widely-used Kernels in Gaussian Processes.

Kernel function	Hyperparameters	Stationary
Constant	σ_0	□
Squared Exponential	ℓ	☑
Rational Quadratic	ℓ, α	☑
Matérn	ℓ, ν	☑
Linear	P, γ	□
Exp-Sine-Squared	ℓ, p	□

Operations Between Kernels

When the relationships between features (input variables) are grouped in a Gram Matrix, it is easy to perform operations between two different matrices, for example, a sum of two different independent matrices gives a new matrix and the expected properties stay the same; these properties also stay the same whenever two matrices are multiplied. Definitively, sum and product between kernels are possible. In particular, the constant kernel is always found to be part of any kernel by multiplication, so that the real function varies not only according to the input covariance but also according to the sample variance output. For example, most authors write the SE kernel as $\sigma_0^2 \cdot \exp\big(-||x-x'||^2/2\ell^2\big)$.

Any type of operation that produces another PSD matrix can be considered a valid kernel. It can be very useful to combine different kernels by adding. For example, when the expected function is known to vary daily, but the data is also smoothly varying according to the temporal input, a simple sum between an SE and an Exp-Sine-Squared kernel can be used. In this case, the periodicity is clearly defined according to the time from day to day and the length scale can be described according to the short-term expected variance. Another example is a special kernel that sums the selected with a kernel that accounts for noise known as White Kernel, where if a kernel matrix K is added by a Whiter Kernel matrix (a diagonal matrix with value σ_n^2), the kernel can be described as

$$\begin{array}{rl} k(x,x') + \sigma_n^2 & \text{if } x = x' \\ k(x,x') & \text{otherwise} \end{array} \tag{3.16}$$

which, if represented as a matrix K, is exactly $K + \sigma_n^2 I$, with I as the identity matrix, which still is a PSD matrix. The White kernel is used to represent the expected noise input level. Noise ε for each measurement performed is not known, but it is safe to assume that it is independent and identically distributed ($i.i.d$) with variance σ_n^2. This special kernel should only be added to the kernel when computing with real data gathered ($\hat{f}(x)$) and not to describe expected output values ($f(x*)$). In that sense, during the evaluation of a GP solution, the White kernel can be mathematically applied during the fitting and not while inferring the data. The next subsection explains the solution of a GP.

3.3.2 Solution of a Gaussian Process

Having the kernel exhaustively defined, obtaining inferred output data for some input is the remainder of the assignment in this section and the ultimate goal of a GP. Taking into account a set of inputs x with unknown output, a distribution *prior* can be elaborated with the information at hand: an expected zero-mean output with a known kernel. However, the result will be a zero-mean output with a variance of σ_o^2 everywhere. Then, if there is any evidence, the GP can be updated so that the regression is conditioned to predict results related to the collected data. A likelihood made from previous evidence can fit the input data into a model known as *posterior*, which can predict the output values $f(x*)$.

$$\begin{bmatrix} y \\ f(x*) \end{bmatrix} = \mathcal{N}\left(0, \begin{bmatrix} K+\sigma_n^2 I & K_* \\ K_*^T & K_{**} \end{bmatrix}\right) \tag{3.17}$$

for simplicity, $\hat{f}(x)$ has been renamed to y, usually named *y-values*. To obtain an inference w.r.t. the obtained data and the covariance function, i.e., to obtain the regression for a GP, the above equation must be solved for the mean $\mu(x*)$ and the standard deviation $\sigma(x*)$ of the test points where we would like to obtain an approximation. The solution is then described as a two-function tuple according to [20], as follows:

$$\mu(x*) = \mu_{prior}(x*) + K_*^T(K+\sigma_n^2 I)^{-1}(y - \mu_{prior}(x)) \tag{3.18}$$

$$\sigma^2(x*) = K_{**} - K_*^T(K+\sigma_n^2 I)^{-1}K_* \tag{3.19}$$

This is known as the posterior model. Notice that in these equations, the prior mean is a non-zero value $\mu_{prior}(\cdot)$ that is typically set directly as the mean of the train targets, so that if the prior mean is zero, Eq. (3.17) is equivalent to this formulation. Since the operations are just a subtraction on the train targets followed by an addition on the test targets, for simplification, Eq. (3.18) can be replaced with

$$\mu(x*) = K_*^T(K+\sigma_n^2 I)^{-1}(y) \tag{3.20}$$

with a hidden preprocessing operation consisting of subtracting the mean of y from y, and a postprocessing operation consisting of adding the mentioned mean to the test targets. On the other hand, note that Eq. (3.19) does not depend on the target *y-values*, which is another good property of GPs.

In each second hand of Eqs. (3.19), (3.20), there exists a term that includes computing the inverse of a matrix. For computational time improvement and numerical stability, this inverse is obtained by using a Cholesky decomposition procedure. Suppose that

$$(K+\sigma_n^2 I)\cdot\mathcal{L} = y = f(x) + \varepsilon \tag{3.21}$$

Taking into account the first equality and that the prior mean can be taken as zero, it can be shown that the likelihood $\mathcal{L}$ is equal to

$$\mathcal{L} = L \setminus (L^T \setminus y) \tag{3.22}$$

where L corresponds to the *lower Cholesky decomposition* of the *Kernel Matrix*, which can be found because the Kernel Matrix K (or $K + \sigma_n^2 I$ for noisy observations) is positive semidefinite, then there exists a triangular matrix L where $K = L \times L^T$ [20]. This matrix $\mathcal{L}$ is used as the correlation between a kernel input and the train targets and, of course, does not depend on the test inputs and targets. Finally, whenever a new set of test inputs $x*$ is presented, a set of output means $\mu(x*) = K_*^T \cdot \mathcal{L}$ can be inferred, having or computing the correlation matrix K_*.

Regarding posterior covariance, there are two efficient ways to solve Eq.(3.19) [20]. In this thesis, the efficient algorithmic approach is used, since $L \setminus L^T \setminus$ was also computed before. Therefore, this expression replaces $(K + \sigma_n^2 I)$ so it is solved according to the following equation.

$$\sigma^2(x*) = K_{**} - K_*^T (L \setminus (L^T \setminus K_*)) \tag{3.23}$$

where the first term corresponds to the prior covariance and the second is a positive term that represents the information obtained from the train inputs.

Another useful information that can be obtained from the train input/target is the marginal likelihood, which will be useful for updating hyperparameters. It is the probability of the data given the model. More importantly, the logarithmic marginal likelihood is obtained, which is obtained by using the integration of Gaussians according to [20], and is as follows.

$$\log p(y|x) = -\frac{1}{2} y^T (K)^{-1} y - \frac{1}{2} \log |K| - \frac{M}{2} \log 2\pi \tag{3.24}$$

Of course, under noisy measurements K can be replaced with $K + \sigma_n^2 I$. Recall that M is the number of measurements performed, i.e., the length of the vector y. In this equation, the determinant of K must also be obtained and, again, PSD matrices offer a simple way to compute the determinant using the Cholesky decomposition [20].

$$\frac{1}{2} \log |K| = \sum_{m=1}^{M} \log L_{mm} \tag{3.25}$$

The logarithmic marginal likelihood value is easily obtainable using the previous equations and can be used to adjust hyperparameters to ensure data fitting and to evaluate the behavior of the model given the data.

3.3.3 Models and Hyperparameters Selection

Rasmussen and Williams [20] defined a series of steps to select an appropriate model according to specifications; these steps help to define hyperparameter values such as length scales, periodicities, etc. In this thesis, the general guideline is followed according to the book of the mentioned authors.

When using GPs, model selection boils down to selecting (or designing) favorable kernel functions, and this is, of course, problem-specific. In that sense, model selection is open ended, but should at least consider some *metadata* that corresponds to the inferred environment. In this thesis, we select some of the kernels described in the preceding section and obtain their performance metric. The selected kernels are tested according to the expected behavior of the water quality parameters using synthetic or simulated data. The complete procedure for selecting a model will be defined in the next chapter.

Due to the dependence on hyperparameters, GPs also need them to be selected so that fitting of the model given the data is also possible and, more importantly, efficient. Using the definition of marginal likelihood, the general procedure is to compute the probability of the model given the data, estimate the generalization error, and finally bound this error. Note that in Eq. (3.24), three distinctive terms rise; the first describes how the data fit to the kernel because of its dependence on the y values, the second directly penalizes the complexity of the kernel, regarding the current measurements performed, and the third is a normalization factor that depends only on the number of measurements made. The algebraic summation of these terms gives a logarithmic value that describes the likelihood of the model given the data. This value can be thought of as the negative complexity of the model, since a larger value will describe a simpler model or a model that is easier to understand. Recognizing that GPs are always dealing with stochastic behaviors, the preference will always be to have simpler models. This is to apply the principle of parsimony (Occam's razor) [83]. This, evidently, yields models that are less certain of the behavior in a local scenario but more aware of the behavior on a broader scale. That is why many authors apply the *maximization* of log marginal likelihood to select the best hyperparameters. And this is another optimization procedure that needs to be done. In this case, since the logarithmic marginal likelihood is deterministic, optimization algorithms of almost any kind can be utilized. Generally, the limited-memory BFGS with box constraints (l-BFGS-B) [84] is selected because it is a fast and efficient algorithm that has been used for parameter optimization in many scenarios. However, the main drawback of trying to optimize hyperparameters using this paradigm is that, with few data, the GP model can be too simple and prevent the algorithm from converging to good hyperparameter values. Additionally, the l-BFGS-B algorithm can easily be trapped in local minima. To resolve this issue, multiple initial points may be considered in an effort to explore the objective landscape more broadly.

3.4 Acquisition Functions

GPs can be good surrogate models, but they depend on the measurements performed. Of course, with a large number of measurements, a good model will have a low error metric, but in many applications, including water quality monitoring, the smaller the number of measurements, the better. To achieve this, the selection of points or places to measure should be based on answering the question "**What point could lead me to have a more confident model of the environment?**" Of course, Active Learning will yield the best results in a movement-unrestricted scenario. However, in the case of this thesis, other acquisition functions can also be useful, since the selection of new measurement locations will be based not only on the GP but also on the current state of the ASV.

Acquisition functions are used to determine the projected utility of measuring a specific point within the search space. Usually in BO, an agent seeks to explore during the beginning of the mission and exploit the maximum/minimum values towards the end. There are several widely used acquisition functions, each balances exploration/exploitation according to the design of the function. This means that the balance of exploration/exploitation is done automatically with the usage of the functions. However, most acquisition functions include an explicit exploration/exploitation parameter ξ, which manually weights the parameter, which in our case should always favor clearance of uncertainty over optimization. All selected acquisition functions will undergo test scenarios and the one that selects the best measurement locations on average will be selected to be part of the final system. Before defining some acquisition functions, it is better to define the normalized optimum response of the surrogate model, which is named Z. According to [79], with the current minimum (best) value measured $f(x^B)$, Z has the form of

$$Z = \begin{cases} \frac{f(x^B)-\mu(x)-\xi}{\sigma(x)} & \text{if } \sigma(x) > 0 \\ 0 & otherwise \end{cases} \tag{3.26}$$

which implies that Z will have higher (positive) values for high negative mean values or high covariance values. Having Z defined, some of the most well-known acquisition functions in a BO setting are described below.

- **Probability of Improvement (PI):** It computes the probability of obtaining a value that is better than the current best minimum value. It is based on the Cumulative Distribution Function (CDF) $\Phi(Z)$ of the normalized surrogate model Z. The probability of improvement of a location can be obtained simply using the following.

  $$PI(x) = \Phi(Z) \tag{3.27}$$

  Since Z is large for an optimum minimum point, it is expected that PI will always prefer to exploit the black-box function once the system evaluates it near a local/-global optimal region, because the CDF is proportional to Z. This statement is true for several reasons, in the context of this thesis, including having a limited bounded box region, expecting a normal distribution that will limit the value of $\mu(x)$ that we can evaluate, and the fact that WQPs are expected to have simple, continuous, smooth behaviors, despite being multimodal. In that sense, PI is usually better for unimodal behaviors (or for very few optimum locations).

- **Expected Improvement (EI):** It improves the PI function using predictive uncertainty to prevent exploitation bias that occurs when only the probability of improvement is considered. Therefore, EI uses not only the CDF but also the Probabilistic Density Function (PDF). EI returns a measure of optimization beliefs based on the current CDF $\Phi(x)$ and PDF $\phi(x)$ of the standard normal Z.

$$\mathrm{EI}(x) = \begin{cases} 0 & \text{if } \sigma(x) = 0 \\ (f(x^B) - \mu(x) - \xi)\Phi(Z) & \\ +\phi(Z)\sigma(x) & \textit{else} \end{cases} \tag{3.28}$$

The exploration/exploitation balance is implicitly found in this AF. The predictive mean (found in the first term of 3.28) manages the exploitation weights, while the predictive uncertainty $\sigma(x)$ provides more exploration weight as it increases.

- **Scaled Expected Improvement (SEI):** It is an extension of the EI method proposed in [85]. In this function, the $\mathrm{EI}(x)$ is scaled to favor the selection of points where the improvement has a small variance, but the predicted value is expected to be high. The scaling factor $\mathbb{V}[x]$ is calculated as

$$\mathbb{V}[x]^2 = k(x,x)[(Z^2+1)\Phi(Z) + Z\phi(Z)] - \mathrm{EI}(x)^2 \tag{3.29}$$

The SEI is expressed as

$$\mathrm{SEI}(x) = \frac{\mathrm{EI}(x)}{\mathbb{V}[x]} \tag{3.30}$$

Active learning should also be introduced in this subsection since it is a straightforward and simpler approach to obtain unknown models based on exploration. But it will be seen in the next sections and chapters that it is not guaranteed to provide the best results in a constrained movement scenario like monitoring through ASVs. Active Learning considers Eq. (3.7) weighting a complete bias towards the uncertainty of the model, such that the next optimal location corresponds to the location in which the uncertainty is the maximum, or, more formally:

$$x^* = \underset{x \in \mathcal{X}}{\text{arg max}}\ \sigma(x) \tag{3.31}$$

which can be considered as one of the possible AF to be used by the system. Note that the usage of this *Acquisition Function* (AF), as well as any other, will be selected according to the constraints of the ASVs, which will be discussed in later sections.

Now, with all the AFs presented, it is important to state that the general idea is to select one of the functions according to their expected utility; i.e., exploration-based functions should be used when the uncertainty of the GP needs to be minimized, while exploitation parameters/functions should be favored whenever a certain value of the function is being searched. These AFs provide useful results in BO when used to optimize costly functions. However, they present some limitations related to their application to the monitoring task of an ASV. Therefore, additional work is required to adapt them so that BO can be applied in large-scale water environmental monitoring tasks.

The process of obtaining a new measurement position in this thesis differs from common approaches, since the assumptions of the search space are also different. Water bodies are generally non-convex sets, and the definition of the space is also limited to where the water or the terrain is present. This constraint leads to the recognition that the monitoring space is not continuous on the shores of the lake as water quality information cannot be obtained in the terrain. With these assumptions, commonly-used fast optimizers like the l-BFGS or the Newton-Conjugate Gradient cannot be used, as they only behave correctly whenever the search space is continuous. l-BFGS-B works great for general BO approaches; however, in the general scenario, a box constraint is present and every point inside of the boundary is available for evaluation. This, of course, is not the case in this thesis, so other methods must be addressed to obtain the best possible next measurement location.

3.4.1 Proposed Acquisition Functions

To overcome the limitations of classical AF for the monitoring task of an ASV, new adaptations are proposed to include mechanisms that encourage the selection of points that are close to the current position of each vehicle. Three different adaptations have been designed and tested; each focuses on achieving optimal measurement locations near the current position of the ASV. The main idea is to take advantage of the long distances traveled from one measurement location to another. Consider the following figures as shown in Figure 3.5 in which there exists a previous measurement already done at locations marked as yellow triangles, the last location that coincides with the current location of the vehicle marked as a red triangle and the measurement locations marked as blue crosses according to one of the Adapted Acquisition Functions that are described as follows.

Adaptation functions

- **Split Path:** During long distances, the ASV can take several samples before updating the surrogate model. This method calculates the new target location according to the classical AFs and the current ASV position. Then, a number of measurements is taken within the segment from the vehicle's position and the target point. These locations are equally distanced between each of them by a constant l, which defines the average distance that the ASV should travel to perform a new measurement and is different from the hyperparameter length scale ℓ. Figure 3.5a shows the waypoints defined by using AF with the split path adaptation. The vehicle should visit all goals and perform measurements on each point, but will only fit the surrogate model and calculate the next optimal location whenever measurements on all points defined by this method have been performed. The real distance traveled between the waypoints will generally be equal to l, except on the last path or whenever the straight line between locations crosses obstacles.

- **Truncated Path:** In this method, the direct path to the next target location according to the AF is "truncated" after a distance l. The idea is to take into account the mobility restrictions of ASV, so it should not travel long distances before updating the surrogate model. Notice that the next optimal measurement location is the same as the first in the split-path method, since it is defined by the above-mentioned AF. In fact, Figure 3.5b shows a measurement location that is the same as the first in the previous subfigure. However, in this adaptation the distance traveled by the ASV is truncated (hence the name), reaching a suboptimal location. Whenever the ASV reaches the computed location and updates the data, a new goal location is obtained.

This new location is generally not the same as the second measurement location using the split-path adaptation due to the new information provided.

- **Masked Path:** This method weights nearby locations using a simple gradient according to (3.32), where x corresponds to the locations on the search space and p_v the position of the ASV. This process is carried out by multiplying the mask by the result of the classical AF, as shown in (3.32). This method allows for exploiting the current position of the ASV in order to obtain more information with less movement.

$$\mathbf{M} = \exp \frac{-(||x - p_v||)}{l} \tag{3.32}$$

Due to the exponent, the rapid decay that this mask applies will give greater values to locations that are closer to the current location of the vehicle. Figure 3.5c shows the mask that is applied to the AF, and the resulting AF is shown in Figure 3.5d.

A summary of the proposed adaptations is included in Table 3.3, showing that they depend only on a tunable hyperparameter l. It is important to note that each has a distinctive characteristic with respect to updating the surrogate model. The split path-adaptation will perform at least one or more measurements before fitting the Gaussian model. The two others will strictly fit the model after each measurement. The truncated-adaptation provides the information to the ASV in its path to the maximum, so it can quickly update the model. Finally, masked-adaptation favors the maximum of nearby locations.

Table 3.3 Summary of properties of the proposed adaptations..

Adaptation	Distance between new data locations and ASV	Measurements done before GP Fitting
Split path	$l, 2l, \ldots, nl, \|\|p_v, p_m\|\|$	$n+1$
Truncated	l	1
Masked	undetermined	1

Definition of maximum distance for adapted measuring

In this thesis, it is preferable that the length of the new segment l is dynamic and related to the length scale ℓ of the fitted (posterior) GP ($l \propto \ell$). This rationale is based on the premise that the length-scale ℓ defines the maximum distance between the inputs ($x \in \mathcal{X}$) that allows these inputs to influence each other. Therefore, there exists a dependence between the optimal distance between measurement locations and the underlying objective function, i.e., large distances may fail to model the objective function, while very short distances can provide redundant information. Hence, since the length-scale is the only parameter that is updated taking into account the real evaluations to fit the model, an optimal distance between measurement locations can be obtained using the length-scale.

The use of length scales to limit the distance traveled is suitable and interesting because, for example, in the SE kernel, ℓ captures information about the spatial frequency response.

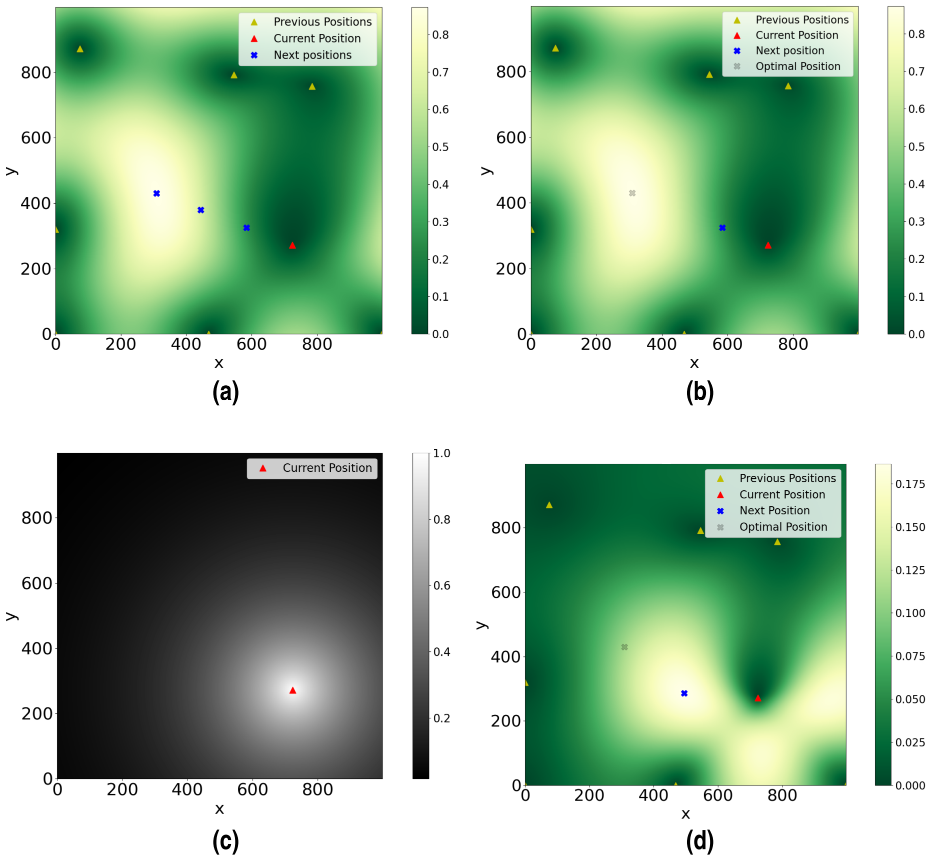


Figure 3.5 Subfigures (a), (b) and (d) correspond to examples of different results using the 3 different adaptations proposed in this thesis for the same initial conditions. Sufigure (c) is the mask applied in the masked adaptation. The images corresponds to a work previously made by the author.

To prevent loss of information, the parameter λ is set to further limit the traveling distance. Finally, the value of l is always given by the following equation.

$$l = \lambda \times \ell \tag{3.33}$$

3.5 Multi-Water Quality Parameter Estimation

At this point, a single vehicle could efficiently obtain a WQP map model just by using an appropriate kernel, and a good acquisition function that is adapted to the needs, but since there exist multiple WQPs, the next step consists of selecting a measurement location according to multiple objectives, i.e., solving a Multiple-Objective Problem. The objectives are defined following the aforementioned BO approach.

3.5.1 Multi-objective problem

Multi-Water Quality Parameter Estimation is done by slightly modifying Eq. (3.6), and consequently Eq. (3.7), to account for multiple $f_s(x)$. Evidently, this implies that there could be different GP kernels that would fit the different WQPs since their behaviors are not necessarily the same and do not have the same scale factor. Furthermore, the acquisition functions could not necessarily be the same, implying different $\alpha_s(\cdot)$ functions, considering the different WQPs. Nonetheless, since only one ASV is available (in this section of the chapter), only one measurement location should be selected considering this MOO setting. Regarding the adaptation, is is not considered as an objective, but as a posterior modification of the selected measurement location, therefore it is not included and the acquisition functions are the classical without modifications. The generalization approach consists of determining an appropriate location x^* so that a set of surrogate models can be efficiently approximated simultaneously.

Next, two ways for solving the multi-objective problem are presented. On the one hand, Multi-Function Estimation (MFE), which linearly combines every AF. On the other hand, the second method finds solutions according to a Pareto-based Multi-Objective Optimization approach.

1. Multi-Function Estimation (MFE):

   For simplicity, the MFE AF can be defined as a composite of different AFs, or a general AF that takes into account the different functions, such as the work in [71]. The proposed approach consists of composing different AFs through a fusion procedure. Based on [71], two different AF Fusions are defined, i) the decoupled and ii) the coupled methods. The only requirement for these methods is that the acquisition functions are normalized, which is normal if the functions utilize the normalized optimum response (Z).

   a) Decoupled Evaluation: It is designed to optimize one single WQP map model at a time, so that different objectives are optimized in different steps. The expression responds to select the next measurement location x^* as the argument of the maximum of the different AFs (eq. 3.34). For example, if all AFs weight the different locations of the set $\mathcal{X}$ according to the uncertainties $\sigma_s(x)$ of the different models, the measurement location will correspond to the position where one of the AFs has the highest uncertainty value. The decoupled method expression is shown below:

   $$x^* = \text{argmax}_x\{\alpha_s(x)\}\ ,\ s = [1, 2,\ \dots\ , S] \tag{3.34}$$

   b) Coupled Evaluation: It is a fusion that acknowledges the importance values of all AFs. Therefore, AFs contribute equally to a general AF. It consists of adding all the AFs. Thus, the next measurement location x^* is calculated as the argument of the maximum of the sum of the AFs, as shown in eq. (3.35). This location x^*

will be selected as the location where the different AFs will have their values maximized as a combined set.

$$x^* = \text{argmax}_x \{ \sum_{s=1}^{S} \alpha_s(x) \} \tag{3.35}$$

2. Multi-Objective Optimization (MOO):

The AFs can also be considered objectives under a MOO framework. Therefore, the optimality of a solution is based on the Pareto dominance over the objective space. A solution, which in our case is a candidate measurement location, is said to be a non-dominated solution when no other solution can maximize one of the objectives without affecting (degrading) the others. The mentioned solution is called Pareto efficient, and the set of Pareto points is called Pareto (efficient) Set. Therefore, the proposal is to use a Pareto-based approach to obtain solutions (measurement locations) that balance the multiple objective functions (acquisition functions) simultaneously.

A Pareto set is obtained using the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [86] as provided by the pymoo python package [87]. Being a Genetic Algorithm (GA), this method requires the usual GA definitions: individual, population size, number of generations, and the genetic operators: crossover, mutation, and selection functions.

- *Individual representation:* Consists of a single 2D point, which indicates the location of a measurement location.
- *Fitness function:* Consists of a vector of AF values at the individual measurement location, such that every value is directly obtained from the selected AF $\alpha(\cdot)$ considering the different models to be obtained.
- *Genetic Operator, Mutation:* Polynomial Mutation [87] has been selected so that individuals mutate locations according to a probability distribution, effectively generating offspring that explore their surroundings to find local optimal values.
- *Genetic Operator, Crossover:* Simulated Binary [87] is selected to exchange information between individuals. It works in a similar fashion to that of the Polynomial Mutation operator, creating a crossover that takes into account the data of two individuals and combines them.
- *Genetic Operator, Selection:* Because of NSGA2[86], Tournament Selection [87] has been used due to its simplicity and its usefulness for faster convergence. This operator selects the best individual out of a group according to the fitness function. Since the fitness function is multi-objective, the complete process is based on Pareto efficiency.

The result of this method is a set of optimal (non-dominated) measurement locations. Finally, since the method provides a set of possible measurement locations, the closest non-dominated solution to the vehicle is selected as the next optimal measurement location x^*. This selection is done based on the assumption that measuring on any of the Pareto efficient locations will yield the same performance. Therefore, selecting the closest location is a suitable approach considering that the measurement should also be energy efficient.

3.5.2 Multi-function adaptation

Whenever a location is selected to be the optimal solution for a future measurement, the next step is to adapt the location so that it can still account for energy consumption; this is to apply one of the adaptations found in the previous section. Subsequently, the length l is changed so that it corresponds to one of the different length scales, more formally:

$$l = \lambda \times \min\{\ell_s\} \tag{3.36}$$

3.6 Multiple Autonomous Surface Vehicles

This thesis proposes a system that monitors multiple WQPs using multiple ASVs. Since the definition of monitoring multiple WQPs has already been described in previous sections, the usage of multiple ASVs is left to be defined. The idea is that each ASV cooperates with the mission according to the collective beliefs and their locations in selecting the measurement location. Cooperation, on a broader scale, can be divided into two categories: i) intentional and ii) unintentional cooperation strategies. Each of them presents different advantages and may be preferable depending on the environment in which the agents operate.

In an active or intentional cooperation setting, the agents share their current goals and information, decide on steps such that every other agent will obtain benefits from the actions and there is a constant interchange of information so that there is a global balance in energy usage/mission accomplishment. It is generally preferred when agents need to exchange a lot of data, cannot fully operate without constant communication, and there is a need for balance in terms of energy usage.

On the other hand, unintentional cooperation describes agents that collectively pursue a mission with actions that are not deliberately joint or on purpose. In the case scenario of this thesis, this strategy would force ASVs to work on their own with the objective of completing the mission without an intended joint effort with other agents. Note that unintentional is not the same as unintended; the last adjective describes that the cooperation was not planned, but resulted as an outcome. In this thesis, ASVs are expected to cooperate, but their actions are not affected or do not consider the actions of other vehicles. The main advantage of this strategy is that an asynchronous execution can be planned and that it does not matter whether another vehicle completes its mission or is added to the fleet after another agent has already defined a measurement location goal.

Despite the chosen cooperation strategy, the plan is to obtain as many WQP measurements as possible using the available ASVs. In this sense, a centralized coordinator is needed to collect the data gathered by the vehicles. The ASVs will work (move and perform measurements) independently, they will communicate to the coordinator their locations, and the values of the WQPs according to their onboard sensors. This implies that the ASVs would only need to receive mission measurement locations instead of receiving all the data from other vehicles, computing a local Gaussian process, applying the BO equation with an AF, and obtaining a measurement location. In a sense, each ASV is unaware of the BO process, only performing Path Planning from one location to another, performing the measurement, and communicating the results afterward.

The unintentional straightforward cooperation strategy is to relegate the decision making solely to the following: i) the centralized coordinator, ii) the available WQP data at measurement location generation time, and iii) the current locations of the different ASVs. On the other hand, intentional cooperation will also consider the current missions of other ASVs (the location where they are heading) and the data that they would gather. The unintentional method will definitely work better with fewer agents available that are capable of joining or leaving the mission at any time, while the intentional method will work better for missions where there is certainty that the starting ASVs will endure throughout the whole mission, while the addition of ASVs in the middle of the mission would be counterproductive in terms of redundancy and energy efficiency.

Another important factor is initialization; normally, ASVs would start from the same harbor, but this is not necessarily the best approach, since vehicles in the same location would be too close to each other to provide useful data (redundant measures). A way to overcome this issue is to efficiently distribute the initial vehicles on the surface of the water body.

3.6.1 Initial Vehicle Positioning

A mathematical approach to solve this problem is to distribute the vehicles in the perimeter of a radius circle r centered at the center of the water body, evenly separated angle-wise, in an attempt to have each vehicle aimed at covering different regions. However, for real scenarios, it is not desirable that the ASVs start the mission in the center of the water body. Therefore, the initial positions are relocated to the shore of the water body taking into account the spacing between vehicles, having all vehicles evenly separated angle-wise but with different distances with respect to the center of the lake according to the shoreline.

3.6.2 Region Partitioning using Voronoi Diagrams

Having the vehicles at “optimal” initial locations, the next step is to partition the feasible space according to the number of available ASVs. In this thesis, Voronoi Diagrams (VDs) can be effectively used to do so and obtain evenly distributed regions. VD is a partition model of a n-dimensional space into regions of the same dimension according to a distance rule. In this work, the 2D space of the map $\mathcal{M}$ is divided into v regions, that is, one region for each of the V vehicles available. Each region has a generator, which corresponds to the position or location p_v of the vth ASV. The VD expression has the form

$$R_v = \{x \in \mathbb{R}^2 \mid d(x, p_v) \leq d(x, p_w) \; \forall \; v \neq w\} \tag{3.37}$$

where $d(\cdot)$ is assumed to be the Euclidean distance function. Using VD as a partition system is efficient and appropriate, but have some drawbacks that need to be addressed. Two generators cannot be at the same location. This is completely avoided if the initial locations are not close to each other. The outermost generators always have an infinite surface area, because the distance to a location x will be smaller than that of any other generator. To overcome this problem, it is important to consider only the feasible locations in $\mathcal{M}$. Finally, these regions are updated as needed, which means that every time an ASV needs a new measurement location, a new VD is computed, the new VD will also comply with the former definitions so that it is always valid.

The use of VD not only helps to appropriately and efficiently divide the regions according to the total number of vehicles available, but also strongly contributes to scalability and robustness. The periodic recalculation of measurement VD regions for each ASV provides an easy and reliable mechanism for robustness, because ASVs could be included and removed from the system at any time. Thus, the location of the measurement will be adapted to the number of available ASVs.

3.7 Proposed Multi-ASV Multi WQP monitoring system

In this section, the proposed approach is adequately defined. Figure 3.6 shows the proposed system, where the CC contains a database, as well as three components that work together to obtain measurement locations. For the definition of the waypoint or measurement location, the CC system performs three main tasks: i) region generation, ii) surrogate modeling acquisition, and iii) evaluation of acquisition functions. The first step finds the measurement region of the vth vehicle to nearby locations, so that a vehicle cannot find an optimal measurement location outside of its region. This search begins in Step ii), modeling the WQP maps with surrogate models. These surrogate models need to consider any previously acquired data (from all ASVs) and the inferred knowledge (prior models) of the environment, and iii) with the search region defined and the surrogate models fitted, a new measurement location is obtained according to the acquisition functions that should consider the constraints of the ASVs.

The three-step process is repeated each time a vehicle needs a measurement location and does not depend on volatile data; therefore, this process can be accomplished asynchronously among vehicles. It is important to note that this behavior helps to alleviate the workload of the CC, as it does not need to continuously transfer data to the available ASVs. In the following subsections, we appropriately define the tasks mentioned.

The general procedure of our centralized monitoring system is shown in Algorithm 1. Whenever an ASV needs a new goal (line 4), a Voronoi region is defined using the positions of the available ASVs (lines 5-6). Next, with the available data, the surrogate models are obtained (line 7) and used to find the optimal measurement location within the Voronoi region of each ASV (line 8). Note that this line is generalized with respect to the acquisition function, and that where the subscript s appears, it considers all WQP sensors. It can use MOO using GA or MFE procedures. Recall that parameter l is obtained using Eq. 3.33. Finally, the CC stores new data $D_{i+1} \leftarrow (\mathbf{s}_{i+1}, p_{i+1}) = (\mathbf{s}_v, p_v)$ when received/available and proceeds to fit the models (lines 10-11).

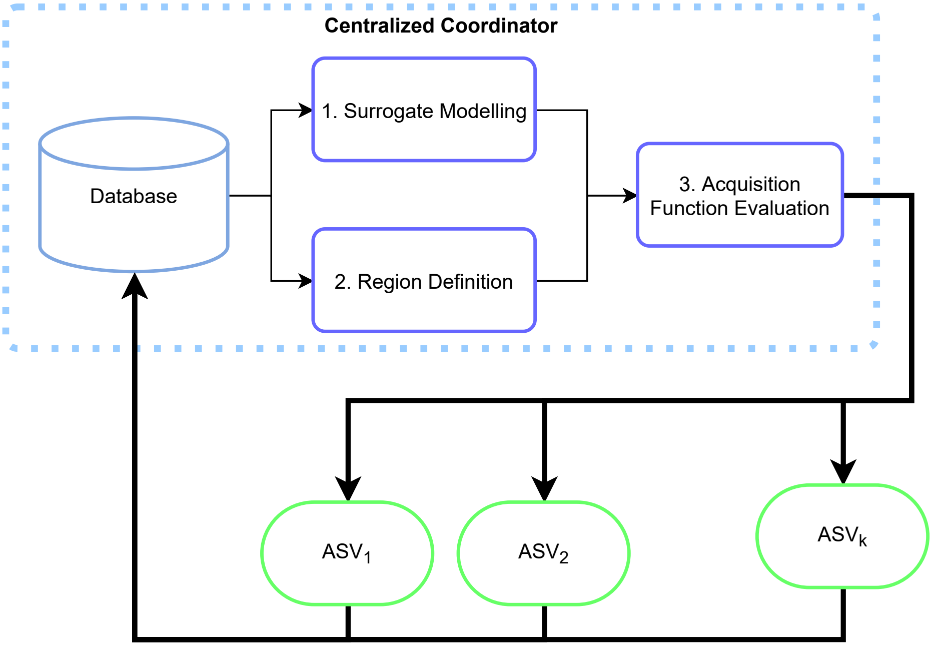


Figure 3.6 Behavior overview of the proposed system. Each ASV communicates with the CC to send data about their current locations and measurement values, and to request new measurement locations.

Algorithm 1: Centralized Coordinator procedure.

```
initialize: $D$ ←Database , $\mathcal{M}$ ← Map Data;
while $ASVs\_available()$ do
    for $asv$ in $ASVs$ do
        if $asv.needs\_new\_goal()$ then
            $v \leftarrow asv.id$;
            $R_v \leftarrow \mathrm{VD}(ASVs, asv, \mathcal{M})$;
            $\mu_s(x), \sigma_s(x) \leftarrow \mathrm{GP}_s(x)$;
            $p_v \leftarrow \mathrm{AF}(\mu_s(x), \sigma_s(x), l, \xi, R_v)$;
        end
        if $asv.performed\_new\_measurement()$ then
            $\mathrm{GP}_s.\mathrm{fit}(asv.measurement)$;
        end
    end
end
```

Finally, each vehicle executes asynchronously Algorithm 2. Whenever a new goal location is obtained (line 4), an ASV manages to perform a movement to reach the destination goal (line 10). When the ASV reaches the optimal position, it performs a measurement and sends the data to update the general water quality model of the lake (lines 6-8).

Algorithm 2: Mission planning component of the Guidance subsystem of the ASV.

```
1  initialize: $v \leftarrow asv.id$;
2  while $asv.is\_online()$ do
3      if $asv.has\_no\_goal()$ then
4          $asv.goal \leftarrow p_v$;
5      else
6          if $asv.reached\_goal()$ then
7              $\mathbf{s}_v \leftarrow obtain\_new\_measurement_values()$;
8              $asv.measurement \leftarrow (\mathbf{s}_v, p_v)$;
9          else
10             $asv.perform\_movement(goal)$;
11         end
12     end
13 end
```

4 Implementation

Vitam Impendere Vero.

Juvenal, 100 B.C.

The proposed system is implemented using embedded and computing systems according to the overview system presented in Figure 3.6, Section 3.7. The system was mainly implemented in Python language (3.9.9). The agents correspond to catamaran-like autonomous vehicles with GNC and WQP sensor systems on board. Their Navigation and Control system (low level) were executed by an embedded autopilot software/hardware system, such as Navio2[1]. For experimentation, ASVs are available that have the on-board system shown in Figure 4.1. The components shown correspond to: (a) Emlid Reach M+ GNSS system [88] for Navigation purposes, (b) a Raspberry Pi 4 [89] using a Navio2 Hat, both are used for Navigation and Control, (c) a DC regulator for the main system, and (d) a Nvidia Jetson AGX Xavier computer [90] that implements communications system as well as the Guidance system.

In what follows, the multiple components described in the previous chapter are implemented in algorithm form. Experiments and selection procedures are also described. The complete documented implementation is available as open source code on Github[2]. Due to the fact that multiple options for the different components were described, it is necessary to test the mentioned components in simulators, which requires a description and implementation of the simulator components, including simulating the environment with benchmark functions and simulating the vehicles in charge of performing the measurements.

[1] *https://navio2.emlid.com/*

[2] *https://github.com/FedePeralta/BO_drones*

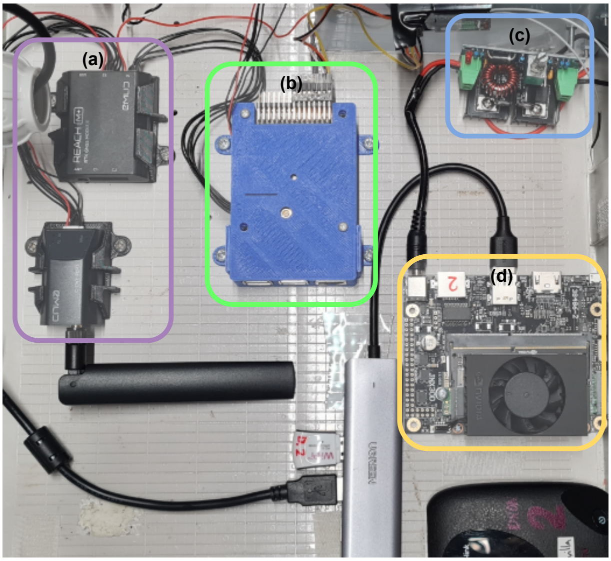


Figure 4.1 ASV on-board system consisting of (a) GNSS for navigation (b) A central computer with an autopilot system (c) DC regulator and (d) main communications and guidance system.

4.1 Map Model

The proposed system aims to monitor the WQPs of large-scale water bodies. In the case of this thesis and many other related works [14, 11, 58, 59, 57], the case study of the water body corresponds to the Ypacarai Lake (San Bernardino, Paraguay), which was previously discussed. In fact, Figure 3.2 (Subsection 3.1.1) corresponds to the Ypacarai lake, where the freshwater lake has dimensions of approximately 24 by 5.5 km and a total surface area of 60 km^2 [3].

In the case of Yparacai Lake, the simulation area is modeled as $\mathcal{M}$ with dimensions 1000×1500, where each element $\mathcal{M}_{i,j}$ corresponds to a square with side $d \sim 10[m]$. As a result, each ASV agent could be easily located in any x location and because of its size and maneuverability, it could move to any of its neighboring locations without problems. Of course, when describing the WQP models in terms of the matrix $\mathcal{M}$, the scales of the hyperparemeters of the GP are changed to fit the modeled map. This makes it easier to find parameters, since there are larger differences between neighbor locations, since the GNSS systems will not discern before in the fourth decimal of latitude and longitude terms.

4.2 Water Quality Parameter Model Maps

Water quality models are often modeled as smooth functions due to fluid dynamics and wind conditions [3, 2]. For test and validation, two approaches can be used to simulate WQP model maps: i) using benchmark functions and ii) synthetic data based on true measurements. Before these approaches are defined, the time dependence of the WQPs must be obtained.

Mas et al. [5] described the use of an online tool to obtain WQP map models of Mar Menor (Spain). In the mentioned work, the authors describe a series of procedures that they used to create synthetic map data based on 16 to 24 real WQP measurements performed continuously. The online tool offers sensor data at these locations with a time difference of ≥ 7 days between measurements and is publicly available. Taking advantage of the tool, the rate of change of the WQPs can be obtained to describe whether the system must consider static WQPs during one mission. Taking into account the available WQPs of the tool and the available WQPs of the sensor system in the ASVs, the temperature and dissolved oxygen can be obtained directly from the online tool. According to initial experiments, the available working time of a single ASV does not exceed 2.68 hours, and with that the supposed change between the initial and final observable WQP maps can be obtained using a simple interpolation method.

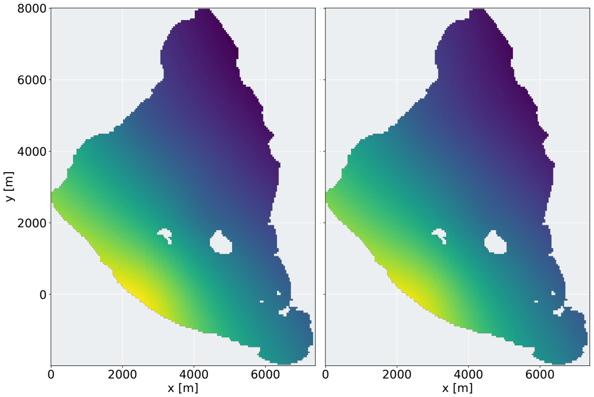


Figure 4.2 Estimated change of the turbidity WQP in Mar Menor (Spain) after 2.68 hours using synthetic data provided by [5] between August 20th and 27th of 2021. Despite the fact the data found in the provided web page varies a lot between these days, the change is not noticeable after just approximately 3 hours.

Considering Figure 4.2, if the mission is to obtain the *current* state of WQPs, a simple GP kernel that does not depend on time or is stationary is the straightforward approach and will provide a good solution for the time window in which the agents are supposed to operate. The test scenarios to validate the proposed method can be modeled as synthetic data. The idea is to create deterministic models that can be used in monitoring missions and systems. For example, a simple approximation can be performed using a radial function, as proposed by [5].

4.2.1 Benchmark Functions

WQP model can be modeled using the classical Bohachevsky or Himmelblau functions as benchmarks. However, their gradients have high values, which are not typical for

WQPs. Moreover, specifically for WQPs, the benchmark functions should be multimodal, continuous, multidimensional, and deterministic. Finally, real WQPs are not expected to behave always like the mentioned benchmarks, and it was more useful to create custom benchmark “ground truths”. This is because a WQP is expected to have many peak locations, to be continuous everywhere (except on the shore), to depend on at least two independent variables (x,y), and to have an underlying ground truth value to which it could be compared. The Shekel function (SF) is a rather unknown benchmark that can be adjusted to provide maximum locations and has gentle gradients. Additionally, the SF can have any number of maximum values, which fairly simplifies the setup procedure to obtain the ground truth for each simulation run. SF has the form of:

$$f_{\text{Shekel}}(x) = \sum_{i=1}^{M} \frac{1}{c_i + \sum_{j=1}^{N}(x_j - a_{i,j})^N} \tag{4.1}$$

where $a_{i,j}$ and c_i are the elements of two sets of matrices A and c. The matrix A has the size of $M \times N$, where M is the number of maximum locations, which are N dimensional. The matrix c is a vector $M \times 1$ that defines the inverse importance value of the maximum locations. The SF provides a parametric function capable of defining multiple optimum locations in a N-dimensional space.

Different ground-truth maps can be generated using Eq. (4.1), each can represent different maps of water quality parameters at different times. They can be post-processed so that they can fit the map $\mathcal{M}$ and $\mu_s(x) = 0$, $\sigma_s(x) = 1$, making the GP fit easier and faster. This post-processing corresponds to subtracting the real obtained values with the expected mean of the measured WQP and dividing by a factor equal to the standard deviation of the measured WQP, effectively normalizing the map, assuming the Central Limit Theorem. Examples of the generated ground truths can be seen in Figure 4.3. In either way of choosing WQP models, the different simulated agents should travel from point to point until their battery is fully used, and they performed multiple WQP measurements.

4.3 Autonomous Surface Vehicles

Regarding ASVs, the implementation was done so that the use of simulated vehicles and real vehicles is not affected by the system. In that sense, the same components of high-level systems have been designed and implemented for both real and simulated ASVs. In what follows, the descriptions of the components will be detailed for the real ASVs, but they also work for the simulated ASVs, unless explicitly described, which refers only to one of them.

4.3.1 Guidance, Navigation & Control (GNC)

The Guidance Component is designed to apply Eq. (3.7), under the assumption that a single next measurement location $p_{m+1} = x^*$ is required for a single vehicle V_v that needs a target location. Evidently, depending on the number of agents present at that time, the available locations x varied according to the region obtained for that vehicle in that scenario R_v (a complete region if only one vehicle is present) and the acquisition function $\alpha(\cdot)$

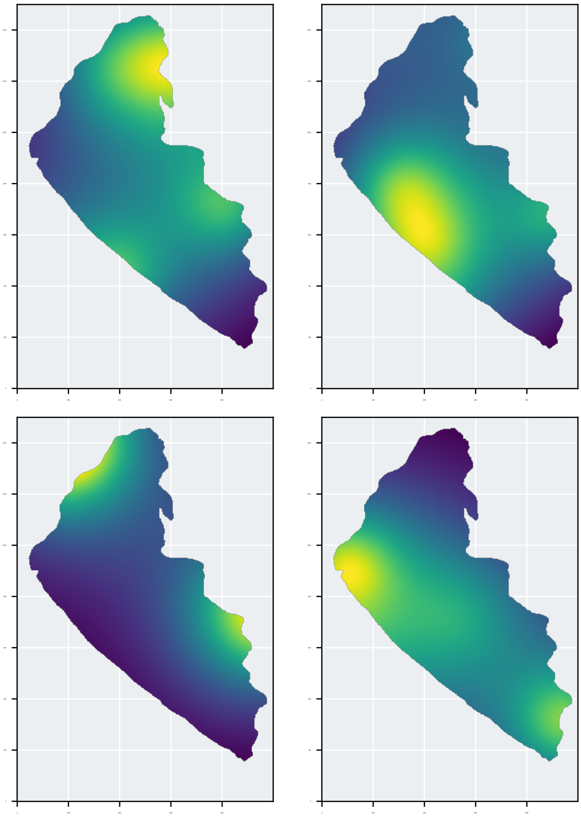


Figure 4.3 Four examples of Shekel functions with random number of maxima points at random locations scaled to a map model of the Ypacarai Lake.

selected. In the case of simulations, multiple $\alpha(\cdot)$ were tested, so this can vary whether it considers Expected Improvement, Probability of Improvement, etc. for one, the WQPs or AF fusion with multifunction estimation or multiobjective optimization for multiple WQPs.

After a measurement location has been obtained and sent to an ASV, the global and local path planner components consider the current location and the goal of obtaining an obstacle-free path; this is done using RRT*. These obstacle-free paths were obtained for the map model $\mathcal{M}$ instead of the real world map, as they were easier and faster to solve.

During the mission, the navigation system of an ASV provided location data to the CC as well as other components of the ASV, so efficient paths were obtained and efficient path following could be executed. As mentioned above, this component was executed by an embedded autopilot system such as Navio2. This autopilot system also executed the control system.

4.3.2 ASV Constraints

Regarding the main ASV constraint, battery consumption, for testing purposes, the maximum distance was drawn from the theoretical maximum battery available according to the tests performed in [91] with the average velocity considered in the same work. As a result, the simulated ASVs obtained the best possible maps. In real experiments, the battery will be used until a certain threshold level that will automatically return the vehicle to launch, or RTL as described in the Navio2 autopilot system.

4.3.3 Water Quality Paremeters Sensor System

WQP sensor systems are the main part of the ASVs, so they were always available for use. If any WQP system of an ASV is supposedly malfunctioning, the ASV will no longer be considered during the mission, so the ASV can return to the shore, be checked, and reenter the ASV system. This is possible by using the unintentional cooperation described in the previous chapter. For real measurements, a certain number of measurements will be performed at the mentioned location, so that the average values can be obtained. In that sense, the system available in the ASVs for real experiments corresponds to Libelium Smart Water [92] (Figure 4.4), which has the same physical and chemical sensors as shown in Table 3.1. On the other hand, for simulated ASVs, a measurement value was drawn directly from the simulated map, and a random i.i.d. noise was added to account for noisy sampling. The CC is an online server that acts as both a decision maker and a visualizer. All data that arrive from the vehicles were used to produce output, such as surrogate models, methods for controlling the state and health of the vehicles, etc.

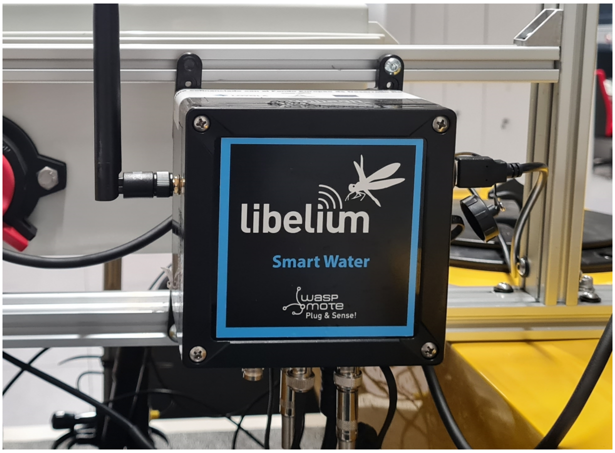


Figure 4.4 Water Quality Parameter Sensor system, Libelium Smart Water, connected to the ASV. The sensors are connected to the bottom of the embedded system and communicates to the main on-board system through an USB cable.

4.3.4 Centralized Coordinator Server

The CC is the only system that worked directly with Bayesian optimization, Gaussian Process, Acquisition Functions, and Voronoi Diagrams. Real vehicles will receive and send data via 4G, assuming that the mobile phone signal is available anywhere on the water body. The data will be sent without processing them, so only a limited amount of processing is done on the onboard computers to save energy. In that sense, the energy usage of vehicles was considered to only power the motors to perform movement.

4.4 Implementation using Python

During simulations, the main tool for comparison is the Coefficient of Determination R^2. The idea behind always considering this performance metric is to have a systematic method for comparing not only the component selection but also to observe and analyze the joint operation of multiple components such as Multi-Objective Optimization or Multi-Agent system operation. When a ground truth is available (which only happens during simulations), R^2 could always be obtained. Therefore, this decision is plausible and valid.

Considering the framework of BO already described, the implementation is to simply code in Python language Eq. (3.7). Then publicly available libraries are used to implement parts of the system in Python language; in some cases modifications of the original library were done so that the implemented code could work seamlessly within the system. The usage of the main libraries involved in the implementation are mentioned next.

- Numpy: *https://numpy.org/* is a mathematical tool used for data handling, all data are mainly handled by this package. All vectors, positions, and most operations are done using this package. Some examples include matrix multiplication, battery usage, and path planning. The libraries described below also use numpy to perform their operations.
- Scikit-Learning: *https://scikit-learn.org/stable/* is the Python package that handles kernels, such as SE, Matern, and others. Additionally, it implements Gaussian Processes, such as data processing, fitting, and even hyperparameter optimization. It is very useful for noise-free sequential decision making. However, for noisy evaluations, the package is not fully implemented. The hyperparameter optimization in this library implements Eq. (3.24) with the observed data and provides the mean and uncertainty of the fitted GP using equations. (3.19) and (3.20), using Cholesky decomposition where possible.
- Scikit-Optimize: The bases of classical acquisition functions are drawn from this package *https://scikit-optimize.github.io/stable/*. To enhance response time and fully utilize the information given by the AFs, all utilized AFs are reimplemented as new classes with added functionality.
- Pymoo: When multiple acquisition functions are to be computed and solved as a multi-objective optimization strategy, the library pymoo, *https://pymoo.org/*, is used. It is a very high-level library that, through a problem definition, could provide a Pareto-efficient set that is supposed to be part of the Pareto frontier.

- Scipy: *https://scipy.org/* is a high-level library for scientific computing. In this thesis, Voronoi diagrams are obtained directly using this library. However, copies of the locations of the vehicles are translated and mirrored to obtain a box partition of the region, since scipy only offers Voronoi computing considering non-restricted regions.
- Matplotlib: All figures regarding the results in this thesis were generated using this library *https://matplotlib.org/*. It is a simple library for creating visualizations. In addition, this library is used for visualization of the resulting WQP maps.

Since the objectives of this thesis are very specific, additional code was written for the adaptation of the acquisition functions and multi-water quality parameter estimation using coupled and decoupled fusion strategies. Both are available online. These pseudo-libraries are largely based on the libraries previously mentioned and work, to the best known knowledge of the author, only with them. In the next chapter, the simulation and experimental results are shown as a process in which, first, the basic components of the system were systematically selected according to their performance metric.

5 Results

Taking action today is worth it.

KURZGESAGT – IN A NUTSHELL, 2022

In this chapter, the results of the proposed method. based on BO for monitoring a simulated aquatic environment are described. The followed roadmap consisted of evaluating different components, selecting the best method, and building the system from the ground up. Comparisons have also been made with related monitoring systems to show its strengths.

5.1 Simulation Setup

This section defines the procedures and parameters for the evaluation of the proposed method. The simulations have been conducted in a 6-core 4.6 GHz processor computer with 16GB RAM. In the simulations, to ensure a fair comparison among different components, the simulations have been performed with the same random seeds.

Simulated Water Quality Model using Benchmark Functions

The WQP maps shown in Figure 4.3 are examples of ground-truths from the test configuration. The simulated WQP value of each location is taken from the result of a scaled standardized shekel function. For the randomization of the shekel function, random values were defined for the input matrices. More precisely, the number of peaks or maximum values was defined as a random integer between two and ten, the maximum altitude was defined to be a real number of minimum values of five to ten, while the locations of the peaks were distributed uniformly random across the map matrix $\mathcal{M}$.

Theoretical maximum energy available

According to Morel et al. [91] and Eq. (3.3), a maximum travel distance can be obtained considering the average velocity $||\vec{v}|| = 1.6[\frac{\text{m}}{\text{s}}]$, the maximum theoretical energy delivered by the battery and the consumption rate provided by the same work. These values are also in accordance with the total time that the ASVs are usually available as it was mentioned in the Chapter 4.2 Implementation. As a side note, since the model map consists of a grid map with square of side $d \approx 10[m]$, the maximum distance in pixels is 1500 pxs.

$$D_{\max} = T_{maxenergy} \frac{||\vec{v}||}{e} = 20000 \times \frac{1.6}{2.1} \approx 15000[\mathrm{m}] \tag{5.1}$$

5.2 Kernel Selection

The first selection corresponds to the surrogate model. This thesis considers that GPs are the appropriate model, but has no preference regarding the kernel. For the expected quiet waters of the Ypacarai lake [3], the prior length scale of the selected kernel should be large enough to ensure smoothness, but bounded enough to allow the kernel enough freedom to adapt to varying maximum/minimum data. The covariance functions to be tested depend all on the hyperparameter ℓ. It was decided that $\ell \sim 10\%$ is the length of the search space. For other hyperparameters, if any, the most common values [20, 79] are used.

Several simulations have been performed for kernel selection, each test consisting of uniformly drawn sets of $M \in [15, 25, 35, 50]$ valid measurement locations. To account for noisy evaluations, a small random value is added to each measurement so that the noise is i.i.d. with $\varepsilon = 0.01$. For each simulation, a kernel is selected and used throughout the experiment, fitting the data and updating their hyperparameters with the equations described in Chapter 3. All kernels are summed with a White Kernel with $\sigma_o^2 = 0.01$ to account for noise.

The results using the SE, the RQ and the Matérn kernel are found in Table 5.1, Table 5.2 and Table 5.3, respectively. For each map, 10 different simulations have been performed, and the resulting average R^2 is presented. The values found in the last column of each table correspond to the average R^2 of the five maps. The results marked in bold correspond to the best average value across different kernels.

Table 5.1 R^2 values obtained for the Squared Exponential (SE) kernel experiments with a prior length scale $\ell = 150$ with different settings regarding WQP Map model (10 different simulations per map model) and number of measurements M.

M	WQP Map 1	WQP Map 2	WQP Map 3	WQP Map 4	WQP Map 5	Avg.
10	0.7458	0.8383	0.6765	0.7265	0.8189	**0.7612**
15	0.8927	0.9229	0.8261	0.8906	0.941	**0.8947**
25	0.9564	0.9652	0.9352	0.959	0.9811	**0.9594**
35	0.9652	0.9769	0.9671	0.9561	0.9871	**0.9705**
50	0.983	0.9935	0.9826	0.9881	0.9952	**0.9885**

Note that the generated WQP maps have different values, have their peaks in different locations, and are not correlated to each other. Despite this, for each kernel, the R^2 score is similar and has less variance as the number of measurement locations increases. Regarding

Table 5.2 R^2 values obtained for the Rational Quadratic (RQ) kernel experiments with prior hyperparameters $\ell = 150$ and $\alpha = 1.0$ with different settings regarding WQP Map model (10 different simulations per map model) and number of measurements M.

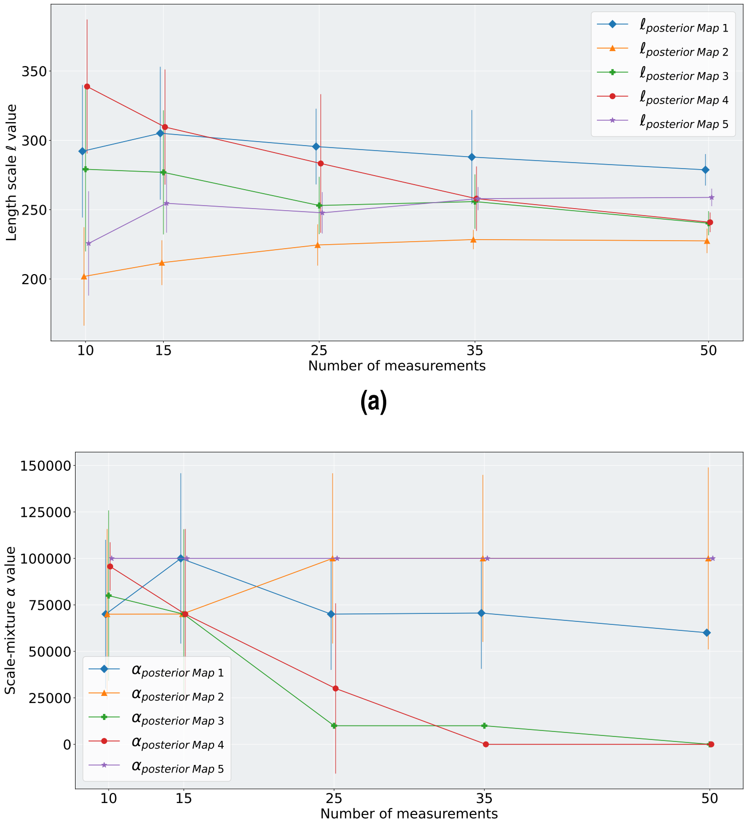
M	WQP Map 1	WQP Map 2	WQP Map 3	WQP Map 4	WQP Map 5	Avg.
10	0.7458	0.8383	0.6765	0.7253	0.8159	0.7604
15	0.8925	0.9225	0.8261	0.891	0.9339	0.8932
25	0.9563	0.9652	0.9352	0.9585	0.9811	0.9592
35	0.9651	0.9769	0.9671	0.9562	0.9871	**0.9705**
50	0.983	0.9935	0.9826	0.988	0.9952	0.9884

Table 5.3 R^2 values obtained for the Matérn kernel experiments with prior hyperparameters $\ell = 150$ and $\nu = 1.5$ with different settings regarding WQP Map model (10 different simulations per map model) and number of measurements M.

M	WQP Map 1	WQP Map 2	WQP Map 3	WQP Map 4	WQP Map 5	Avg.
10	0.6936	0.7925	0.5959	0.707	0.7952	0.7168
15	0.8709	0.9066	0.7505	0.8863	0.9065	0.8642
25	0.9447	0.9592	0.8949	0.9509	0.9619	0.9423
35	0.9527	0.9613	0.9451	0.9468	0.9703	0.9552
50	0.9794	0.9882	0.9709	0.9851	0.9895	0.9826

the different hyperparameters, Figure 5.1 shows the evolution of different hyperparameters for different kernels. Specifically, Figure 5.1a shows the evolution of the length-scale value ℓ according to the different number of measurements per WQP Map for the SE kernel. Of course, the length scale value varies according to the map, but it can be seen that the value tends to be a constant. In the case of the other kernels, the same behavior is observed. Moreover, hyperparameter α (the scale-mixture), for RQ (Figure 5.1b) kernels, reaches the maximum allowed values in simulations for some WQP maps, which means it tends to ∞, effectively, transforming the kernels into SE kernels. The application of the parsimony principle [83] also applies in this case, since a simpler method of modeling should be preferred.

The summarized results using the different kernels can be found in Table 5.4. The values in Table 5.4 correspond to the average R^2 considering five different WQP maps. The table shows that, for the Ypacarai Lake, using the SE or RQ kernels provides the best results. Consequently, either of these kernels should be indifferently selected. But since RQ is the generalization of SE and the observed hyperparameter value α reached its maximum possible value, SE is the selected kernel. Moreover, RQ is a more complex expression and provides approximately the same results. For this reason, SE is the selected kernel for the proposed BO-based monitoring system, and it will be used as the prior/posterior model for the next simulations.



(a)

(b)

Figure 5.1 Hyperparamter evolution corresponding to (a) Hyperparameter length-scale ℓ for the SE kernel, and (b) Hyperaparmeter scale-mixture α for the RQ kernel .

Table 5.4 Summarized average R^2 values obtained in the simulations using different kernels with prior hyperparameters as shown and different settings regarding WQP Map model (10 different simulations per map) and number of measurements M.

M	SE ($\ell_p = 150$)	RQ $\ell_p = 150, \alpha_p = 1.0$)	Matérn ($\ell_p = 150, \nu_p = 1.5$)
10	**0.7612**	0.7604	0.7168
15	**0.8947**	0.8932	0.8642
25	**0.9594**	0.9592	0.9423
35	**0.9705**	**0.9705**	0.9552
50	**0.9885**	0.9884	0.9826

Figure 5.2 shows an example of the prediction generated using a SE kernel with the best prior hyperparameters found in Table 5.4. The first subfigure corresponds to the Ground Truth for that experiment, whereas the second and third subfigures show the mean and standard deviation, respectively, using the measurement values obtained at random locations marked with red diamonds. Regardless of the random distribution of points, the same average result is always obtained using this kernel. This is sensible because when using SE, as a covariance function, one expects smooth behaviors and gentle gradients, as opposed to using Matérn, which creates "noisy" covariance between inputs. The hyperparameter *length scale* was changed to fit the given measurement locations on each scenario. It has been observed that there were different values of length scale when using only 10 locations, but as the number of measurement locations increased, the variance of the final length scales decreased and in most cases was very different from the prior length scale. This implies that the GP can effectively condition a WQP that behaves like the SF. A careful adjustement of the length scale can be done so that the next experiments start with a better understanding of the underlying black box function, but since different models will produce different length scale and since the real behaviors will not necesarrily always be like a SF, the prior length scale is selected as designed first along with the assumed *iid* noise. With the kernel appropriately selected, the next step is to select an acquisition function for the system.

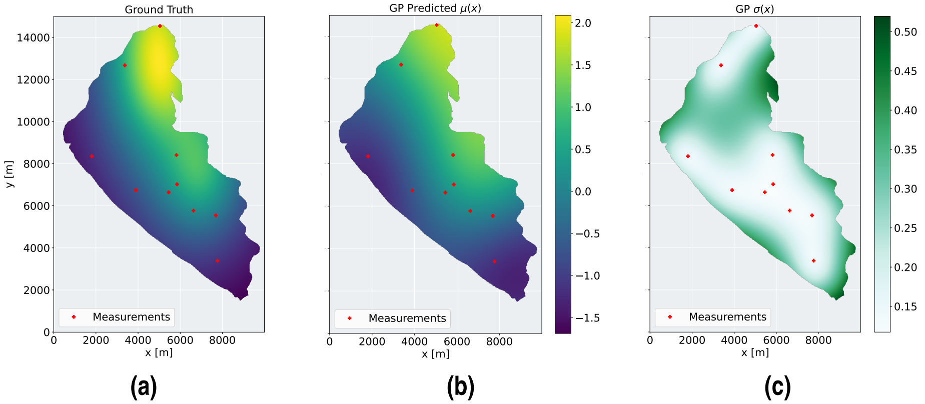


Figure 5.2 Example of posterior GP regression using the Squared Exponential kernel with $\ell = 150$. In this case, with the selected random 10 measurement locations, the fitted length scale changed to a value of $\ell = 307.346$ using the maximization of logarithmic likelihood.

5.3 Acquisition Function Selection

In this section, the results are presented to evaluate the different AFs. Since the idea is to obtain the best-adapted acquisition function for the monitoring mission, a full evaluation needs to be performed considering only one vehicle. This implies that for each acquisition function, not only should an adaptation be selected, but for each one, a proportion of the current length scale for limiting the distance needs to be carefully selected. In conclusion, two hyperparameters (ξ, λ) and an adaptation should be selected for each acquisition function. Therefore, for testing, an ASV was simulated considering different runs for different WQP map models.

Recall that after every new measurement is performed in the test scenarios, the total distances traveled were calculated and stored together with R^2 and the number of measurements carried out so far M. For the comparison framework, the R^2 and the total distance traveled (TDT) of multiple simulation runs were stored for each method. Additionally, from this section on, ASVs start at the shore of the water body, which is sensible, since surface vehicles almost always start their mission at the shore. The initial vehicle positioning is the same method described in Subsection 3.6.1, but only considering one vehicle at a time.

For the AFs: Probability of Improvement, Expected Improvement, and Scaled Expected Improvement, the following tables are presented: Table 5.5, Table 5.6, and Table 5.7, in which the average R^2 values can be observed. The results are shown for all different combinations of ξ and λ, and in each table, the best R^2 value is shown in bold. Since a good R^2 value is only required, the selected hyperparameters are the ones that give the best R^2 value, regardless of the number of measurements performed.

Probability of Improvement is the simplest AF tested. Table 5.5 shows the results according to different settings. Only one average R^2 was greater than 0.5, but it does have a large standard deviation value (0.395), implying that this method does not present a good average value and also it is not robust against initial locations. PI does present a small number of average measurements and this is one of the reasons of the average small R^2 value. Definitely, it cannot be used for the monitoring system.

According to the simulations, Expected Improvement is the best AF using the BO framework (See Table 5.6). The results are not only better regarding the objective of the system (Average R^2 value) but also considering the number of measurements performed. It is remarkable how with only a few more measurements on average, the value of coefficient of determination (R^2) is better on average. Moreover, using the truncated adaptation with a exploration bias of 2.0 and a limit of half of the current length scale, the R^2 value reaches 0.582 with a lesser variance value, only needing to measure on approximately 10 locations within the whole water body.

The results obtained shown in Table 5.7 correspond to the experiments performed with the Scaled Expected Improvement as AF, which is a scaled version of the EI that favors searching for the peaks. It has been observed that the scaling factor reduced the acquisition function to a point in which the selected measurement location presented little to no improvement regarding the decrease in uncertainty. Moreover, while using the masked adaptation, the scaling prevented the system to find new measurement locations other than the current location after the first 3-4 measurements, effectively not moving the ASV after some measurements regardless of the initial location.

Taking into account the Active Learning strategy, only different λ values can be tested, since Eq. 3.31 does not include a parameter ξ. Additionally, masking the results is not possible when the kernel is a proportion of the AF itself, which happens since both consider only an exponential value according to the distance and a length that is proportional to

Table 5.5 Results for the Probability of Improvement AF.

Adaptation f.	ξ	λ	Avg. R^2	Avg. M
Split Path	1.0	0.375	0.433 ± 0.435	16.47 ± 5.449
		0.5	0.43 ± 0.438	13.06 ± 4.204
		0.75	0.375 ± 0.475	9.64 ± 2.633
		1.0	0.393 ± 0.455	8.16 ± 2.176
	2.0	0.375	0.469 ± 0.314	17.16 ± 5.499
		0.5	0.467 ± 0.319	13.77 ± 4.176
		0.75	0.435 ± 0.35	9.86 ± 2.706
		1.0	0.43 ± 0.322	8.46 ± 2.17
Truncated	1.0	0.375	0.375 ± 0.532	10.19 ± 5.114
		0.5	0.437 ± 0.422	8.66 ± 3.653
		0.75	0.4 ± 0.485	7.139 ± 2.518
		1.0	0.357 ± 0.446	6.22 ± 1.874
	2.0	0.375	0.48 ± 0.431	10.759 ± 5.679
		0.5	0.495 ± 0.429	8.909 ± 4.103
		0.75	$\mathbf{0.508 \pm 0.395}$	7.24 ± 2.653
		1.0	0.482 ± 0.323	6.16 ± 1.88
Masked	1.0	0.375	0.41 ± 0.373	6.48 ± 1.947
		0.5	0.359 ± 0.453	6.27 ± 1.771
		0.75	0.373 ± 0.451	6.0 ± 1.772
		1.0	0.322 ± 0.479	5.59 ± 1.477
	2.0	0.375	0.408 ± 0.435	5.63 ± 1.507
		0.5	0.387 ± 0.452	5.6 ± 1.407
		0.75	0.383 ± 0.429	5.42 ± 1.335
		1.0	0.391 ± 0.327	5.21 ± 1.344

the hyperparameter. In this sense, Table 5.8 shows all the different available components tested, which consider both different values λ and two adaptations.

A summary of the best set (ξ, λ and adaptation function) for each Acquisition Function is found in Table 5.9. It is evident that using EI(x) with $\xi = 2.0, \lambda = 0.5$ or max $\sigma(x)$ ($\lambda = 0.5$) will yield the best results, and the other methods do not perform as well as these two. When considering the average number of measurements, EI(x) presents the best results, as it requires a significantly lower number of measurements to provide approximately the same average R^2 score.

In Figure 5.3, the best paths found provided by each of the best acquisition functions are presented. They start at different locations because what is best for each is not necessarily the same. In addition, different ground truths were considered to obtain the different results. However, in terms of Squared Error, a comparison can be made. Both maps are depicted according to the squared difference between their ground truth and the output mean of the fitted GP. Figure 5.3a shows that the myopic path found chooses to evaluate points that are initially approximately in the same direction but change drastically after reaching a certain

Table 5.6 Results for the Expected Improvement AF.

Adaptation f.	ξ	λ	Avg. R^2	Avg. M
Split Path	1.0	0.375	0.423 ± 0.397	16.6 ± 5.265
		0.5	0.429 ± 0.376	13.27 ± 4.094
		0.75	0.403 ± 0.401	9.69 ± 2.509
		1.0	0.417 ± 0.359	8.26 ± 1.968
	2.0	0.375	0.481 ± 0.332	17.18 ± 4.952
		0.5	0.472 ± 0.35	13.69 ± 3.9
		0.75	0.404 ± 0.447	9.88 ± 2.519
		1.0	0.413 ± 0.416	8.38 ± 1.933
Truncated	1.0	0.375	0.475 ± 0.39	10.15 ± 5.113
		0.5	0.51 ± 0.349	8.88 ± 3.653
		0.75	0.439 ± 0.419	7.169 ± 2.417
		1.0	0.425 ± 0.397	6.23 ± 1.66
	2.0	0.375	0.545 ± 0.396	10.85 ± 5.617
		0.5	$\mathbf{0.582 \pm 0.303}$	9.561 ± 3.839
		0.75	0.486 ± 0.451	7.4 ± 2.565
		1.0	0.471 ± 0.401	6.31 ± 1.912
Masked	1.0	0.375	0.395 ± 0.461	6.119 ± 1.74
		0.5	0.377 ± 0.417	5.97 ± 1.558
		0.75	0.378 ± 0.411	5.59 ± 1.422
		1.0	0.321 ± 0.389	5.48 ± 1.292
	2.0	0.375	0.484 ± 0.317	5.81 ± 1.501
		0.5	0.428 ± 0.417	5.64 ± 1.466
		0.75	0.412 ± 0.374	5.34 ± 1.305
		1.0	0.353 ± 0.443	5.18 ± 1.268

point. This behavior is possible due to the usage of the *truncated* adaptation, and the results show that on average this adaptation works better when using the EI AF. On the other hand, Figure 5.3b shows that two straight routes managed to obtain a lot of information on their way. Note that the length between measurements is always the same, and, generally speaking, it has been observed that the selected final locations are always on the shore of the water body. A combination of both methods (AFs and adaptation) seems a good way to obtain the best from both, but for initial assessments, EI with truncated adaptation is better since it not only finds new measurement locations after each measurement, but also requires fewer measurements on average to provide the same results. The limitation is due only to the distance between locations, which is a constraint already discussed. The final selection of components for a single ASV is shown in Table 5.10. The kernel also includes the addition of a White Kernel with noise value that needs to be adjusted for simulations and/or real experiments.

Regarding the different hyperparameters, a higher bias towards exploration is sensible since the optimality is based on the knowledge of the model. Most of the results found

Table 5.7 Results for the Scaled Expected Improvement AF.

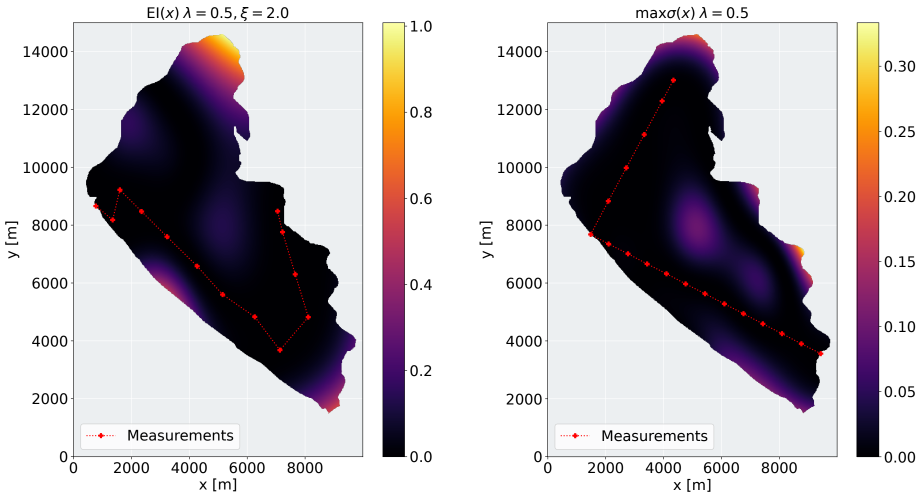
Adaptation f.	ξ	λ	Avg. R^2	Avg. M
Split Path	1.0	0.375	0.46 ± 0.345	17.009 ± 5.232
		0.5	0.474 ± 0.34	13.73 ± 3.987
		0.75	0.438 ± 0.375	9.89 ± 2.592
		1.0	0.426 ± 0.364	8.42 ± 1.955
	2.0	0.375	0.504 ± 0.317	17.36 ± 5.235
		0.5	0.481 ± 0.371	13.83 ± 4.062
		0.75	0.408 ± 0.706	10.01 ± 2.496
		1.0	0.412 ± 0.733	8.4 ± 1.985
Truncated	1.0	0.375	0.503 ± 0.455	11.049 ± 5.22
		0.5	0.487 ± 0.404	9.309 ± 3.839
		0.75	0.514 ± 0.372	7.41 ± 2.478
		1.0	0.484 ± 0.318	6.37 ± 1.82
	2.0	0.375	0.51 ± 0.418	10.81 ± 5.524
		0.5	**0.539 ± 0.352**	9.499 ± 3.918
		0.75	0.519 ± 0.391	7.45 ± 2.535
		1.0	0.446 ± 0.718	6.46 ± 1.9
Masked	1.0	0.375	0.168 ± 0.478	3.4 ± 0.49
		0.5	0.168 ± 0.478	3.4 ± 0.49
		0.75	0.168 ± 0.478	3.4 ± 0.49
		1.0	0.168 ± 0.478	3.4 ± 0.49
	2.0	0.375	0.167 ± 0.478	3.4 ± 0.49
		0.5	0.167 ± 0.478	3.4 ± 0.49
		0.75	0.167 ± 0.478	3.4 ± 0.49
		1.0	0.167 ± 0.478	3.4 ± 0.49

Table 5.8 Results for the maximum uncertainty (Active Learning) strategy.

Adaptation f.	λ	Avg. R^2	Avg. M
Split Path	0.375	0.567 ± 0.286	21.55 ± 3.354
	0.5	**0.581 ± 0.274**	16.928 ± 2.584
	0.75	0.535 ± 0.326	11.67 ± 1.674
	1.0	0.568 ± 0.281	9.49 ± 1.345
Truncated	0.375	0.513 ± 0.353	11.899 ± 5.962
	0.5	0.529 ± 0.311	9.99 ± 3.931
	0.75	0.532 ± 0.356	7.79 ± 2.593
	1.0	0.538 ± 0.296	6.55 ± 1.868

Table 5.9 Summarized results for different AFs according to their best setup. The fourth column presents the mean and standard deviation of the R^2, while the last column the average number of performed measurements M. Both columns consider the final distance traveled of $15000[m]$.

Acq. F.	Adaptation f.	(ξ, λ)	Avg. R^2	Avg. M
PI(x)	truncated	2.0, 0.75	0.508 ± 0.395	7.24 ± 2.653
EI(x)	truncated	2.0, 0.5	0.582 ± 0.303	9.561 ± 3.839
SEI(x)	truncated	2.0, 0.5	0.539 ± 0.352	9.499 ± 3.918
max $\sigma(x)$	split path	λ=0.5	0.581 ± 0.274	16.928 ± 2.584



(a) **(b)**

Figure 5.3 Example Squared Error maps considering the best seed using as utility functions: **(a)** EI(x) and **(b)** max $\sigma(x)$.

Table 5.10 Bayesian optimization Framework component for a single ASV.

GP Kernel	Prior ℓ	σ_n^2	Acq. F.	ξ	Adapt. f.	λ
Squared Exp. + White K.	150	0.01	EI(x)	2.0	truncated	0.5

show better results for the highest ξ value for the different AFs. The high amount of variance found in the tables shows that it greatly depends on the starting location, but for real applications, usually there already exist a few measures that will help in selecting the starting location. Through simulations, empirical foundations of optimal measurement performance have been laid out for efficient exploration. It is proposed that exploration of unknown functions take into account the underlying hyper-parameters of GPs such as the length-scale. Simulations have shown that, similar to the Nyquist–Shannon sampling theorem, the distance between measurements locations for surrogate model acquisition need to be approximately half of a supposed *frequency of similarity* between different locations (namely length-scale ℓ).

5.4 Multi-ASV

In this thesis, only one proposition for multi-agent system has been proposed. Unintentional cooperation has many advantages for the monitoring mission compared to intentional cooperation. The flexibility of execution that it provides is sufficient to select it as the proposed method. Therefore, the Voronoi diagram is used to split or create several regions according to each of the vehicles at the selection time of the measurement location.

Voronoi regions can be described solely by the locations (x, y) of the generators, which in the case of this thesis are the current locations of the ASV if they are available. However, since the Voronoi regions need to be limited in surface area, so that each agent aims to cover a finite area, each region needs to be defined according to their Voronoi vertices, which are fairly easy to obtain. However, additional calculations are needed to fully define the Voronoi regions for all generators. Figure 5.4 shows an example Voronoi region generated considering four different ASVs, using this technique the searching region is limited to the bounds of the map.

Each time a new measurement location is needed, the VD is first generated according to the available ASVs and the boundary of the map. Next, the CC only searches for a possible measurement location within the feasible region and the Voronoi region generated for the vehicle. Figure 5.5 shows some examples of different Voronoi regions according to 3, 4 and 5 ASVs. It shows the robustness against adding and removing ASVs in the middle of a mission, because the regions will be updated to include only the available ASVs as Voronoi generators. It is important to note that if only one vehicle is present in the system, a box “Voronoi” region will be created, which will create the same working scenario as the preceding results in this chapter. Moreover, due to the fact that the selected AF is heavily biased towards explorations, two ASVs will never be in the same location.

5.5 Proposed approach: Multi Water Quality Parameter monitoring using Multiple Autonomous Surface Vehicles

Continuing with the results, in this section, the results for a multi-water quality parameter monitoring mission are presented. In these experiments, a simulated ASV proceeds to measure several WQP simultaneously and lets the CC decide on measurement locations according to a designed system. Recall that Multi-Function Estimation and Multi-Objective Optimization were the two approaches proposed in this thesis. The procedure was to test the methods with their respective components, select the best components for each approach, and then perform a final comparison against each other. Considering multiple WQPs, the results should include different simulated WQP models. In that sense, Table 5.11 shows

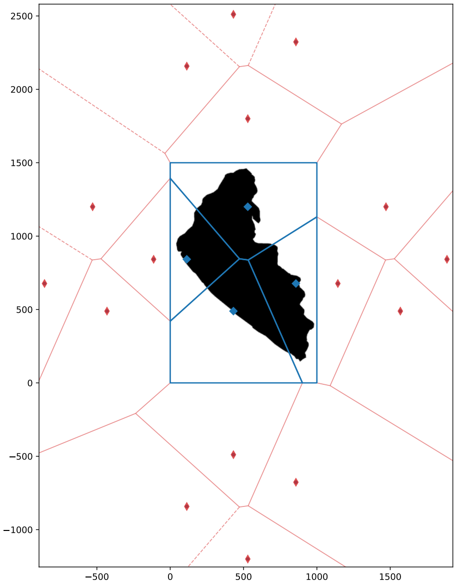


Figure 5.4 Example of bounded VD for the Ypacarai Lake considering 4 (blue diamonds) initial points and their respective mirrored points (red).

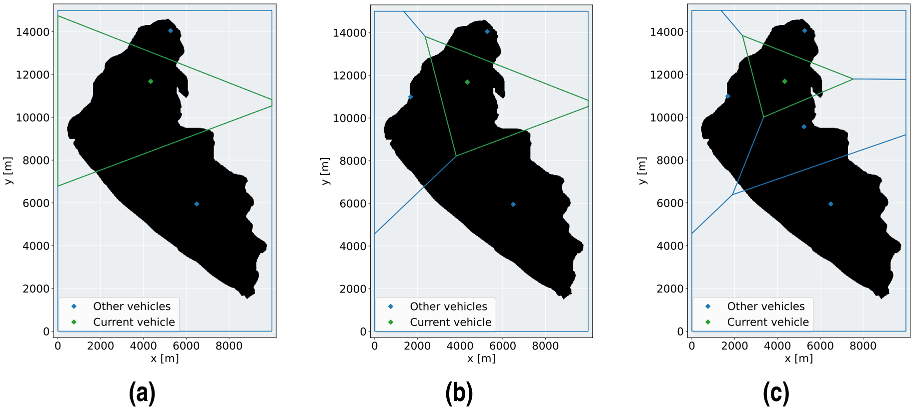


Figure 5.5 Voronoi Partitioning Region in action for (a) 3, (b) 4 and (c) 5 vehicles. Note the different regions generated automatically updates and distributes regions according to the number of available ASVs.

the different combinations of WQP simulated models used, i.e., the different sensors that were used during a mission, where for each simulation model ID, different runs were made according to different initial locations and number of ASVs. Each S_m corresponds to a different WQP model generated using the shekel function with random peaks and locations. Note that there exist 4 groups of number of sensors involved, each one containing 4 different combinations of WQP models so that generalization was effectively tested.

Table 5.11 Different groups of sensors involved.

Set	Sensors Involved
1	$[S0, S1]$
2	$[S2, S3]$
3	$[S4, S5]$
4	$[S6, S7]$
5	$[S0, S1, S2]$
6	$[S3, S4, S5]$
7	$[S6, S7, S8]$
8	$[S0, S4, S9]$
9	$[S0, S1, S6, S7]$
10	$[S2, S3, S4, S5]$
11	$[S2, S4, S6, S8]$
12	$[S6, S7, S8, S9]$

The number of vehicles available were 2, 3 or 4 that had initial locations on the shore of the water body and had a full battery level. The CC used MFE (coupled or decoupled) or MOO (through the Genetic Algorithm) to obtain the best locations for measurements, which were adapted to limit traveling distance. The results are grouped by these strategies. For each of the ASV fleets (2, 3 or 4 ASVs), the mean R^2 scores for 120 different simulations using different sets of WQP simulated models are shown in Table 5.12, Table 5.13, for the coupled and decoupled variations, respectively. Regarding this approach, Multi-Function Estimation, no additional hyperparameters need to be determined.

Regarding the MOO using GA, several hyperparameters needs to be described. Table 5.14 shows the hyperparameters selected. The code was fully implemented using the library *pymoo* [87], and the respective parameters were selected to efficiently explore the feasible search space. In this implementation, the constraints functions are defined according to the map model and the Voronoi region for each vehicle.

Table 5.15 shows the results for 120 simulations considering a MOO framework. The table is distributed in the same way as the MFE variations found above. The main difference using this strategy is that a population of candidate measurement locations is evolved to obtain an optimal Pareto efficient set instead of obtaining the location according to an equally weighted approach (MFE coupled) or to an unilateral weighting approach (MFE decoupled). Figure 5.6 shows an example of the MOO in action where there exists 4 ASVs measuring 4 different WQPs simultaneously. In the example, the WQPs correspond to Figure 4.3 found in Section 4.2.1. All Pareto Efficient points that were obtained are within the Voronoi region for the green ASV generated with the current locations of the ASVs, not the last measurement locations. It is remarkable how the proposed approach can efficiently

Table 5.12 Results for MFE strategy considering coupled method for different number of ASVs and different number of sensors available.

# ASVs	**#** S	**Avg.** R^2	**Avg.** M
2	2	0.928 ± 0.069	27.498 ± 3.294
	3	0.95 ± 0.055	28.673 ± 3.26
	4	0.949 ± 0.035	28.645 ± 3.462
3	2	0.974 ± 0.017	38.922 ± 3.546
	3	0.981 ± 0.009	40.547 ± 3.59
	4	0.979 ± 0.013	41.474 ± 4.78
4	2	0.985 ± 0.01	51.923 ± 3.783
	3	0.989 ± 0.011	53.872 ± 3.696
	4	0.99 ± 0.006	54.096 ± 4.701
Average		0.969 ± 0.039	40.628 ± 10.948

Table 5.13 Results for MFE strategy considering decoupled method for different number of ASVs and different number of sensors available.

# ASVs	**#** S	**Avg.** R^2	**Avg.** M
2	2	0.931 ± 0.067	27.623 ± 3.646
	3	0.943 ± 0.057	27.999 ± 3.563
	4	0.936 ± 0.062	28.174 ± 4.312
3	2	0.975 ± 0.016	38.372 ± 3.089
	3	0.983 ± 0.01	42.048 ± 3.263
	4	0.979 ± 0.014	41.398 ± 4.774
4	2	0.986 ± 0.008	52.922 ± 3.524
	3	0.99 ± 0.008	53.622 ± 4.287
	4	0.988 ± 0.009	52.84 ± 5.344
Average		0.968 ± 0.044	40.487 ± 11.078

obtain candidate optimal measurement locations, and how efficient the system is because is the generated paths are all close to the different initial locations of the ASVs.

It has been observed that evidently, with a larger number of ASVs, a larger number of measurements will be performed, this will inevitably yield better R^2 scores. The obtained GPs manage to correctly fit the data (4 ASVs almost always obtain $R^2 = 0.99$) and approximate the real function with ease. Furthermore, the Voronoi Diagram really enhances robustness and provides scalability. Without the VD, the vehicles will perform measurements on the same regions, and consequently, provide redundant information. With the proposed VD system, the fleet cooperates unintentionally while monitoring. Table 5.16 shows the summarized results of the methods. MOO using GA presents the best result. In this case, it also presents the average greater number of measurements, but it is not so different from the other methods. MOO seems to be more robust regarding initial

Table 5.14 Multi Objective Optimization parameter values.

Parameter	**Value**
Optimizer	NSGA-II [86]
Individual	2D Gene: (x, y)
Population	50
Generations	150
Selection	Binary Tournament size $= 2$
Crossover	Simulated Binary $\eta = 15$; $\text{prob}_{cx} = 0.9$
Mutation	Polynomial Mutation $\eta = 20$; $\text{prob}_{mut} = 0.01$

Table 5.15 Results for MOO strategy using GA for different number of ASVs and different number of sensors available.

# ASVs	**#** S	**Avg.** R^2	**Avg.** M
2	2	0.953 ± 0.027	28.049 ± 2.991
	3	0.949 ± 0.049	28.774 ± 2.894
	4	0.963 ± 0.027	29.273 ± 3.513
3	2	0.979 ± 0.041	40.045 ± 5.002
	3	0.984 ± 0.014	42.747 ± 4.646
	4	0.987 ± 0.01	43.122 ± 4.713
4	2	0.99 ± 0.01	53.053 ± 8.996
	3	0.989 ± 0.014	52.873 ± 10.924
	4	0.987 ± 0.018	51.722 ± 12.033
Average		0.976 ± 0.031	41.073 ± 12.055

conditions, presenting a lower standard deviation value of R^2. In that sense, the proposed method should include the usage of MOO using GA for accomplishing monitoring of multiple WQPs through multiple ASVs using the Bayesian optimization framework.

Table 5.16 Summary of the results using Multi-Water Quality Parameter strategies.

Strategy	**Avg.** R^2	**Avg.** M
MFE Coupled	0.969 ± 0.039	40.628 ± 10.948
MFE Decoupled	0.968 ± 0.044	40.487 ± 11.078
MOO using GA	0.976 ± 0.031	41.073 ± 12.055

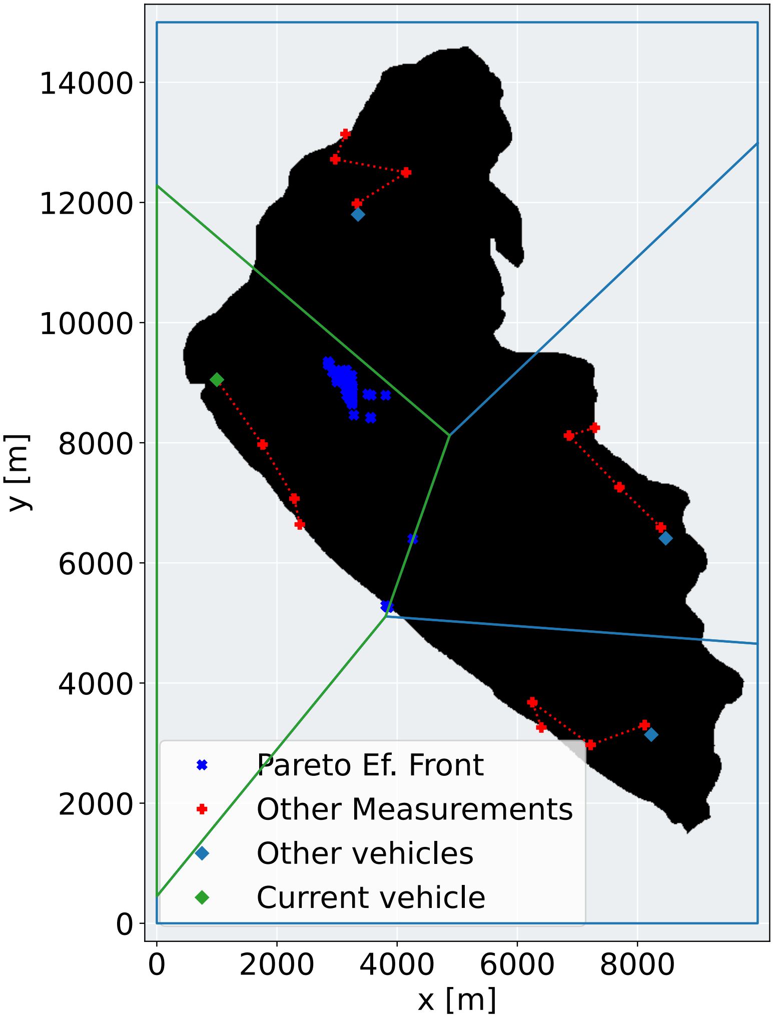


Figure 5.6 Example of the Multi-ASV Multi-WQP system. A Pareto Efficient set is generated according to the current values returned by the selected acquisition function and the constraint function. The next measurement location is selected to be the closest to the current location of the vehicle and is adapted according to the adaptation function.

5.6 Comparison with other methods

Before the results were discussed, a comparison with three different environmental monitoring approaches was performed. The methods are: the Predictive Entropy Search for Multi-Objective Bayesian optimization with Constraints (PESMOC) [71], applied to the monitoring objective, the TSP-Genetic Algorithm for monitoring [56], and finally a lawnmower efficient method for monitoring. The three approaches are modified if possible to achieve maximum exploration and are tested in the same scenario (including ASVs constraints) as the proposed method. Evidently, the proposed system (Figure 3.6) is the same, with the difference in the evaluation component of the acquisition function. The three methods are defined as follows.

1. PESMOC for Environmental Monitoring:

   In this work, we use the method proposed in [71] but with some modifications to ensure monitoring. It consists of obtaining the difference between the logarithm of uncertainty of a predictive distribution (PD) σ_i^{PD} and the average of logarithms of uncertainties of Conditioned PD σ_i^{CPD} (conditioned to $x|\mathscr{X}*_{(m)}$). $\mathscr{X}*_{(m)}$ is one of the m different locations of a supposed Pareto set. For a complete explanation of PESMOC, see [71]. The PES expression is taken directly from the work, considering deterministic constraints, and has the form of

   $$\alpha \approx \sum_{s=1}^{S}\left(\log \sigma_s^{PD}(x) - \frac{1}{MS}\sum_{m=1}^{MS}\log \sigma_i^{CPD}(x|\mathscr{X}*_{(m)})\right) \tag{5.2}$$

   This expression can be evaluated with the coupled or decoupled strategy. As shown in the expression above, the coupled evaluation sums up the differences for each AF objective α_s. The decoupled version considers one difference at a time.The parameters that suffered changes in order to fit the purpose of exploration are as follows.

   - Pareto set $\mathscr{X}*$: Since the objective of this work can be thought of as minimizing uncertainty, the Pareto set is taken as the positions where the sum of the predicted standard deviations reaches its maximum values.
   - Conditioning $p(\mathbf{y}|D,\mathbf{x},\mathscr{X}*)$: Conditioning is carried out through a cloned GP model to include a supposed evaluation according to the items of the Pareto set.
   - Monte Carlo Sampling MS: For efficient evaluations, only one point of the Pareto set is used. Therefore, the number of Monte Carlo samples is reduced to one. Indeed, this sacrifices accuracy but definitely improves computational efficiency, which has been observed to be less efficient than our method because it needs to calculate the GP Regression twice for each water quality parameter or objective.

   Voronoi regions were used to define the constraints, so that each vehicle could only obtain a PESMOC value within its coverage region. With $M = 1$, as predicted, and using the same simulation sensors as in the evaluation of the proposed method with the best parameter, $\lambda = 0.5$ the results are shown in Table 5.17.

2. TSP-based Environmental Monitoring:

   In [56], a set of 60 waypoints were defined on the shore of Ypacarai Lake. Subsequently, the best TSP solution (waypoint visit order) was found by a GA evolved to

Table 5.17 Results for PESMOC strategy for different number of ASVs and different number of sensors available.

# ASVs	# S	Avg. R^2	Avg. M
2	2	0.907 ± 0.056	26.824 ± 4.727
	3	0.91 ± 0.057	27.174 ± 4.358
	4	0.918 ± 0.066	28.021 ± 4.213
3	2	0.983 ± 0.01	41.82 ± 4.161
	3	0.984 ± 0.01	43.02 ± 4.49
	4	0.983 ± 0.008	42.721 ± 4.612
4	2	0.995 ± 0.002	54.232 ± 5.024
	3	0.995 ± 0.002	54.662 ± 3.851
	4	0.995 ± 0.003	55.469 ± 3.168
Average		0.962 ± 0.052	41.269 ± 12.137

optimize the exploration of Ypacarai Lake. This method was discussed in Chapter 2. Contrary to the continuous measurement approach stated in the aforementioned work [56], for comparison, the monitoring system is modified so that the vehicle can take measurements only while the ASV is not moving.

The distance between the measurement locations is the same as that proposed in this work, which is based on the length scale. Therefore, the ASV travels from point to point, making measurements every $l = \lambda \times \min\{\ell_s\}$ meters. Whenever the total distance traveled reaches 15,000[m], the ASV stops, performs a final measurement, and the mission ends. To account for multiple vehicles, the available shoreline points were divided into groups according to the previously obtained visit order. This is the only difference regarding the usage of multiple ASVs w.r.t. the original proposed system. The results are shown in Table 5.18.

3. Lawnmower applied for monitoring/exploration:

The classic lawnmower implementation is also a straightforward approach that seems to be efficient for monitoring, due to its definition. For comparison, the lawnmower strategy found in [93] was used as part of the acquisition function component, obtaining straight paths that will intend to cover the water body in this problem. The generation of paths considers the Voronoi regions created at execution time and generates waypoint locations that are equally spaced according to the expected prior length scale of the GP. The classic implementation can only deliver paths that are left-to-right and bottom-up (or vice versa, for both pairs of directions), so the vehicles necessarily need to initially be location on the far-most edge of the water body. The results can then only be obtained for multiple sets of simulations that vary the WQP maps but not the initial vehicle locations. The results are available in Table 5.19.

None of the comparison methods performed on average better than the proposed method. In Table 5.20, a summary of the comparison methods can be observed, and none of them

Table 5.18 Results for TSP-based GA strategy for different number of ASVs and different number of sensors available.

# ASVs	# S	Avg. R^2	Avg. M
2	2	0.837 ± 0.086	32.522 ± 3.344
	3	0.848 ± 0.082	33.574 ± 3.065
	4	0.853 ± 0.082	34.722 ± 2.923
3	2	0.891 ± 0.107	50.171 ± 4.017
	3	0.902 ± 0.093	51.223 ± 3.636
	4	0.903 ± 0.093	51.898 ± 3.491
4	2	0.962 ± 0.019	60.2 ± 0.4
	3	0.96 ± 0.023	60.1 ± 0.944
	4	0.958 ± 0.022	60.125 ± 0.331
Average		0.902 ± 0.088	48.281 ± 11.39

Table 5.19 Results for Lawnmower strategy for different number of ASVs and different number of sensors available.

# ASVs	# S	Avg. R^2	Avg. M
2	2	0.946 ± 0.036	28.9 ± 2.447
	3	0.957 ± 0.023	29.75 ± 2.808
	4	0.953 ± 0.024	30.2 ± 2.421
3	2	0.815 ± 0.109	25.15 ± 3.864
	3	0.848 ± 0.089	26.25 ± 3.819
	4	0.853 ± 0.088	26.9 ± 2.488
4	2	0.739 ± 0.132	16.75 ± 4.448
	3	0.779 ± 0.105	18.25 ± 5.29
	4	0.791 ± 0.089	18.5 ± 5.045
Average		0.853 ± 0.115	24.517 ± 6.258

shows better behavior than any of the methods shown in Table 5.16, which contains the results of the proposed method. Of course, because lawnmower and TSP-based using GA are offline methods, they cannot obtain similar R^2 values. PESMOC, on the other hand, is an online method that reached a satisfactory average value but is not as good as the proposed method. Figure 5.7 shows a comparison between the methods with respect to the distance of travel. To obtain this figure, the mean values of R^2 and the standard deviations were obtained considering the different measurements performed for each method during each simulation. It can be regarded as the average R^2 per distance obtained using the different methods. It shows that the MOO using the GA method is not only at the end of the experiments, but is better during the missions because it always presents a better coefficient of determination. Not that the TSP-based method and the lawmower methods have different starting values because their initial positions are selected according to the generated offline paths rather than random initial locations at the shore of the water body.

Table 5.20 Summary of the comparison method results.

Strategy	Avg. R^2	Avg. M
PESMOC	0.962 ± 0.052	41.269 ± 12.137
TSP-based GA	0.902 ± 0.088	48.281 ± 11.39
Lawnmower	0.853 ± 0.115	24.517 ± 6.258

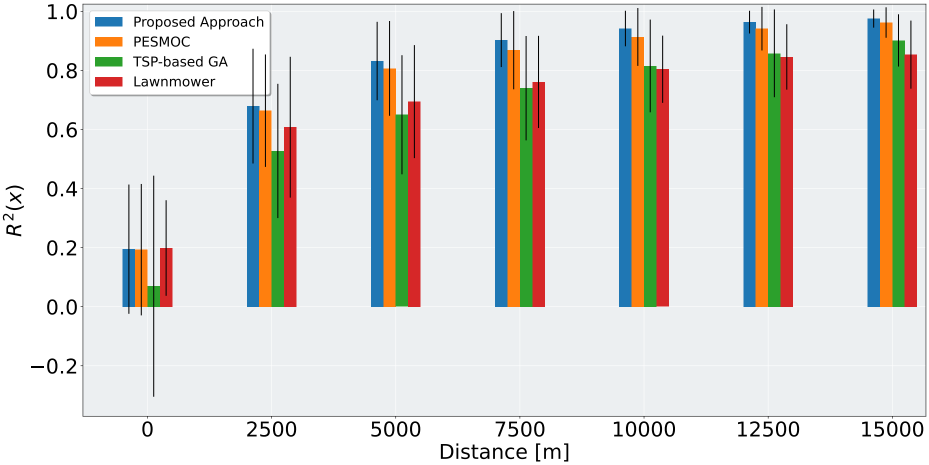


Figure 5.7 Average interpolated R^2 value obtained per distance for the proposed approach, as well as the comparison approaches applied to online monitoring.

5.7 Summary of the Results

After the obtained results were observed, analyzed, and discussed, a complete summary of them has been done and is as following:

- *Gaussian Process* can be used as surrogate models, since they can efficiently provide regressions using a limited amount of data. With our proposal, as well as with the others, the average coefficient of determination is close to 1, implying that the selected model is very good when approximating unknown models.
- *Voronoi Regions* provide a good systematic approach for multiagent cooperation. This approach only allows for a scalable and robust region partitioning and additionally, in the case of our approach, it can be updated to consider only available ASVs, which is very helpful whenever some agents can be added or removed from the system, in the middle of a mission.
- The *Multi-Objective problem* is appropriately addressed, and is, to best knowledge of the author, one of the firsts works to perform intelligent online water quality

parameter measuring for model obtainment. This work expands the Bayesian optimization method to account for multiple objectives (multiple acquisition/utility functions).

- *Efficiency of Multi-Function Estimation:* MFE using multiple vehicles is a quick method for solving multi-objective problems such as this case. Provides the same level of performance as the MOO approach.
- *Efficiency of Multi-Objective Optimization:* It is one of the first approaches that intelligently explore a large-scale environment in the search of data acquisition. The path, which are generated during the mission, is adapted to any scenario and considers both: the acquired data and the endurance of the agent. According to the performed simulations, the MOO approach showed the best results, not only considering the average final R2S obtained, but also when considering the traveled distance.
- Comparison with *PESMOC*: PESMOC is the online method that mos closely resembles the proposed system in this thesis, but the results are not as good. Insights of the method show that it can be trapped in considering points that are too far away, failing to obtain the surrounding behavior but only the behavior between a starting location and the other edges of the map.
- Comparison with *Genetic Algorithm TSP-based coverage approach*: Focusing on the fact that for real applications the measurements should be performed in a discrete manner, the method proposed in this work performs better. As in the LM, this algorithm presents offline paths, so the same differences in efficiency can also be concluded.
- Comparison with *Lawnmower*: Specifically, with this monitoring approach, our method focuses on efficient data acquisition, rather than global, offline, geometric-based data acquisition. The LM method does not account for whether the obtained data fit better or worse to the underlying surrogate model.

6 Conclusion and Future Work

Maybe we've spent too long trying to figure this out with theory.

Amelia Brand, Interstellar, 2014

6.1 Conclusion

This thesis proposed one of the first BO methods for environmental informative path planning of multiple ASV, to the best knowledge of the author, and it aims to minimize the uncertainties of multiple WQPs. The proposed system includes the usage of a covariance function for a GP that fits the expected Water Quality Parameter behaviors. In that sense, multiple kernels were tested and compared through simulations to select the best one. The simulations were designed so that they resemble real life limitations. Therefore, an ASV simulator was designed as part of a modular component so that the main system (centralized coordinator) is unaware of the ASVs that are executing the mission. Among the different kernels, the SE (summed with a white kernel) fitted the shekel function, including noisy measurements.

Following the BO framework, several AFs were also considered for their use, but were modified and adapted to the specific constraints of ASV monitoring. Specifically, three different adaptations were proposed to deal with the mobility (and energy) restrictions of ASVs. These adaptations included a hyperparameter that dealt with energy constraints. Instead of considering that this hyperparameter should be *parameter*, it was considered a variable related to the kernel hyperparameter length scale. Among the different proportions of length scales tested, the best results were generally less than or equal to half of the fit length scale value, which is consistent with the Nyquist-Shannon sampling theorem, because it limits the distance between measurement locations or samples in the feasible space. Evidently, various "ground truths" were used to obtain the best results, and various initial seeds were also used to ensure the reliability and statistical validity of the system. Results have shown that, considering one WQP and one ASV, the best method consists of selecting the Expected Improvement AF highly biased towards exploration, whose best proposed measurement location was modified according to the truncated adaptation with half of the length scale as maximum traveling distance. Compared to a simplistic Active learning approach, the proposed system shows equal values of the coefficient of determination, but the number of measurement locations is drastically lower, which is a desirable characteristic.

Today, not only can ASVs be equipped with a multi-WQP sensor system, but multiple ASVs can also be acquired/developed because of their relative economic factor. This was not feasible until recently, so related works do not consider this possibility, and therefore they are not developed to be a Multi-WQP Multi-ASV system. In this thesis, this assumption was a possibility; hence, the proposed system takes advantage of this and improves the monitoring system. A multi-function estimation using Bayesian optimization approach for environmental monitoring with multiple ASVs was presented. First, following a non-intentional cooperation paradigm, Voronoi regions were defined for each of the available ASVs. Afterward, considering the region of an ASV, a measurement location was obtained according to the functions previously mentioned and the fusion of WQP surrogate functions: Since multiple WQP values are obtained, multiple AFs can also be obtained, which can be combined so that a new function presents a new best measurement location. Multi-Objective Optimization can also be used to address the existence of various water quality parameter models that need to be obtained. A quick and efficient implementation of NSGA-2 was used to obtain, for an ASV, a Pareto-efficient set of locations that were within a Voronoi region. Since in this setting, any location is a valid optimal next measurement location, the closest one to the current position of the ASV was selected, and after adapting this location, the new adapted optimal measurement location was selected to be the goal of the ASV, where new measurements were performed.

Furthermore, we compared the proposed monitoring approach with other alternatives found in the literature, such as the GA-based exploration algorithm and the lawnmower method. The results obtained demonstrated the validity of the proposed approach, since it clearly outperforms the other techniques in the simulated scenarios. The proposed method can also be improved with a multi-agent system composed of several ASVs. Using a centralized coordination could decrease the total distance traveled by each ASV through multi-objective optimization. This multi-objective approach has proven to be better than other methods. The application of this is shown to be an efficient and reliable approach when the mission is to monitor a large-scale water environment while also measuring its quality parameters. Since the objectives consist of minimizing the error of these maps, the comparison was made considering the coefficient of determination, and it was observed that the system has proven to be efficient, obtaining very good values. In addition, it has been proven to be robust against redundancy, as search regions are continuously adapting to the locations of available ASVs. Furthermore, the proposed approach has been shown to be scalable with the number of vehicles.

Compared to other similar monitoring methods, our proposal not only outperforms the PESMOC, GA, and LM methods at the end of the mission, but also during the whole mission. It is also more efficient with respect to measurement performing with coverage and patrolling approaches that aim to obtain initial assessments of multiple WQPs simultaneously for real scenarios. The results show that the monitoring of WPQs should not be done offline but by considering the newly collected data to decide on new measurement locations, regardless of the number of agents involved or the number of measured WQPs.

6.2 Future Work

To ensure that further improvement can be made, the future directions of this work can include dynamic modeling of WQPs and time-based Bayesian optimization approaches because recurrent monitoring missions can be performed. Therefore, an efficient system that can include loss of information regarding the elapsed time can be developed. Dynamic systems should be robust against sudden changes in WQPs. In that sense, and taking advantage of the fact that kernels can be designed, a temporal-dependent kernel can be designed to account for the mentioned dynamism. Kernels are usually stationary, but considering time as an extra dimension can be effective and maintain the output stationary.

Multi-Output Gaussian Distributions can also be used for WQP monitoring. While in this work different kernels are used to account for the different WQPs, a Multi-Output GP can be designed so that not only the system is more robust, but also the GP can have feedback and find correlations between WPQs. This will help, for example, when some sensor data is not available or the sensor system could not obtain it, because the output WQPs will also depend on themselves, or when a full WQP can be obtained using other methods like temperature from satellite images, etc. In this case, the WQP surrogate models will have less uncertainty about their values with fewer measurements.

Additional tests can also be performed on ASV availability or communication constraints. ASVs can, indeed, fail in the middle of a mission. Although the system is designed to be robust regarding this issue, experimental results need to demonstrate the robustness of the system. Real experiments in large-scale scenarios such as Mar Menor or open seas can also lead to the decision of using other methods and approaches when the sensors are too noisy or the waters are in constant movement. An heterogeneous system can also be designed so that it includes the usage of the information provided by fixed monitoring stations as well as aerial vehicles for faster monitoring. This heterogeneous approach can also include different sensors for different ASVs, considering that WQPs should be somehow correlated and that there exist some sensors that are not yet so economically viable. This generalizes the problem to a multi-heterogeneous constrained ASV system for obtaining multiple WQPs.

Finally, a combination of myopic and nonmyopic monitoring missions can be done as the work in [56] suggests. First, creating an efficient informative path planning between harbors obtaining a set of future measurement locations (non-myopic approach). These measurement locations will be ordered and visited accordingly, and the surrogate models will be updated with any new obtained data. Afterwards, a decision making system will update the next measurement location (myopic approach) so that the multiple ASVs do not drive to far away from the initial planned location but to a nearby location that has the potential to add more information to the system. Indeed, the information entropy will be very useful in the mentioned scenario and can improve the results.

6.3 Publication List

This thesis is a work that made possible a sequence of several manuscripts and conference papers that have been previously published by the author. The complete list is as follows.

- Peralta, F., Arzamendia, M., Gregor, D., Cikel, K., Santacruz, M., Reina, D. G., & Toral, S. (2019, November). Development of a simulator for the study of path planning of an autonomous surface vehicle in lake environments. In 2019 IEEE CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON) (pp. 1-6). IEEE. [94]
- Peralta, F., Arzamendia, M., Gregor, D., Reina, D. G., & Toral, S. (2020). A comparison of local path planning techniques of autonomous surface vehicles for monitoring applications: The ypacarai lake case-study. Sensors, 20(5), 1488. [54]
- Samaniego, F. P., Reina, D. G., Toral, S. L., Arzamendia, M., & Gregor, D. O. (2021). A bayesian optimization approach for water resources monitoring through an autonomous surface vehicle: The ypacarai lake case study. IEEE Access, 9, 9163-9179.[95]
- Peralta, F., Reina, D. G., Toral, S., Arzamendia, M., & Gregor, D. (2021). A bayesian optimization approach for multi-function estimation for environmental monitoring using an autonomous surface vehicle: Ypacarai lake case study. Electronics, 10(8), 963.[96]
- Luis, S. Y., Peralta, F., Córdoba, A. T., del Nozal, Á. R., Toral, S., & Reina, D. G. (2022). An evolutionary multi-objective path planning of a fleet of ASVs for patrolling water resources. Engineering Applications of Artificial Intelligence, 112, 104852.[62]
- F. Peralta, D. G. Reina, and S. Toral, "Water quality online modeling using multi-objective and multi-agent bayesian optimization with region partitioning," in Engineering Applications of Artificial Intelligence. Elsevier, 2022 under review.[97]
- Peralta, F., Yanes, S., Reina, D. G., & Toral, S. (2021, June). Monitoring Water Resources through a Bayesian Optimization-based Approach using Multiple Surface Vehicles: The Ypacarai Lake Case Study. In 2021 IEEE Congress on Evolutionary Computation (CEC) (pp. 1511-1518). IEEE. [98]
- Peralta, F., Reina, D. G., & Toral, S. Towards an Online Water Quality Monitoring system of Dynamic Environments using an Autonomous Surface Vehicle. In International Conference on Optimization and Learning OLA'2022, 2022 in press. [99]
- Federico Peralta, Michael Pearce, Matthias Poloczek, Daniel Gutierrez Reina, Sergio Toral, & Juergen Branke. 2022. Multi-Objective Path Planning for Environmental Monitoring using an Autonomous Surface Vehicle. In Genetic and Evolutionary Computation Conference Companion (GECCO '22 Companion), July 9–13, 2022, Boston, MA, USA. ACM, New York, NY, USA, 5 pages. [100]

Bibliography

[1] M. Sargen, "Biological roles of water: Why is water necessary for life?" https://sitn.hms.harvard.edu/uncategorized/2019/biological-roles-of-water-why-is-water-necessary-for-life/, 2019, [Online; accessed 17-05-2022].

[2] T. Malone and A. Newton, "The globalization of cultural eutrophication in the coastal ocean: Causes and consequences," *Frontiers in Marine Science*, vol. 7, p. 670, 2020.

[3] M. L. Moreira, A. Gregorio, L. Hinegk, A. Salvadore, G. Zolezzi, F. Hölker, S. M. Domecq, A. Roger, M. Bocci, S. Carrer, L. D. Nat, J. Escribá *et al.*, "Eutrophication, research and management history of the shallow ypacaraí lake (paraguay)," *Sustainability*, vol. 10, no. 7, p. 2426, 2018.

[4] Paraguay.com, "Emiten alerta por cianobacterias en el lago ypacaraí," https://www.paraguay.com/nacionales/emiten-alerta-por-cianobacterias-en-el-lago-ypacarai-98094, 2013, [Online; accessed 17-05-2022].

[5] M. Mas Monsonis and M. Alcaraz Aparicio, "Uso de la teledetección y los sig en la vigilancia de la calidad del agua: aplicación al mar menor," *PFC/TFG-Escuela de Ingeniería de Caminos, Canales y Puertos y de Ingeniería de Minas*, 2017.

[6] E. Henao, J. R. Cantera, and P. Rzymski, "Conserving the amazon river basin: The case study of the yahuarcaca lakes system in colombia," *Science of The Total Environment*, vol. 724, p. 138186, 2020.

[7] A. Dutta, A. Bhattacharya, O. P. Kreidl, A. Ghosh, and P. Dasgupta, "Multi-robot informative path planning in unknown environments through continuous region partitioning," *International Journal of Advanced Robotic Systems*, vol. 17, no. 6, p. 1729881420970461, 2020.

[8] K. Ma, Z. Ma, L. Liu, and G. S. Sukhatme, "Multi-robot informative and adaptive planning for persistent environmental monitoring," in *Distributed Autonomous Robotic Systems*. Springer, 2018, pp. 285–298.

[9] M. Arzamendia, D. Gregor, D. G. Reina, and S. Toral, “An evolutionary approach to constrained path planning of an autonomous surface vehicle for maximizing the covered area of ypacarai lake,” *Soft Computing*, vol. 23, no. 5, pp. 1723–1734, 2019.

[10] S. Y. Luis, D. G. Reina, and S. T. Marín, “A deep reinforcement learning approach for the patrolling problem of water resources through autonomous surface vehicles: The ypacarai lake case,” *IEEE Access*, vol. 8, pp. 204 076–204 093, 2020.

[11] S. Y. Luis, D. G. Reina, and S. L. Toral, “A multiagent deep reinforcement learning approach for path planning in autonomous surface vehicles: The ypacaraí lake patrolling case,” *IEEE Access*, vol. 9, pp. 17 084–17 099, 2021.

[12] C. Chauvin-Hameau, “Informative path planning for algae farm surveying,” 2020.

[13] Z. Peng, J. Wang, D. Wang, and Q.-L. Han, “An overview of recent advances in coordinated control of multiple autonomous surface vehicles,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 732–745, 2020.

[14] M. Arzamendia, D. Gregor, D. G. Reina, and S. Toral, “A path planning approach of an autonomous surface vehicle for water quality monitoring using evolutionary computation,” in *Technology for Smart Futures*. Springer, 2018, pp. 55–73.

[15] M. Arzamendia, I. Espartza, D. G. Reina, S. Toral, and D. Gregor, “Comparison of eulerian and hamiltonian circuits for evolutionary-based path planning of an autonomous surface vehicle for monitoring ypacarai lake,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 4, pp. 1495–1507, 2019.

[16] N. Karapetyan, A. Braude, J. Moulton, J. A. Burstein, S. White, J. M. O’Kane, and I. Rekleitis, “Riverine coverage with an autonomous surface vehicle over known environments,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 3098–3104.

[17] G. Hitz, E. Galceran, M. Garneau, F. Pomerleau, and R. Siegwart, “Adaptive continuous-space informative path planning for online environmental monitoring,” *Journal of Field Robotics*, vol. 34, no. 8, pp. 1427–1449, 2017.

[18] L. Jin, J. Rückin, S. H. Kiss, T. Vidal-Calleja, and M. Popović, “Adaptive-resolution gaussian process mapping for efficient uav-based terrain monitoring,” *arXiv preprint arXiv:2109.14257*, 2021.

[19] P. Morere, R. Marchant, and F. Ramos, “Sequential bayesian optimization as a pomdp for environment monitoring with uavs,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 6381–6388.

[20] C. Williams and C. Rasmussen, *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006, vol. 2.

[21] 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 *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 2124–2130.

[22] Z. Liu, Y. Zhang, X. Yu, and C. Yuan, “Unmanned surface vehicles: An overview of developments and challenges,” *Annual Reviews in Control*, vol. 41, pp. 71–93, 2016.

[23] C. Zhou, S. Gu, Y. Wen, Z. Du, C. Xiao, L. Huang, and M. Zhu, “The review unmanned surface vehicle path planning: Based on multi-modality constraint,” *Ocean Engineering*, vol. 200, p. 107043, 2020.

[24] L. Bottarelli, M. Bicego, and A. F. J. Blum, “Orienteering-based informative path planning for environmental monitoring,” *Engineering Applications of Artificial Intelligence*, vol. 77, pp. 46–58, 2019.

[25] L. Iftekhar, H. N. Rahman, and I. Rahman, “Area coverage algorithms for networked multi-robot systems,” in *New Developments and Advances in Robot Control*. Springer, 2019, pp. 301–320.

[26] Z. Zhen, D. Xing, and C. Gao, “Cooperative search-attack mission planning for multi-uav based on intelligent self-organized algorithm,” *Aerospace Science and Technology*, vol. 76, pp. 402–411, 2018.

[27] C. Sampedro, H. Bavle, J. L. Sanchez-Lopez, R. A. S. Fernández, A. Rodríguez-Ramos, M. Molina, and P. Campoy, “A flexible and dynamic mission planning architecture for uav swarm coordination,” in *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2016, pp. 355–363.

[28] T. T. Mac, C. Copot, D. T. Tran, and R. De Keyser, “Heuristic approaches in robot path planning: A survey,” *Robotics and Autonomous Systems*, vol. 86, pp. 13–28, 2016.

[29] H. Shin and J. Chae, “A performance review of collision-free path planning algorithms,” *Electronics*, vol. 9, no. 2, p. 316, 2020.

[30] I. Chaari, A. Koubaa, H. Bennaceur, A. Ammar, M. Alajlan, and H. Youssef, “Design and performance analysis of global path planning techniques for autonomous mobile robots in grid environments,” *International Journal of Advanced Robotic Systems*, vol. 14, no. 2, p. 1729881416663663, 2017.

[31] C. Zammit and E.-J. Van Kampen, “Comparison between a* and rrt algorithms for uav path planning,” in *2018 AIAA guidance, navigation, and control conference*, 2018, p. 1846.

[32] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, “A survey of motion planning and control techniques for self-driving urban vehicles,” *IEEE Transactions on intelligent vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

[33] H. Abraham, B. Seppelt, B. Mehler, and B. Reimer, “What’s in a name: Vehicle technology branding & consumer expectations for automation,” in *Proceedings of the 9th international conference on automotive user interfaces and interactive vehicular applications*, 2017, pp. 226–234.

[34] C. Wickens, “Automation stages & levels, 20 years after,” *Journal of Cognitive Engineering and Decision Making*, vol. 12, no. 1, pp. 35–41, 2018.

[35] P. Tokekar, J. Vander Hook, D. Mulla, and V. Isler, “Sensor planning for a symbiotic uav and ugv system for precision agriculture,” *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1498–1511, 2016.

[36] C. Liu, Q. Mao, X. Chu, and S. Xie, “An improved a-star algorithm considering water current, traffic separation and berthing for vessel path planning,” *Applied Sciences*, vol. 9, no. 6, p. 1057, 2019.

[37] L. Bayındır, “A review of swarm robotics tasks,” *Neurocomputing*, vol. 172, pp. 292–321, 2016.

[38] A. Kapoutsis, “Towards a fully autonomous and cooperative deployment of multi-robot teams for exploration and coverage in unknown or partially known environments,” Ph.D. dissertation, $\Delta\eta\mu o\kappa\rho\acute{\iota}\tau\epsilon\iota o$ $\Pi\alpha\nu\epsilon\pi\iota\sigma\tau\acute{\eta}\mu\iota o$ $\Theta\rho\acute{\alpha}\kappa\eta\varsigma$ (ΔΠΘ). $\Sigma\chi o\lambda\acute{\eta}$ $\Pi o\lambda\upsilon\tau\epsilon\chi\nu\iota\kappa\acute{\eta}$. $\mathrm{T}\mu\acute{\eta}\mu\alpha$ $\mathrm{H}\lambda\epsilon\kappa\tau\rho o\lambda\acute{o}\gamma\omega\nu$ …, 2017.

[39] D. Brandtner and M. Saska, “Coherent swarming of unmanned micro aerial vehicles with minimum computational and communication requirements,” in *2017 European Conference on Mobile Robots (ECMR)*. IEEE, 2017, pp. 1–6.

[40] J. Alonso-Mora, S. Baker, and D. Rus, “Multi-robot formation control and object transport in dynamic environments via constrained optimization,” *The International Journal of Robotics Research*, vol. 36, no. 9, pp. 1000–1021, 2017.

[41] T. Yang, S. Hsiung, K. P. CH. Kuo, YD. Tsai, Y. Hsieh, Z. Shen, J. Feng, and C. Kuo, “Development of unmanned surface vehicle for water quality monitoring and measurement,” in *2018 IEEE International Conference on Applied System Invention (ICASI)*. IEEE, 2018, pp. 566–569.

[42] J. Paez, J. Villa, J. Cabrera, and E. Yime, “Implementation of an unmanned surface vehicle for environmental monitoring applications,” in *2018 IEEE 2nd Colombian Conference on Robotics and Automation (CCRA)*. IEEE, 2018, pp. 1–6.

[43] J. Sánchez-García, J. García-Campos, M. Arzamendia, D. G. Reina, S. Toral, and D. Gregor, “A survey on unmanned aerial and aquatic vehicle multi-hop networks: Wireless communications, evaluation tools and applications,” *Computer Communications*, vol. 119, pp. 43–65, 2018.

[44] V. A. Jorge, R. Granada, R. G. Maidana, D. A. Jurak, G. Heck, A. P. Negreiros, D. H. Dos Santos, L. M. Gonçalves, and A. M. Amory, “A survey on unmanned surface vehicles for disaster robotics: Main challenges and directions,” *Sensors*, vol. 19, no. 3, p. 702, 2019.

[45] Q. Yang, S.-J. Jang, and S.-J. Yoo, “Q-learning-based fuzzy logic for multi-objective routing algorithm in flying ad hoc networks,” *Wireless Personal Communications*, pp. 1–24, 2020.

[46] D. Angley, B. Ristic, W. Moran, and B. Himed, “Search for targets in a risky environment using multi-objective optimisation,” *IET Radar, Sonar & Navigation*, vol. 13, no. 1, pp. 123–127, 2018.

[47] S. Sawadsitang, D. Niyato, P. S. Tan, P. Wang, and S. Nutanong, “Multi-objective optimization for drone delivery,” in *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*. IEEE, 2019, pp. 1–5.

[48] K. I. Kilic, O. Gemikonakli, and L. Mostarda, “Multi-objective priority based heuristic optimization for region coverage with uavs,” in *International Conference on Advanced Information Networking and Applications*. Springer, 2020, pp. 768–779.

[49] Z. Gosiewski and K. Kwaśniewski, “Time minimization of rescue action realized by an autonomous vehicle,” *Electronics*, vol. 9, no. 12, p. 2099, 2020.

[50] C. Ju and H. I. Son, “Multiple uav systems for agricultural applications: control, implementation, and evaluation,” *Electronics*, vol. 7, no. 9, p. 162, 2018.

[51] R. Almadhoun, T. Taha, L. Seneviratne, and Y. Zweiri, “A survey on multi-robot coverage path planning for model reconstruction and mapping,” *SN Applied Sciences*, vol. 1, no. 8, p. 847, 2019.

[52] K. Yu, J. O’Kane, and P. Tokekar, “Coverage of an environment using energy-constrained unmanned aerial vehicles,” in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 3259–3265.

[53] L. Schmid, M. Pantic, R. Khanna, L. Ott, R. Siegwart, and J. Nieto, “An efficient sampling-based method for online informative path planning in unknown environments,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1500–1507, 2020.

[54] F. Peralta, M. Arzamendia, D. Gregor, D. G. Reina, and S. Toral, “A comparison of local path planning techniques of autonomous surface vehicles for monitoring applications: The ypacarai lake case-study,” *Sensors*, vol. 20, no. 5, p. 1488, 2020.

[55] Itaipú, “Centro internacional de hidroinformática,” https://hidroinformatica.itaipu.gov.py/, 2016, accessed: 2020-10-15.

[56] M. Arzamendia, D. Gutierrez, S. Toral, D. Gregor, E. Asimakopoulou, and N. Bessis, “Intelligent online learning strategy for an autonomous surface vehicle in lake environments using evolutionary computation,” *IEEE Intelligent Transportation Systems Magazine*, vol. 11, no. 4, pp. 110–125, 2019.

[57] S. Yanes, D. Gutiérrez-Reina, and S. Toral, "A dimensional comparison between evolutionary algorithm and deep reinforcement learning methodologies for autonomous surface vehicles with water quality sensors," *Sensors*, vol. 21, no. 8, p. 2862, 2021.

[58] M. J. T. Kathen, I. J. Flores, and D. G. Reina, "An informative path planner for a swarm of asvs based on an enhanced pso with gaussian surrogate model components intended for water monitoring applications," *Electronics*, vol. 10, no. 13, p. 1605, 2021.

[59] M. J. Ten Kathen, I. J. Flores, D. G. Reina, and A. T. Córdoba, "Autonomous monitoring system for water resources based on pso and gaussian process," in *2021 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2021, pp. 1777–1784.

[60] M. J. Ten Kathen, I. J. Flores, and D. G. Reina, "A comparison of pso-based informative path planners for autonomous surface vehicles for water resource monitoring," in *7th International Conference on Machine Learning Technologies (ICMLT) (ICMLT 2022)*. ACM, 2022 in press.

[61] M. J. Ten Kathen, D. G. Reina, and I. J. Flores, "A comparison of pso-based informative path planners for detecting pollution peaks of the ypacarai lake with autonomous surface vehicles," in *International Conference on Optimization and Learning OLA'2022*, 2022 in press.

[62] S. Y. Luis, F. Peralta, A. T. Córdoba, Á. R. del Nozal, S. T. Marín, and D. G. Reina, "An evolutionary multi-objective path planning of a fleet of asvs for patrolling water resources," *Engineering Applications of Artificial Intelligence*, vol. 112, p. 104852, 2022.

[63] M. Popović, T. Vidal-Calleja, G. Hitz, J. Chung, I. Sa, R. Siegwart, and J. Nieto, "An informative path planning framework for uav-based terrain monitoring," *Autonomous Robots*, pp. 1–23, 2020.

[64] Y. Qiming, Z. Jiandong, and S. Guoqing, "Modeling of uav path planning based on imm under pomdp framework," *Journal of Systems Engineering and Electronics*, vol. 30, no. 3, pp. 545–554, 2019.

[65] R. Marchant and F. Ramos, "Bayesian optimisation for informative continuous path planning," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 6136–6143.

[66] R. Martinez-Cantin, N. D. Freitas, E. Brochu, J. Castellanos, and A. Doucet, "A bayesian exploration-exploitation approach for optimal online sensing and planning with a visually guided mobile robot," *Autonomous Robots*, vol. 27, no. 2, pp. 93–103, 2009.

[67] R. Oliveira, L. Ott, V. Guizilini, and F. Ramos, "Bayesian optimisation for safe navigation under localisation uncertainty," in *Robotics Research*. Springer, 2020, pp. 489–504.

[68] J. Gao, L. Zeng, C. Cao, W. Ye, and X. Zhang, “Multi-objective optimization for sensor placement against suddenly released contaminant in air duct system,” in *Building simulation*, vol. 11. Springer, 2018, pp. 139–153.

[69] S. Pourshahabi, M. R. Nikoo, E. Raei, and J. F. Adamowski, “An entropy-based approach to fuzzy multi-objective optimization of reservoir water quality monitoring networks considering uncertainties,” *Water Resources Management*, vol. 32, no. 13, pp. 4425–4443, 2018.

[70] D. Reina, H. Tawfik, and S. Toral, “Multi-subpopulation evolutionary algorithms for coverage deployment of uav-networks,” *Ad Hoc Networks*, vol. 68, pp. 16–32, 2018.

[71] E. C. Garrido-Merchán and D. Hernández-Lobato, “Predictive entropy search for multi-objective bayesian optimization with constraints,” *Neurocomputing*, vol. 361, pp. 50–68, 2019.

[72] H. M. Torun, M. Swaminathan, A. K. Davis, and M. L. F. Bellaredj, “A global bayesian optimization algorithm and its application to integrated system design,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 4, pp. 792–802, 2018.

[73] N. Karapetyan, J. Moulton, J. Lewis, A. Li, A. Quattrini, J. O’Kane, and I. Rekleitis, “Multi-robot dubins coverage with autonomous surface vehicles,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 2373–2379.

[74] A. Vasilijević, ĐNađ., F. Mandić, N. Mišković, and Z. Vukić, “Coordinated navigation of surface and underwater marine robotic vehicles for ocean sampling and environmental monitoring,” *IEEE/ASME transactions on mechatronics*, vol. 22, no. 3, pp. 1174–1184, 2017.

[75] A. Dutta, A. Ghosh, and O. P. Kreidl, “Multi-robot informative path planning with continuous connectivity constraints,” in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 3245–3251.

[76] E. Camci, D. Kripalani, L. Ma, E. Kayacan, and M. Khanesar, “An aerial robot for rice farm quality inspection with type-2 fuzzy neural networks tuned by particle swarm optimization-sliding mode control hybrid algorithm,” *Swarm and evolutionary computation*, vol. 41, pp. 1–8, 2018.

[77] T. Gao, H. Emadi, H. Saha, J. Zhang, A. Lofquist, A. Singh, B. Ganapathysubramanian, S. Sarkar, A. K. Singh, and S. Bhattacharya, “A novel multirobot system for plant phenotyping,” *Robotics*, vol. 7, no. 4, p. 61, 2018.

[78] D. P. Nicholson, A. P. M. Michel, S. D. Wankel, K. Manganini, R. A. Sugrue, Z. O. Sandwith, and S. A. Monk, “Rapid mapping of dissolved methane and carbon dioxide in coastal ecosystems using the chemyak autonomous surface vehicle,” *Environmental science & technology*, vol. 52, no. 22, pp. 13 314–13 324, 2018.

[79] F. Archetti and A. Candelieri, *Bayesian Optimization and Data Science*. Springer, 2019.

[80] T. Abbasi and S. Abbasi, "Conventional'indices for determining fitness of waters for different uses," *Water Quality Indices; Elsevier: Amsterdam, The Netherlands*, pp. 25–62, 2012.

[81] M. Buisson-Fenet, F. Solowjow, and S. Trimpe, "Actively learning gaussian process dynamics," in *Learning for dynamics and control*. PMLR, 2020, pp. 5–15.

[82] J. Görtler, R. Kehlbeck, and O. Deussen, "A visual exploration of gaussian processes," *Distill*, 2019, https://distill.pub/2019/visual-exploration-gaussian-processes.

[83] C. F. Falk and M. Muthukrishna, "Parsimony in model selection: Tools for assessing fit propensity." *Psychological Methods*, 2021.

[84] D. C. Liu and J. Nocedal, "On the limited memory bfgs method for large scale optimization," *Mathematical programming*, vol. 45, no. 1, pp. 503–528, 1989.

[85] U. Noè and D. Husmeier, "On a new improvement-based acquisition function for bayesian optimization," *arXiv preprint arXiv:1808.06918*, 2018.

[86] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[87] J. Blank and K. Deb, "Pymoo: Multi-objective optimization in python," *IEEE Access*, vol. 8, pp. 89 497–89 509, 2020.

[88] Emlid, "Emlid reach docs," https://docs.emlid.com/reach/, 2019, [Online; accessed 17-05-2022].

[89] R. P. Foundation, "Raspberyy pi 4 model b," https://www.raspberrypi.com/products/raspberry-pi-4-model-b/, 2019, [Online; accessed 17-05-2022].

[90] Nvidia, "Nvidia jetson agx xavier," nvidia.com/es-es/autonomous-machines/embedded-systems/jetson-agx-xavier/, 2019, [Online; accessed 17-05-2022].

[91] T. A. Morel, J. M. Manzano, G. Bejarano, L. Orihuela *et al.*, "Modelling and identification of an autonomous surface vehicle: Technical report," *Informes Técnicos*, 2022.

[92] Libelium, "Smart water sensor guide," https://development.libelium.com/smart_water_sensor_guide/sensors, 2019, [Online; accessed 17-05-2022].

[93] S. A, D. Ingram, J. Dinius, K. Chawla, A. Raffin, and A. Paques, "Pythonrobotics: a python code collection of robotics algorithms," *arXiv preprint arXiv:1808.10703*, 2018.

[94] F. Peralta, M. Arzamendia, D. Gregor, K. Cikel, M. Santacruz, D. G. Reina, and S. Toral, "Development of a simulator for the study of path planning of an autonomous surface vehicle in lake environments," in *2019 IEEE CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON).* IEEE, 2019, pp. 1–6.

[95] F. P. Samaniego, D. G. Reina, S. L. T. Marín, M. Arzamendia, and D. O. Gregor, "A bayesian optimization approach for water resources monitoring through an autonomous surface vehicle: The ypacarai lake case study," *IEEE Access*, vol. 9, pp. 9163–9179, 2021.

[96] F. Peralta, D. G. Reina, S. Toral, M. Arzamendia, and D. Gregor, "A bayesian optimization approach for multi-function estimation for environmental monitoring using an autonomous surface vehicle: Ypacarai lake case study," *Electronics*, vol. 10, no. 8, p. 963, 2021.

[97] F. Peralta, D. G. Reina, and S. Toral, "Water quality online modeling using multi-objective and multi-agent bayesian optimization with region partitioning," in *Engineering Applications of Artificial Intelligence.* Elsevier, 2022 under review.

[98] F. Peralta, S. Yanes, D. G. Reina, and S. Toral, "Monitoring water resources through a bayesian optimization-based approach using multiple surface vehicles: The ypacarai lake case study," in *2021 IEEE Congress on Evolutionary Computation (CEC).* IEEE, 2021, pp. 1511–1518.

[99] F. Peralta, D. G. Reina, and S. Toral, "Towards an online water quality monitoring system of dynamic environments using an autonomous surface vehicle," in *International Conference on Optimization and Learning OLA'2022*, 2022 in press.

[100] F. Peralta, M. Pearce, M. Poloczek, D. G. Reina, S. Toral, and J. Branke, "Multi-objective path planning for environmental monitoring using an autonomous surface vehicle," *Genetic and Evolutionary Computation Conference Companion (GECCO '22 Companion)*, 2022.