PATH PLANNING AND PERFORMANCE EVALUATION STRATEGIES FOR MARINE ROBOTIC SYSTEMS

by

Paul Stankiewicz

A dissertation submitted to Johns Hopkins University
in conformity with the requirements for the degree of
Doctor of Philosophy

Baltimore, Maryland
January, 2022

Abstract

The field of marine robotics offers many new capabilities for completing dangerous missions such as deep-sea exploration and underwater demining. The harshness of marine environments, however, means that without effective onboard decision-making, vehicle loss or mission failure are likely. Thus, to enable more autonomous operation while building trust that these systems will perform as expected, this thesis develops improved path planning and testing strategies for two different types of marine robotic platforms.

The first portion of the research focuses on improved environmental data collection with an autonomous underwater vehicle (AUV). Gaussian process-based modeling is combined with informative path planning to explore an environment, while preferentially collecting data in regions of interest that exhibit extreme sensory measurements. The performance of this adaptive data sampling framework with a torpedo-style AUV is studied in both simulation and field experiments. Results show that the proposed methodology is able to be fielded on an operational platform and collect measurements in regions of interest without sacrificing overall model fidelity of the full sampling area.

The second portion of the research then focuses on autonomous surface vessel (ASV) navigation that must comply with international collision avoidance standards and basic ship handling principles. The approach introduces a novel quantification of good seamanship that is used within an ASV path planner to minimize the collision risk with other vessels. This approach generalizes well to both single-vessel and multi-vessel encounters by avoiding rule-based conditions. The performance of this ASV planning strategy is evaluated in simulation against other baseline planners, and the results of on-water testing with a 29-ft ASV demonstrate that the approach is scalable to real systems.

Beyond developing improved path planning frameworks, this research also explores methods for improved testing and evaluation of black-box autonomous systems. Statistical learning techniques such as adaptive scenario generation and unsupervised clustering are used to extract the failure modes of the autonomy from large-scale simulation datasets. Subsequently, changes in these failure modes are tracked in a novel form of performance-based regression testing. The effectiveness of this testing framework is demonstrated on the aforementioned ASV planner by discovering several types of unexpected failures.

Thesis Committee

Readers

Marin Kobilarov (Primary Advisor)
Assistant Professor
Department of Mechanical Engineering
Johns Hopkins Whiting School of Engineering

Louis Whitcomb
Professor
Department of Mechanical Engineering
Johns Hopkins Whiting School of Engineering

Adam Watkins
Principal Professional Staff
Johns Hopkins Applied Physics Laboratory

Acknowledgments

As with any large undertaking, completing this thesis involved the help of many different people along the way. I'd like to give special thanks to Marin Kobilarov for advising me throughout the process. I have learned a lot working with you on a variety of research projects and in the classes that you've taught.

I also want to acknowledge many of my colleagues at the Johns Hopkins Applied Physics Laboratory (APL). Much of the work contained in this thesis is the direct result of support I received from APL that allowed me to pursue these interesting research topics. This organization has been invaluable in the assistance they provide from a financial perspective, in their flexibility to accommodate a full-time doctoral student, and also in the wealth of resources / engineering expertise that is available to consult.

Another special thanks goes to my thesis committee for reviewing this thesis and offering comments for improvement. I understand the time that is required to support quality research reviews and I appreciate your help.

Finally, I want to thank my wife Morgan for her support in dealing with a partner trying to balance a full-time job while completing this program.

Table of Contents

Abstract ii

Acknowledgments v

Table of Contents vi

List of Tables xiii

List of Figures xv

1 Introduction 1

1.1 Marine Robotic Systems 1

1.2 Organization & Contributions 3

1.2.1 Part I Overview 4

1.2.2 Part II Overview 6

1.3 Model-based Optimal Motion Planning 9

1.3.1 General Problem Setup 9

1.3.2 Finite-dimensional Optimization through Vehicle Motion Parameterization 10

I Adaptive Sampling With an Autonomous Underwater Vehicle in Static Marine Environments 12

2 Background on Environmental Monitoring with Marine Robotics 13

2.1 Motivation 13

2.2 Limitations 16

2.3 Related Work 17

3 Methodology for Information-Seeking Path Planning 22

3.1 Problem Setup 22

3.1.1 Vehicle Dynamic Model 22

3.1.2 AUV Motion Parameterization 26

3.1.3 Environmental Modeling and Prediction as a Gaussian Process 27

3.1.4 Hyperparameter Adaptation 30

3.1.5 Acquisition Function 30

3.2 Adaptive Trajectory Planning 33

3.2.1 Branch & Bound Informative Path Planning (BB-IPP) . 35

3.2.1.1 Branching Procedure 37

3.2.1.2 Branching Reachability Analysis 39

3.2.1.3 Bounding Procedure 42

3.2.2 Cross-Entropy Informative Path Planning (CE-IPP) . . 43

4 Performance Evaluation of Gaussian Process Adaptive Sampling 46

4.1 Simulation Studies 46

4.1.1 Algorithm Parameter Tuning 49

4.1.1.1 BB-IPP Parameter Tuning 49

4.1.1.2 CE-IPP Parameter Tuning 51

4.1.2 Performance Over the Sampling Mission 53

4.1.2.1 Quantitative Performance 55

4.1.2.2 Example Trajectories 58

4.1.3 Performance on Randomized Environmental Fields . . 61

4.2 Field Experiments 65

4.2.1 Test Platform 65

4.2.2 Implementation Details 68

4.2.3 Experiment #1: Virtual Environmental Field 69

4.2.4 Experiment #2: In Situ Measurements From Dissolved Oxygen Field 71

4.3 Summary of Findings 74

II Good Seamanship Path Planning and Performance Evaluation for Autonomous Surface Vessels 75

5 Background on Autonomous Surface Vessel Navigation 76

5.1 Review of COLREGS 77

5.2 Collision Risk Assessment 80

5.3 ASV Navigation Related Work 82

6 Encoding Good Seamanship Principles Within a Path Planning Framework 85

6.1 Surface Vessel Dynamic Model 86

6.2 Surface Vessel Motion Parameterization 89

6.3 Quantifying Good Seamanship 90

6.3.1 Collision Risk Quantification 91

6.3.1.1 Ship Domain 91

6.3.1.2 Collision Index 96

6.3.2 Appropriate Action Quantification 101

6.3.2.1 Ship Arena 101

6.3.2.2 Action Index 103

6.3.3 Overall Risk Index 103

6.4 Path Planning Approach 106

6.4.1 Situational Awareness 107

6.4.2 Cost Function 108

6.4.2.1 Mission Cost 109

6.4.2.2 Safety Cost 109

6.4.2.3 COLREGS Cost 110

6.4.3 Branch & Bound Path Planning 112

6.4.3.1 Branching Procedure 113

6.4.3.2 Bounding Procedure 115

7 Towards Verifying Autonomy Decision-Making through Simulation-Based Testing 117

7.1 Introduction 118

7.1.1 Autonomy Testing & Evaluation 118

7.1.2 Performance Regression Testing 120

7.2 Performance Analysis Methodology 123

7.2.1 Problem Setup 124

7.2.2 Adaptive Scenario Generation 125

7.2.3 Failure Mode Identification 129

7.2.3.1 Performance Space Clustering 129

7.2.3.2 Testing Space Clustering 130

7.2.3.3 Feature Scaling 131

7.3 Identifying Performance Regression in Autonomous Systems 132

7.3.1 Problem Setup 133

7.3.2 Regression Modeling 134

7.3.3 Regression Clustering 136

7.4 Preliminary Analysis on Test Functions 136

8 Performance Evaluation for Autonomous Surface Vessel Navigation 141

8.1 Analysis Setup 142

8.1.1 ASV Testing Space Design 142

8.1.1.1 Single-Vessel Standard Testing Space 145

8.1.1.2 Single-Vessel Extended Testing Space 145

8.1.1.3 Multi-Vessel Testing Space 146

8.1.1.4 Handcrafted Test Scenarios 146

8.1.2 ASV Performance Space Design 146

8.1.2.1 Mission Score 150

8.1.2.2 Safety Score 150

8.1.2.3 COLREGS Score 152

8.1.2.3.1 Single-Vessel COLREGS Scoring: . . 152

8.1.2.3.2 Multi-Vessel COLREGS Scoring: . . 153

8.2 Performance Evaluation Results 154

8.2.1 Comparison to Baseline Planning Strategies 154

8.2.2 Failure Mode Introspection 157

8.2.3 Performance Regression Analysis Case Study 161

8.3 Field Experiments on Candidate Scenarios 165

8.3.1 Single-Vessel Scenarios 166

8.3.1.1 Crossing Stand-on – Fig. 8.11 166

8.3.1.2 Offset Head-on – Fig. 8.12 168

8.3.1.3 Overtaking – Fig. 8.13 169

8.3.2 Multi-Vessel Scenarios 170

8.3.2.1 Conflicting Crossing Expectations – Fig. 8.14 170

8.3.2.2 Double Crossing Give-way – Fig. 8.15 171

8.3.2.3 Conflicting Overtaking Expectation – Fig. 8.16 172

9 Conclusion **174**

9.1 Part I Summary 174

9.2 Part II Summary 175

9.3 Future Work 176

Bibliography **180**

List of Tables

3.1 Hydrodynamic model coefficients for the Johns Hopkins University Iver3 AUV. 26

4.1 Summary of the performance metrics used to evaluate the planning approaches. 48

4.2 Algorithm parameters used for simulation studies. 50

4.3 Mean and standard deviation of the evaluation metrics (Table 4.1) for parameter variations of the BB-IPP algorithm. Fifty simulations were performed for each parameter combination. The bolded values are those that were chosen from the parameter tuning for further simulation and field experiments. 52

4.4 Mean and standard deviation of the evaluation metrics (Table 4.1) for parameter variations of the CE-IPP algorithm. Fifty simulations were performed for each parameter combination. The bolded values are those that were chosen from the parameter tuning for further simulation experiments. 54

4.5 JHU Iver3 AUV measurement sources and resolutions. 66

4.6 Summary of performance metrics observed for each field experiment. Values for the GP RMSE and GP Variance are measured at the end of the sampling mission. GP RMSE was not calculated for the experiment in TA2 due to the lack of a ground truth dissolved oxygen source. 71

6.1 Surface vessel hydrodynamic model coefficients 88

8.1 ASV planner parameters used for simulation studies and field experiments. 143

8.2 Test parameter ranges and descriptions for the ASV testing spaces. The intercept time in the following descriptions is defined as the time at which the target vessel and ASV are collinear along the North-South axis assuming constant speed and heading. 145

8.3 Performance metrics for each ASV planning method. 158

List of Figures

1.1 Two examples of existing marine robotic platforms: the Sentry AUV (left), operated by Woods Hole Oceanographic Institution, and the C-Worker 7 ASV (right), operated by L3Harris. 2

2.1 Hypoxic dead zone formation in the Chesapeake Bay. 14

2.2 Concept of environmental sampling using a heterogeneous robot team. High-mobility vehicles such as UAVs and ASVs collect sparse samples over a large area, whereas AUVs provide focused samples in a region of interest (this paper focuses on the latter). 21

3.1 Johns Hopkins University L3 OceanServer Iver3 AUV. 23

3.2 Visualizations of three trajectories generated from different values of z over a period of 20 sec. Units for v_x, ω_z, and v_z are m/s, rad/s, and m/s, respectively. 28

3.3 Illustrative example of the receding horizon BB-IPP algorithm driving towards a high-reward (yellow) ROI along the right edge of the sampling area. This 2D example uses $H = 2$ and trajectory parameterizations of $\boldsymbol{z} = (v_x, \omega_z)$ corresponding to speed and turn rate, respectively. For $v_x > 0$, the gray region represents the reachable space for one branching step over $t + \delta_t$. 37

3.4 (a) The approximate reachable workspace $\mathcal{W}_{\boldsymbol{x}_0,T}$ over different planning horizons for $v_x \in [0.8, 1.6]$ m/s and $\omega_z \in [-0.09, 0.09]$ rad/s. The nonzero minimum speed means that the area immediately in front of the vehicle is not reachable for the shorter horizons. (b) Candidate sampling locations $\widehat{\mathcal{W}}_{\boldsymbol{x}_0,T}$ (gray dots) evaluated for M trajectories over $T = 140$ sec. The largest empty circle for the samples is shown on the right. 40

3.5 Monte Carlo analysis of the maximum empty radius (a) and maximum percentage of empty reachable space (b) as a function of the number of sampled trajectories for various planning horizons. Each data point shows the mean calculated over 30 trials with the error bars representing two standard deviations. The parameter combination implemented for the BB-IPP algorithm is highlighted. 41

3.6 Illustrative example of the receding horizon CE-IPP algorithm driving towards a high-reward (yellow) ROI along the right edge of the sampling area. This 2D example uses $H = 2$ and trajectory parameterizations of $\boldsymbol{z} = (v_x, \omega_z)$ corresponding to speed and turn rate, respectively. For $v_x > 0$, the gray region represents the reachable space for one branching step over $t + \delta_t$. 44

4.1 True environmental distribution used for simulation studies with a static field normalized in the range $0 - 1$. ROIs are shown in red as the volume below the desired measurement threshold of 0.22. 47

4.2 Metrics used to evaluate the performance of each path planning method over a set of 150 simulations (LM is the baseline undulating lawnmower path). Plots begin at the 10th sampling step so as to let the GP train with an initial set of measurements. The lines are the mean values and the shaded regions capture values within one standard deviation around the mean. 56

4.3 Hyperparameter estimates $\boldsymbol{\theta}^* = [l_x, l_y, l_z, \sigma_f, \sigma_w]^T$ for each path planning method over a set of 150 simulations (LM is the baseline undulating lawnmower path). The lines are the mean values and the shaded regions capture values within one standard deviation around the mean. 57

4.4 Simulation results for three different planning surveys. The top images shows the sampling points in 3D while the middle images show a bird's-eye view of the vehicle trajectory with the estimated distribution at the seafloor. The time series plot on the bottom shows the normalized measurement value over the simulation where measurements within an ROI are highlighted in red. 60

4.5 An example of a randomly generated environmental distribution and the resulting sampling points. Bathymetry is shown in brown where the right figure illustrates a portion of the trajectory exhibiting bottom-following at a safe minimum height. 61

4.6 Performance metric box plots for each path planning method over a simulation set of 500 randomly generated sampling environments. The GP RMSE and variance results were based on a GP model trained at the end of each mission. 64

4.7 AUV field experiments were conducted in the Severn River, Maryland. The northernmost test area, referred to as TA1, ran the GPAS algorithm using virtual measurements from a simulated environmental field with multiple ROIs. The southernmost test area, referred to as TA2, ran the GPAS algorithm using in situ dissolved oxygen measurements. 67

4.8 Results for the field experiment performed in TA1 using the virtual environmental distribution of Section 4.1. The left image shows the estimated GP model with sampling points in 3D. The measured bathymetry of the test area is shown in brown. The top right plot shows a bird's-eye view of the AUV trajectory with the estimated distribution at the seafloor. The time series plots in the bottom right show the normalized virtual measurements, where measurements within an ROI are highlighted in red, and the vehicle altitude. 70

4.9 Results for the field experiment performed in TA2 using in situ dissolved oxygen measurements. The left image shows the estimated GP model with sampling points in 3D. The measured bathymetry of the test area is shown in brown. The top right plot shows a bird's-eye view of the AUV trajectory. The time series plots in the bottom right show the DO measurements, where measurements below the normalized ROI range (22nd percentile) are highlighted in red, and the vehicle altitude. . . 72

5.1 Diagrams of COLREGS rules 13 – 15 and the expected actions by each vessel. 80

5.2 Example of a multi-vessel scenario with conflicting COLREGS rules. 84

6.1 Autonomous surface vessel used in field experiments for which the hydrodynamic model was derived. 86

6.2 Values for the yaw acceleration model coefficients based on vessel surge speed. 88

6.3 Two configurations with the same d_{cpa} value. It is clear, however, that the scenario on the left presents a much higher risk at CPA than the scenario on the right. 91

6.4 Decentralized ellipse ship domain, where d is the distance to the edge of ownship's domain in the direction of the target ship. The fuzzy ship domain, which translates domain violation $f_d(t)$ into a risk value, is shown by the inner contour plots. 92

6.5 Diagram depicting variables used in the derivation of $f_d(t)$. . 97

6.6 Illustration of overall risk methodology. Θ^i_C is calculated as the maximum mutual domain risk with the i-th target ship over a future time horizon $t + T$. Θ^i_A is calculated based on the i-th target ship's degree of penetration within ownship's arena. . . 102

6.7 Overall risk curve for a two-vessel crossing scenario. Snapshots of the vessel configuration are given at the time of maximum overall risk (left) and the time of CPA (right). 105

6.8 Primitive-based receding horizon planning that encodes good seamanship principles for multi-vessel scenarios. 107

6.9 Example of the procedure used to estimate the J_{LB} heuristic, where $H = 4$ was set arbitrarily for illustration purposes. . . . 116

7.1 Simulation performance datasets for two versions of autonomy software. The red box highlights performance improvement whereas the gray box highlights performance regression. . . . 121

7.2 A design for augmented continuous integration for testing autonomy software. In addition to traditional unit testing that checks for software quality issues, automated simulation-based testing is used to evaluate the holistic performance of the system. Additionally, statistical learning is used to detect performance regression, i.e., areas where the SUT may have degraded in performance when compared to its previous versions. . . . 122

7.3 High-level flow diagram of the performance analysis methodology to identify failure modes, influenced by the work in [93]. 124

7.4 Overall performance regression analysis approach. Surrogate modeling is used to normalize each study to the other and predict whether samples have regressed. Unsupervised clustering is then used to extract unique regression clusters in both the severity of the performance decrease and its location within the testing space. 134

7.5 Example of the regression analysis applied to two randomly generated test functions (left). The resulting regression dataset and three high-priority clusters are shown by the middle and right plots, respectively. 137

7.6 Regression modeling performance on the prediction of $\Delta\hat{Y}_*$ for randomized test functions. 139

8.1 Illustration of the test parameters used for each testing space. Ranges and descriptions are given in Table 8.2. 144

8.2 Depiction of the handcrafted COLREGS scenarios test suite. Each black arrow designates a separate test scenario, with the length of the arrow representing the relative velocity of the target ship with respect to the ASV. 147

8.3 The $\mathcal{Y}_S$ performance surface resulting from Eq. (8.6). 152

8.4 Performance landscapes of each planning strategy calculated from 3000 adaptively-generated scenarios on the $\mathcal{X}^3_{1v}$ testing space. The color of each scenario represents the performance scores $\mathcal{Y}$, while the black dots represent the set of handcrafted scenarios (introduced in Fig. 8.2) as a baseline for comparison. 155

8.5 Histograms for each planner representing the percent volume of the testing space occupied with respect to different scoring bins of $\mathcal{Y}$. 156

8.6 (a) The "top 3" failure clusters from the Q_* set (Q_1 = blue, Q_2 = red, Q_3 = gold). The ellipses display each cluster's covariance structure and the the black circle highlights the scenario shown in (b). 159

8.7 Example scenario from the primary failure mode of the $\mathcal{X}^7_{2v}$ testing space, characterized by a violation of ownship's collision radius by TS2. 160

8.8 Case study results of applying the performance regression analysis to the $\mathcal{X}^3_{1v}$ testing space of the good seamanship planner. . 162

8.9 Example scenarios of the primary R_1 regression cluster. After the software updates, scenarios within the R_1 cluster show that the ASV is susceptible to being herded when traveling on a similar velocity vector to the target ship. 164

8.10 Field tests with a 29-ft autonomous RHIB were performed in the Chesapeake Bay to evaluate the good seamanship planner on an operational platform. 166

8.11 Crossing scenario where ownship is expected to stand on. . . 167

8.12 Offset head-on scenario where both vessels are expected to give way. 169

8.13 Overtaking scenario where ownship is expected to give way. . 170

8.14 Scenario with two vessels crossing from opposite directions and conflicting COLREGS expectations. 171

8.15 Scenario with two vessels crossing where ownship is expected to give way to both vessels. 172

8.16 Scenario where ownship overtakes one vessel while being overtaken by a second vessel. 173

Chapter 1

Introduction

This chapter gives the reader a brief introduction into some of the main themes of this thesis, primarily marine robotics path planning and the associated challenges of ensuring the system's performance. The organization of the thesis and its contributions are then summarized.

1.1 Marine Robotic Systems

The past several decades have seen robotic system development rapidly accelerate beyond use in traditional factory settings to now solve challenging real-world problems. Marine environments are one domain where these advancements have been invaluable, opening up the possibility for applications that were previously not achievable. Autonomous underwater vehicles (AUVs), a class of mobile robot designed to operate sub-surface, offer the promise and capability to explore underwater worlds for scientific data collection that humans cannot safely access. The Sentry AUV shown in Fig. 1.1, designed by the Woods Hole Oceanographic Institution, is one example that

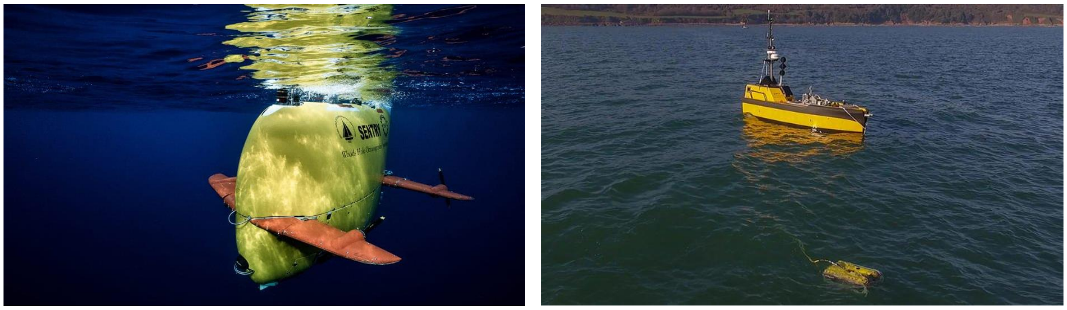


Image credit: Woods Hole Oceanographic Institution

Image credit: L3Harris

Figure 1.1: Two examples of existing marine robotic platforms: the Sentry AUV (left), operated by Woods Hole Oceanographic Institution, and the C-Worker 7 ASV (right), operated by L3Harris.

can operate at depths of 6,000 meters and has been used to autonomously survey the seafloor to create bathymetric maps, locate hydrothermal vents, and explore shipwrecks.

Another class of system within marine robotics is autonomous surface vessels (ASVs), which can essentially be thought of as self-driving boats. ASVs offer a complementary line of capabilities to that of AUVs, including environmental monitoring tasks, opportunities in commercial shipping, and various defense applications such as harbor patrol and mine countermeasures. Figure 1.1 also shows an example of an existing ASV platform: the C-Worker 7, designed by L3Harris. This platform is capable of operating offshore and deploying its own underwater vehicle for infrastructure inspection.

While the promise of marine robotics is enormous, there are many challenges that remain in the development of these systems due to the harsh, unpredictable environment in which they operate. From a mechanical design perspective, marine robotics must contend with corrosive seawater, the extreme pressures seen in the deep sea, and the constant wear from wind and waves. Limited communication bandwidth in the the open ocean (and even more so in underwater applications) also means that these systems must make decisions with very little operator interaction. AUVs must be able to localize both themselves and their mission objectives with little sensor information, while ASVs have their own navigation challenges in needing to safely avoid obstacles and other vessels on the water. These requirements demand highly robust systems, especially given their high cost and safety-critical nature.

Additionally, unlike traditional marine systems, AUVs and ASVs (as well as other autonomous systems more generally) must not only achieve robustness in their hardware components, but also in their decision-making software components. Properly verifying that these platforms will behave as expected is crucial to gaining the trust needed for their full adoption. In essence, it becomes necessary to test the "brain" of the system in addition to its structure, even when many competing requirements may result in unpredictable emergent behavior from the decision-making logic. Thus, it becomes crucial to pursue improved testing and evaluation (T&E) technologies for marine robotics in lockstep with development of the system's algorithmic capabilities.

1.2 Organization & Contributions

The challenges described above highlight some of the main themes tackled by this thesis: improving high-level planning strategies within marine robotics, and also improving the testing methods used to evaluate their performance. The remaining text is broken up into two distinct parts based on applications to different marine robotic platforms.

1.2.1 Part I Overview

The first part of the thesis focuses on the problem of AUV path planning that optimizes data collection for improved environmental monitoring. Specifically, the AUV is tasked with using in situ measurements of the surrounding environment to locate and collect data from environmental regions of interest, e.g., hypoxic zones. Much of the text in Part I was originally published by the author [122] in collaboration with Yew Teck Tan and Marin Kobilarov, and includes the following novel contributions.

- **Receding-horizon informative planning.** An adaptive sampling framework is proposed in conjunction with two informative path planning strategies for generating high-information trajectories (from a data quality perspective). The first informative path planner is based on branch-and-bound techniques and the second is based on cross-entropy optimization. Both of these planning techniques explicitly consider the motion constraints of an underactuated AUV operating within a constrained environment. This is in contrast to much of the environmental sampling research in the marine domain [114, 72, 22], which focuses on large-scale, low-resolution environmental monitoring where motion constraints need not be prioritized.

- **Constrained 3D environments.** Applications of AUV adaptive sampling are explored where the complete 3D environmental area is considered for collecting data samples. To date, most studies have focused on environmental sampling in 2D environments. While a handful of studies collect data in 3D, the depth component is typically only controlled via predefined yo-yo patterns [23, 114]. These approaches simply generate and utilize 2D models of the environment for planning as opposed to considering the added depth component. The methodology presented in Part I allows for dense data collection near the seafloor, which would be difficult for underwater gliders or surface vessels. Further, this work also considers environmental constraints, such as bathymetry and bottom structure, and operational constraints, such as limited communication and navigation capabilities, that require the AUV to periodically surface.

- **Simulations and real-world deployments.** Monte Carlo simulation experiments are used to measure the performance of the proposed methods over a large set of auto-generated 3D environments. Additionally, several field experiments with an AUV show the capability of the proposed algorithms to be implemented on operational platforms for real-time identification of regions of interest in a 3D sampling environment.

Chapter 2 motivates the need for improved marine sampling mechanisms and reviews related work in the area of adaptive data collection. Chapter 3 introduces the problem formulation and proposes the methodology for adaptive trajectory planning that maximizes information collection. Results from both simulation studies and field experiments with an AUV are then used to demonstrate the efficacy of the approach in Chapter 4.

1.2.2 Part II Overview

The second part of the thesis then focuses on a different problem within marine robotics: improved path planning strategies for ASVs. Navigation on the open ocean is challenging because not only must ASVs perform standard obstacle avoidance, they must also adhere to protocols that are expected of all ships (i.e., the "rules of the road" for marine surface craft). Additionally, a performance-based continuous integration framework is proposed for improved autonomy T&E that is applied to the ASV navigation problem. Much of the content in Part II was originally published by the author [121, 118, 120, 119] in collaboration with Galen Mullins, Michael Heistand, and Marin Kobilarov, and contains the following novel contributions.

- **Quantification of good seamanship principles.** Good seamanship[1] is an important requirement of maritime navigation; however, the exact principles that embody good seamanship are notoriously vague. This research proposes a novel quantification of good seamanship principles in complicated multi-vessel scenarios. The basis of this quantification comes from using ship domain and ship arena concepts to derive future collision risk indices based on maximum mutual ship domain violation. The combination of both ship domain and ship arena gives a more complete picture of the overall risk rather than only considering penetration within a single safety zone [28].

[1] The term "seamanship" is used throughout this thesis because it is explicitly referenced in international maritime protocols [41] and contains intrinsic concepts that are understood by the practicing community. In an effort to avoid gender-biased language, however, terms such as seamanship should begin to be phased out and replaced with neutral terms (e.g., ship handling) in the future.

- **Receding horizon good seamanship planning.** This research then offers a multi-layer planning approach for ASVs that incorporates the aforementioned good seamanship quantification into metrics leveraged for path planning. The approach combines novel situational awareness logic with motion primitive-based planners in a receding horizon framework. By relying on metrics-driven motion planning as opposed to rule-based conditions, the proposed framework scales naturally to non-trivial single-vessel and multi-vessel situations when compared to more standard approaches based on closest point of approach [113, 123, 137, 57].

- **Holistic performance-based autonomy evaluation.** A series of improvements to autonomy T&E methods are also proposed. While the performance evaluation is applied to the ASV navigation problem, the general formulation is applicable to testing autonomous systems in any domain.

  1. *Adaptive scenario generation.* A methodology for adaptive simulation-based testing is proposed to statistically measure system performance. The framework uses adaptive scenario generation to hone in on the failure modes of the system, thus giving greater resolution of these failures when compared to one-off analysis on handcrafted scenarios or random Monte Carlo methods. Additionally, these failure modes are automatically grouped into explainable clusters from large, high-dimensional datasets using unsupervised learning techniques.

  2. *Identifying performance regression in autonomy.* Further, this research

presents one of the first frameworks specifically designed to identify whether/how a black-box autonomous system has regressed in performance when compared to previous versions. The approach analyzes multiple performance datasets (typically gathered through the type of simulation-based testing describe above) and again applies statistical learning techniques to determine regions of predicted performance regression between the datasets. This capability is proposed in the context of an augmented continuous integration pipeline that automatically checks for new failure modes in the decision-making components of the system.

3. *On-water field testing.* A variety of field experiments are described that test the good seamanship planner with a 29-ft ASV on the Chesapeake Bay. These experiments test the planner in single-vessel and multi-vessel collision encounters, where each scenario prescribes a different collision geometry in order to demonstrate how the planner generalizes to realistic conditions.

The challenge of developing ASVs to operate within existing maritime navigation conventions is first posed in Chapter 5. The good seamanship quantification and its adaptation for use in path planning are then described in Chapter 6. Chapter 7 details a simulation-based evaluation framework for autonomous systems geared around failure-seeking scenario generation and performance regression testing that compares previous versions of the system against the current iteration. This evaluation framework is then applied to the proposed ASV planner in Chapter 8 to analyze its performance against baseline approaches and perform an introspection of its failure modes. Field testing is also presented that validates these results with on-water data.

1.3 Model-based Optimal Motion Planning

Before jumping to each of the two applications described above, there are some commonalities in the motion planning problems that can be generally stated. There is a distinction within robotics between global path planning, which considers the high-level action sequence of the system over mission-length timescales, and local path planning, which considers the immediate actions that are both feasible and follow desired states at a much shorter timescale. Both of the applications in this thesis fall more into the former category, focusing on high-level path generation that satisfies mission criteria on a timescale longer than the system's dynamics. We can formalize this path planning problem for use in both parts of the thesis.

1.3.1 General Problem Setup

Consider a vehicle trajectory evolving according to an ordinary differential equation, $\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{u}(t), t)$, where the state and control at time $t > 0$ are denoted by $\boldsymbol{x}(t) \in \mathbb{X}$ and $\boldsymbol{u}(t) \in \mathbb{U}$, respectively, with $\mathbb{X} = \mathbb{R}^n$ and $\mathbb{U} \subset \mathbb{R}^m$ denoting the state space of dimensionality n and control space of dimensionality m. Letting t_f be the final time of the trajectory (i.e., the allotted mission time), a given control curve $\boldsymbol{u} : [0, t_f] \to \mathbb{U}$ determines a unique state trajectory $\boldsymbol{x} : [0, t_f] \to \mathbb{X}$ by evolving the dynamics from an initial state $\boldsymbol{x}_0 \in \mathbb{X}$. We would like to compute the optimal controls $\boldsymbol{u}^*(\cdot)$ that maximize a given acquisition function $L : \mathbb{U} \times \mathbb{X} \to \mathbb{R}$ while driving the system from its initial state $\boldsymbol{x}_0$ until the final time t_f, i.e.,

$$\begin{aligned}
\underset{\boldsymbol{u}}{\text{maximize}} \quad & \phi(\boldsymbol{x}(t_f), t_f) + \int_0^{t_f} L(\boldsymbol{x}(t), \boldsymbol{u}(t), t)\, dt \\
\text{subject to} \quad & \dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{u}(t), t), \\
& \boldsymbol{x}(0) = \boldsymbol{x}_0, \\
& g(\boldsymbol{x}(t), \boldsymbol{u}(t), t) \leq 0,
\end{aligned} \tag{1.1}$$

for all $t \in [0, t_f]$. Here, $g(\boldsymbol{x}(t), \boldsymbol{u}(t), t) \leq 0$ are constraints on the state and input arising from actuator bounds and obstacles in the environment, and $\phi(\boldsymbol{x}(t_f), t_f)$ is a reward placed on the terminal state of the trajectory. Next, we can convert the problem of Eq. (1.1) into a numerically convenient form that serves as the basis for the proposed planning algorithms presented in Part I and Part II.

1.3.2 Finite-dimensional Optimization through Vehicle Motion Parameterization

The infinite-dimensional trajectory optimization problem of Eq. (1.1) can be converted into a finite-dimensional optimization by parameterizing vehicle trajectories as sequences of motion primitives. Let π represent a particular trajectory, with the space of all trajectories originating at point $\boldsymbol{x}_0$ and satisfying the dynamics of the ODE denoted by

$$\mathcal{P} = \{\pi : t \in [0, t_f] \to \{\boldsymbol{u}(t), \boldsymbol{x}(t)\} \mid \dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{u}(t), t),\ \boldsymbol{x}(0) = \boldsymbol{x}_0,\ t_f > 0\}. \tag{1.2}$$

Trajectories are parameterized with a function $\varphi : \mathcal{Z} \rightarrow \mathcal{P}$ according to:

$$\pi = \varphi(\boldsymbol{z}) \equiv \varphi_{\boldsymbol{z}}, \tag{1.3}$$

where $\boldsymbol{z} \in \mathcal{Z} \subset \mathbb{R}^z$ is the parameter space. Let $\Gamma : \mathbb{U} \times \mathbb{X} \rightarrow \mathbb{X}$ project onto the state component, i.e., $\Gamma(\boldsymbol{u}, \boldsymbol{x}) = \boldsymbol{x}$. The constrained parameter space $\mathcal{Z}_{con} \subset \mathcal{Z}$ is the set of parameters satisfying the boundary conditions and constraints, defined by

$$\mathcal{Z}_{con} = \{\boldsymbol{z} \in \mathcal{Z} \mid g(\Gamma(\varphi_{\boldsymbol{z}}(t))) \leq 0,\ t \in [0, t_f]\}. \tag{1.4}$$

Define the reward γ and the reward function $J : \mathcal{Z} \rightarrow \mathbb{R}$ according to:

$$\gamma = J(\boldsymbol{z}) = \int_0^{t_f} L(\varphi_{\boldsymbol{z}}(t))\, dt. \tag{1.5}$$

Equation (1.1) can now be solved approximately by finding $\{\boldsymbol{x}^*, \boldsymbol{u}^*\} = \varphi(\boldsymbol{z}^*)$ such that

$$\boldsymbol{z}^* = \underset{\boldsymbol{z} \in \mathcal{Z}_{con}}{\operatorname{argmax}}\, J(\boldsymbol{z}). \tag{1.6}$$

Similarly, we let γ^* be the reward function optimum over the space $\mathcal{Z}_{con}$:

$$\gamma^* = \max_{\boldsymbol{z} \in \mathcal{Z}_{con}} J(\boldsymbol{z}). \tag{1.7}$$

Planning in the parameter space $\mathcal{Z}$ offers a means to efficiently generate trajectories that consider the constraints of the system through closed-form motion primitives, as opposed to requiring expensive integration of the full system dynamics. The instantiations of $\mathbb{X}$, $\mathbb{U}$, $\mathcal{Z}$, φ, and L for the AUV and ASV planning problems of this thesis are each defined in their respective parts.

Part I

Adaptive Sampling With an Autonomous Underwater Vehicle in Static Marine Environments

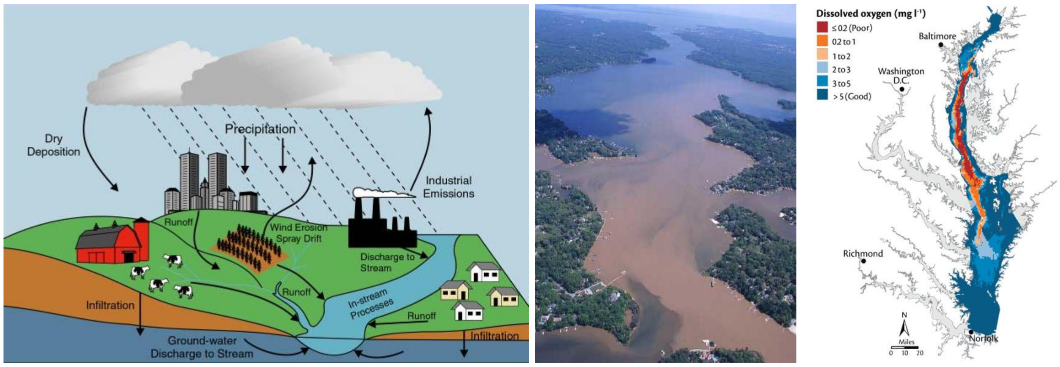
Chapter 2

Background on Environmental Monitoring with Marine Robotics

2.1 Motivation

One of the most practical and influential applications of the recent advances in autonomous systems is in marine environmental monitoring. For a given measurement of interest, e.g., salinity, dissolved oxygen, etc., a natural goal is to use collected data to create an environmental model for further study. These models can be used to localize areas that contain either high measurement variability or extreme values to help authorities identify potential abnormalities and address the problem.

As a specific use case, consider the study of hypoxia in a large estuary such as the Chesapeake Bay (Fig. 2.1), that in last few decades has become increasingly sensitive to pollution from agricultural and wastewater discharge, urban runoffs, and atmospheric deposition [105]. To establish effective treatment and prevention practices, it is necessary to monitor and assess the complex



Nutrient movement
(credit: USGS)

Runoff affecting water quality
(credit: Severn Riverkeeper)

Dead zones
(credit: Chesapeake Bay Foundation)

Figure 2.1: Hypoxic dead zone formation in the Chesapeake Bay.

dynamic processes affecting water quality. The standard approach is to supplement networks of fixed sampling stations with manual sampling from ships, and perform data integration and analysis in the lab. This approach has several limitations that could hinder water management: short timescale events such as rapid rainstorm runoffs cannot be captured at sufficient resolution; diffusive pollution sources are difficult to track from sparse samples; critical locations such as shallow and obstacle-cluttered areas as well as creeks are difficult to access by ship and require additional infrastructure, integration, and manpower. Additionally, simply interpolating between stations has the potential to miss critical regions of interest (ROIs), i.e., regions that exhibit extreme sensory measurements and high spatially-correlated variability when compared to the surrounding area.

More recently, autonomous underwater vehicles (AUVs) and autonomous surface vessels (ASVs) have begun to be used in these data collection operations; however, they are typically constrained to area coverage objectives and thus follow predefined sampling trajectories such as a lawnmower pattern

[36]. These strategies are highly inefficient when the overall goal is to detect and localize ROIs within a larger area. Marine environments exacerbate this problem (particularly for AUVs) because it is difficult for an operator to update the vehicle trajectory due to communication limitations. The result is that the data collected by the system is still only available to provide insights in a post-processed fashion.

Rather, autonomous systems now have the capability to adapt their trajectories based on in situ data processing to focus sampling efforts on higher-level goals such as localizing ROIs. Recent researchers have made use of the spatial correlation structure of these ROIs to design adaptive sampling strategies for robotic environmental monitoring [70, 97]. In these informative path planning strategies, the sensor information observed along the vehicle trajectory is assimilated into an environmental model, which is then used to guide future sampling locations. The ability to adapt the sampling paths using in situ sensor information has shown to be a more effective method for sensing and localizing ROIs in a given environment.

This work aims to overcome some of the challenges in marine monitoring through adaptive sampling with robotic vehicles guided by real-time information-seeking algorithms, which autonomously decide sampling locations to optimize a given task such as pollution source localization. Specifically, we focus on single-vehicle planning in a 3D environment for collecting a dense set of measurements within highly localized ROIs, e.g., channels with varying bathymetry in a large estuary. This research extends upon previous work [128], which was limited to 2D environments and applied to surface vehicles.

2.2 Limitations

While the application and use of AUV sampling platforms allows increased data resolution in constrained environments, these systems typically have limited on-board computational resources. This constraint results in some limitations of the proposed approach that must be acknowledged. In order to facilitate real-time planning and model-building for a 3D environment, trajectory generation is performed in a receding horizon fashion. This approach, while precluding a globally optimal solution over the full mission profile, is able to generate high-information trajectories in a local sense up to the planning horizon of the sampling platform. The optimality of these solutions is then also dependent on approximations of both the platform's dynamics and reachable space over the local planning horizon.

An additional assumption within this research is that the environmental field is static with respect to time. In other words, we assume that the temporal dynamics of the environment are much slower than the duration of the sampling mission (typical AUV deployments within the application of interest only last on the order of hours). Generalization of the proposed framework for long-term monitoring that considers a dynamic environment is left to future work, perhaps leveraging concepts from the non-dimensional approach of Leonard et al. [62]. In spite of these limitations, the proposed approach provides significantly improved data collection when compared to traditional methods.

2.3 Related Work

The use of robotics for environmental monitoring has become a significant research effort, with a survey of applications provided by Dunbabin and Marques [26]. For example, applications have been studied in diverse domains including volcanic environments [96], data collection in severe weather [34], forest fire mapping [97], and acoustic predictions in ocean fields [133]. A typical approach to this application is to use sensor information to build a model of the environment. Predictions from this model and their associated uncertainties are then taken into account when planning sampling paths. When performing sensing in an unknown environment, balancing a trade-off between exploration and exploitation [106] is crucial. Exploration refers to collecting samples that fully survey the environmental area, whereas exploitation refers to collecting samples in areas that contain the desired measurements, i.e., in ROIs. Thus, care must be taken to design algorithms that balance these two competing objectives when selecting sampling paths.

A commonly used model for representing environmental phenomena is the Gaussian process (GP) [101, 23, 39]. A Gaussian process is a non-parametric regression model that allows estimation and prediction of the spatial correlation structure of an environmental field. Sensor data, as it is acquired, is used to learn the GP by estimating the model's hyperparameters. In turn, the mean and variance predictions of the model are used to guide future sampling locations. This iterative, sequential model update and prediction is well-suited for adaptive sampling missions where sensing platforms must repeatedly select the the best sampling path. Such a problem can be formulated as Bayesian optimization by using the GP to maintain a probabilistic belief about the environmental distribution while using its predictions to design an appropriate acquisition function [51, 135, 13]. For example, Marchant and Ramos use a multi-layered Bayesian optimization approach to generate maximum information sampling paths [79], a field known as informative path planning.

The problem of informative path planning is known to be NP-hard and there are many algorithms aimed at efficiently finding paths that maximize information gain subject to budget constraints (e.g., time, distance, etc.). Randomized approaches were explored in the work of Hollinger [48] by adapting sampling-based motion planning such as RRT* and PRM* [50] to include expected information gain in the cost of each path segment. Extensions to reduce planning time for real-time operation can be achieved by restricting the space from which paths are sampled [3]. To avoid discretization of the environment, Hitz et al. use evolutionary techniques to optimize a parameterized B-spline in continuous space [47]. Further, Reid et al. explore hierarchical informative planning with applications to sampling using reconfigurable systems such as a planetary rover [109]. Additional approaches to informative path planning based on hypothesis testing [67], fast marching methods [59], and orienteering [10] have also been proposed.

The idea of model-based informative planning has been applied extensively in the marine domain, a recent example of which uses partially-observable Markov decision processes for coral reef localization [32]. Adaptive sampling in marine environments is a natural application due to operational constraints such as limited communication bandwidth. In these instances, it is preferred to design planning methods that allow the system to adapt without human interaction. A series of studies by Manjanna et al. applied adaptive sampling to surface vessels focusing on policy search [77], multi-scale resolution [74], and integration with static sensor nodes [75]. A second series of studies by Das et al. provide sampling strategies for environmental monitoring off the California coast. Initial work in this series used a combination of Lagrangian drifters and AUVs to track and sample algal blooms, where drifters are used to identify a region of interest that is then sent to the AUV for further sampling [24]. Subsequent work proposed an approach that uses observable environmental features (e.g., salinity or temperature) to inform where unobservable features (e.g., organism abundance) should be collected to reduce uncertainty in a combined environmental model [22]. In addition, Das et al. employ prior sampling campaigns to inform the collection of a very limited number of plankton samples [23].

An outstanding challenge in the area of marine monitoring is measurement collection over a sustained period of time to understand the temporal dynamics of the environment. Initial work in this area by Leonard et al. [61] used coordinated control of a fleet of underwater gliders to intelligently collect samples over several months, while the work of Smith et al. [114] improved long-term monitoring for gliders by developing planning techniques that considered both information gain and regions of undesirable ocean currents.

Further work by Ma et al. [73, 72] specifically includes dynamics of the environment to capture temporal effects when planning sampling paths.

Teaming approaches as conceptualized in Fig. 2.2 have also gained popularity due to the challenges of long-term monitoring in large areas [89, 24, 70, 63]. A key advantage of robot teaming is that the strengths of each individual vehicle can be exploited to improve the overall system performance. For example, the mobility advantage of aerial vehicles can be used to collect sparse samples over large distances that identify general areas for further examination. Subsequently, an AUV could be deployed to these areas to collect dense measurements on a smaller scale near the seafloor to localize ROIs. Additionally, surface vessels can operate as a charging station and communication hub between vehicles. Manjanna et al. [76] employ this concept by using a team of surface vessels to coordinate both exploring the environment and collecting physical water samples. Similarly, work by Munafo [94] uses AUV teams to minimize the uncertainty in the estimated GP model while staying within communication range, whereas other methods examine the performance of decentralized coordination as a function of the number of agents in the team [101]. Alternative planning methods proposed for AUV teams use RRT* and a selective basis Kalman filter [21].

Of particular importance in adaptive sampling methods is the choice of the acquisition function for evaluating candidate sample locations. Common approaches for this optimization criteria are based on entropy [70], mutual information [56], or the combination of the predictive mean and information gain [69]. Additionally, work by Cao et al. [15] compares the effectiveness of

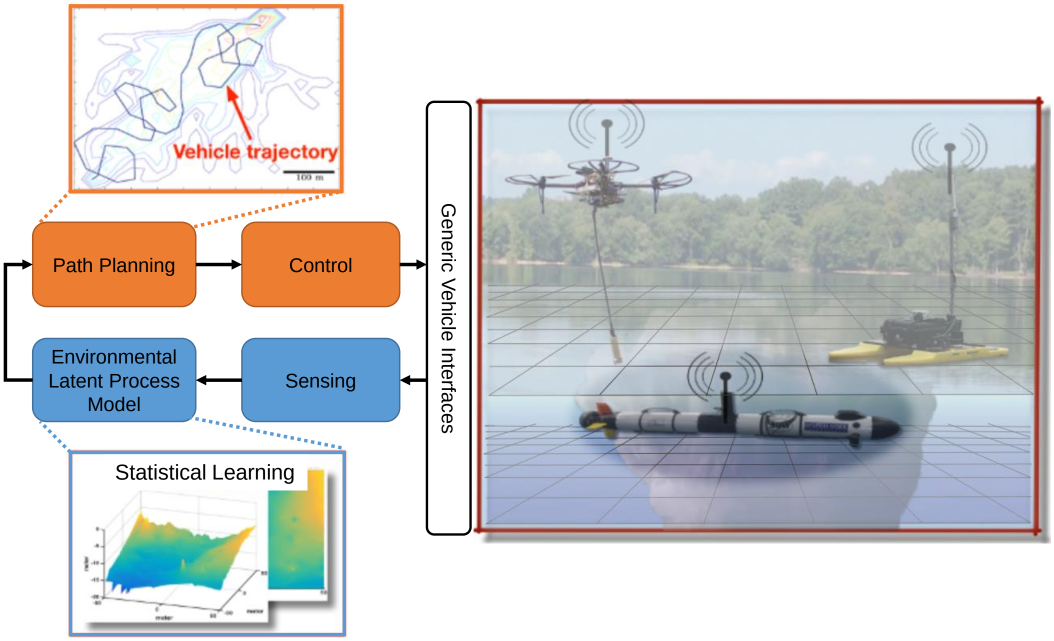


Figure 2.2: Concept of environmental sampling using a heterogeneous robot team. High-mobility vehicles such as UAVs and ASVs collect sparse samples over a large area, whereas AUVs provide focused samples in a region of interest (this paper focuses on the latter).

entropy and mutual information criteria. In this work, we adopt the Gaussian process upper confidence bound (GP-UCB) algorithm [117] as the criteria for choosing future sampling locations. The GP-UCB algorithm calculates the utility of future sampling locations according to a weighted combination of both their predictive mean and variance, thus allowing control over the exploration-exploitation trade-off at different stages of the sampling mission. The following sections introduce the problem formulation and detail both the Gaussian process model estimation as well as the formulation of the GP-UCB as a sequential stochastic optimization problem.

Chapter 3

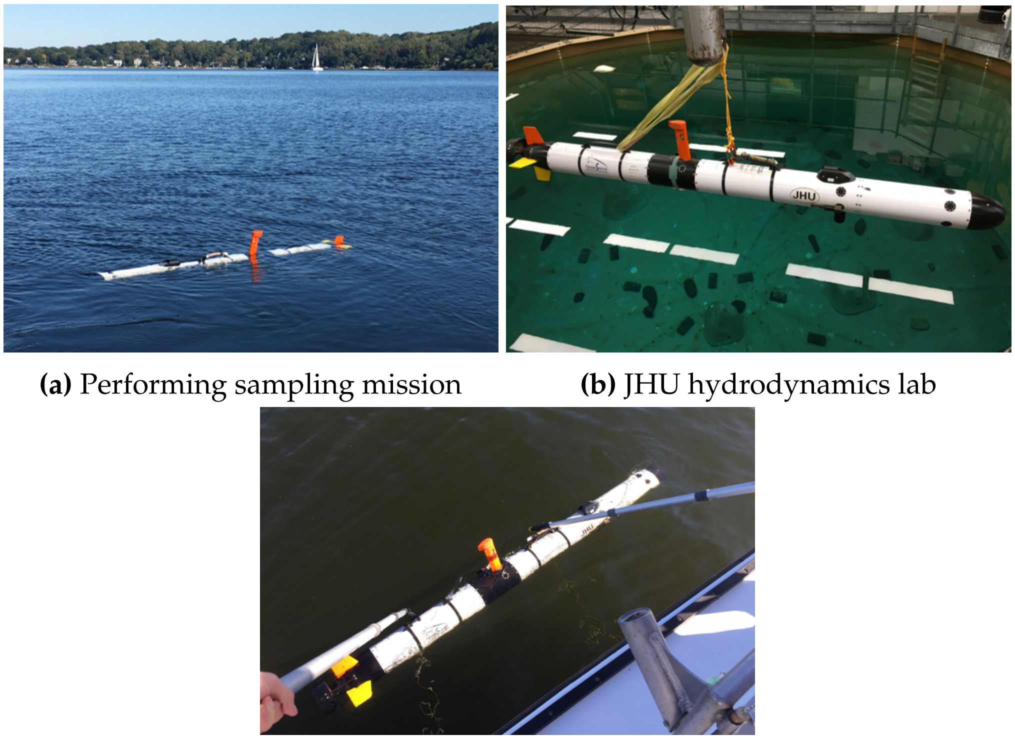
Methodology for Information-Seeking Path Planning

3.1 Problem Setup

The informative path planning problem takes on the form given by Eq. (1.1), where we would like to maximize the collective information over the planned trajectory. Next, we define the specific vehicle dynamics, their parameterization through motion primitives, and the acquisition function over the environment as a specific application of this problem statement.

3.1.1 Vehicle Dynamic Model

The robotic sampling platform employed in this problem formulation is a torpedo-style AUV. Specifically, the Johns Hopkins University (JHU) L3 OceanServer Iver3 (Fig. 3.1) is used as the testbed. This AUV is actuated by a rear propeller for thrust, a pair of rudder fins for yaw control, and a pair of diving planes for pitch control. Because this system is underactuated, it is



(a) Performing sampling mission **(b)** JHU hydrodynamics lab

(c) Deployment from vessel

Figure 3.1: Johns Hopkins University L3 OceanServer Iver3 AUV.

particularly important to consider the motion constraints of the vehicle during sample collection.

The hydrodynamic model used for this vehicle is a simplified version of the equations of motion provided by Gertler and Hagen [37]. The 4-DOF model presented here captures realistic vehicle motion while minimizing the number of required model coefficients. Thus, sufficient fidelity is captured for simulating high-level motion planning while low-level control (such as roll stability) is assumed to be controlled by the vehicle's on-board autopilot.

The state $\boldsymbol{x} = (p_x, p_y, p_z, \theta, \psi, u, w, q, r)$ includes the position $\boldsymbol{p} = (p_x, p_y, p_z) \in \mathbb{R}^3$, pitch θ, yaw ψ, surge velocity u, heave velocity w, pitch rate q, and yaw rate r. The control inputs $\boldsymbol{u} = (\delta_u, \delta_q, \delta_r)$ define the normalized thrust input, normalized pitch rate input, and normalized yaw rate input, respectively. These control inputs are scaled and constrained in order to produce high-level motion characteristics at steady-state. The normalized thrust input $\delta_u \in [0, 1]$ represents the range between zero and maximum propeller speed, the normalized pitch rate input $\delta_q \in [-1, 1]$ represents the range between maximal deflections of the vehicle diving planes, and the normalized yaw rate input $\delta_r \in [-1, 1]$ represents the range between maximal deflections of the vehicle rudder fins.

Assuming negligible effects in the sway and roll dynamics, the equations of motion $\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u})$ are given as the following:

$$\dot{p}_x = u \cos\psi \cos\theta + w \cos\psi \sin\theta, \tag{3.1a}$$

$$\dot{p}_y = u \sin\psi \cos\theta + w \sin\psi \sin\theta, \tag{3.1b}$$

$$\dot{p}_z = w \cos\theta - u \sin\theta, \tag{3.1c}$$

$$\dot{\theta} = q, \tag{3.1d}$$

$$\dot{\psi} = r / \cos\theta, \tag{3.1e}$$

$$\dot{u} = \overline{X}_{uu} u^2 + \overline{k}\delta_u, \tag{3.1f}$$

$$\dot{w} = \overline{Z}_{w|w|} w|w| + \overline{WB} \cos\theta, \tag{3.1g}$$

$$\dot{q} = \overline{M}_{uq} uq + \overline{M}_q q - \overline{Bz_B} \sin\theta + \overline{b}u^2\delta_q, \tag{3.1h}$$

$$\dot{r} = \overline{N}_{ur} ur + \overline{c}u^2\delta_r, \tag{3.1i}$$

where $(\bar{\cdot})$ indicates that the hydrodynamic coefficients have been made independent of the vehicle's mass and inertia. Further, $\overline{WB}$ represents an out-of-ballast term based on the vehicle's weight and buoyancy ratio, and $\overline{Bz_B}$ represents a buoyancy term that accounts for the center of buoyancy vertical offset from the center of gravity.

The coefficient values for the JHU Iver3 were determined using nonlinear gray-box model estimation (available in MATLAB [81]) that minimizes the normalized root-mean-square error between the model response and experimental data. The experimental data used to fit the model consisted of the

Table 3.1: Hydrodynamic model coefficients for the Johns Hopkins University Iver3 AUV.

Coefficient	Value	Units
$\overline{X}_{uu}$	-0.179	1/m
$\overline{k}$	0.519	m/s^2
$\overline{Z}_{w\lvert w\rvert}$	0.098	1/m
$\overline{WB}$	-2.452	m/s^2
$\overline{M}_{uq}$	-3.519	1/m
$\overline{M}_{q}$	-0.748	1/s
$\overline{Bz_B}$	8.947	1/s^2
$\overline{b}$	3.096	1/m^2
$\overline{N}_{ur}$	-0.441	1/m
$\bar{c}$	0.065	1/m^2

Iver3 executing a series of undulating lawnmower trajectories under calm conditions. The final values of these coefficients are listed in Table 3.1.

3.1.2 AUV Motion Parameterization

We now define the parameterization given in Section 1.3.2 for the hydrodynamic model of Eq. (3.1). For our specific setting, a trajectory is parameterized using a sequence of H motion primitives, where in this context a primitive is defined as a section of a helix with constant horizontal forward velocity v_x, depth rate v_z, and turn rate ω_z. More formally, the trajectory parameter is $\boldsymbol{z} = \{(v_x, v_z, \omega_z)_j\}_{j=1}^{H}$. During a time interval, $t \in [t_j, t_{j+1}]$, the trajectory state

$\boldsymbol{x}(t) = \varphi_{\boldsymbol{z}}(t)$ takes the form

$$\theta(t) = \arctan(v_z / v_x), \tag{3.2a}$$

$$\psi(t) = \psi(t_j) + \delta t_j \omega_z, \tag{3.2b}$$

$$u(t) = \sqrt{v_x^2 + v_z^2}, \tag{3.2c}$$

$$w(t) = \big(v_z + u(t) \sin\theta(t)\big) / \cos\theta(t), \tag{3.2d}$$

$$q(t) = 0, \tag{3.2e}$$

$$r(t) = \omega_z \cos\theta(t), \tag{3.2f}$$

$$p_x(t) = p_x(t_j) + \delta t_j \big(u(t) \cos\psi(t) \cos\theta(t) + w(t) \cos\psi(t) \sin\theta(t)\big), \tag{3.2g}$$

$$p_y(t) = p_y(t_j) + \delta t_j \big(u(t) \sin\psi(t) \cos\theta(t) + w(t) \sin\psi(t) \sin\theta(t)\big), \tag{3.2h}$$

$$p_z(t) = \max\{p_z(t_j) + \delta t_j v_z, 0\}, \tag{3.2i}$$

where $\delta t_j = t - t_j$. In other words, we have established a mapping between the simple and geometrically intuitive parameters $\boldsymbol{z}$ and trajectory in the full state space $\mathbb{X}$. A visualization of three example trajectories using different parameter values of $\boldsymbol{z}$ is shown in Fig. 3.2.

3.1.3 Environmental Modeling and Prediction as a Gaussian Process

The field of interest is defined as a continuous function $h : \mathcal{W} \to \mathbb{R}$ that maps locations $\boldsymbol{p} \in \mathcal{W}$ in the environment workspace of interest $\mathcal{W} \subset \mathbb{R}^3$ to the

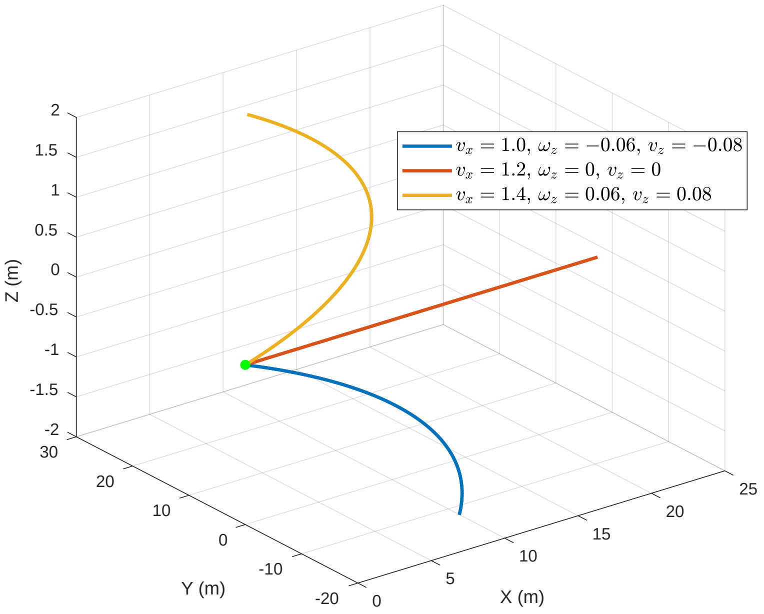


Figure 3.2: Visualizations of three trajectories generated from different values of $\boldsymbol{z}$ over a period of 20 sec. Units for v_x, ω_z, and v_z are m/s, rad/s, and m/s, respectively.

scalar environmental phenomena under study. Each measurement of this phenomena, denoted by $y \in \mathbb{R}$ is assumed to have some spatial regularity (i.e., measurements at nearby points have a physically meaningful correlation) and in addition is assumed to be corrupted by Gaussian noise: $y = h(\boldsymbol{p}) + \mathcal{N}(0, \sigma(\boldsymbol{p})^2)$. These assumptions allow the environment to be modeled as a Gaussian process (GP): a collection of dependent random variables, one for each $\boldsymbol{p} \in \mathcal{W}$, every finite subset of which is multivariate Gaussian-distributed [108]. A GP can be fully specified by its mean function $\mu(\boldsymbol{p}) = \mathbb{E}[h(\boldsymbol{p})]$ and covariance function $\kappa(\boldsymbol{p}_i, \boldsymbol{p}_j) = \mathrm{cov}(\boldsymbol{p}_i, \boldsymbol{p}_j | \boldsymbol{\theta})$ for all $\boldsymbol{p}_i, \boldsymbol{p}_j \in \mathcal{W}$, where $\boldsymbol{\theta}$ parameterizes the covariance function that models the spatial correlation of the environmental phenomena. A popular form for the covariance function is the squared exponential function [108]:

$$\kappa(\boldsymbol{p}_i, \boldsymbol{p}_j) = \sigma_f^2 \exp\left(-\frac{1}{2}\sum_{k=1}^{d}\frac{(\boldsymbol{p}_{i,k} - \boldsymbol{p}_{j,k})^2}{l_k^2}\right) + \delta(\boldsymbol{p}_i, \boldsymbol{p}_j)\sigma_w^2. \tag{3.3}$$

The squared exponential covariance function for this 3D application has a hyperparameter vector $\boldsymbol{\theta} = [l_x, l_y, l_z, \sigma_f, \sigma_w]^T$, where σ_f and σ_w are the variance of the signal and noise, respectively. The parameters l are the length-scales in each dimension, which determine the level of correlation between measurements. $\delta(\boldsymbol{p}_i, \boldsymbol{p}_j)$ is the Kronecker delta function, which is one if and only if $\boldsymbol{p}_i = \boldsymbol{p}_j$ and zero otherwise.

Let $P = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_N\} \subset \mathcal{W}$ be a set of N observed locations with recorded measurements $Y = \{y_1, \ldots, y_N\}$. The GP representation allows us to predict measurements Y_* at a finite set of unobserved locations $P_* \subset \mathcal{W}$ by defining the joint distribution of measurements at observed and unobserved locations [108]:

$$\begin{bmatrix} Y \\ Y_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K + \sigma_w^2 I & K_*^T \\ K_* & K_{**} \end{bmatrix}\right), \tag{3.4}$$

where $K_* = K(P_*, P)$ contains the pairwise covariances between unobserved and observed points, with analogous definition of $K = K(P, P)$ and $K_{**} = K(P_*, P_*)$. Conditioning on the known data [108], the unobserved measurements take the form $Y_* \sim \mathcal{N}(\widehat{Y}_*, \widehat{\Sigma}_*)$ with

$$\widehat{Y}_* = K_*(K_y)^{-1}Y, \tag{3.5}$$

$$\widehat{\Sigma}_* = K_{**} - K_*(K_y)^{-1}K_*^T, \tag{3.6}$$

where $K_y = K + \sigma_w^2 I$.

3.1.4 Hyperparameter Adaptation

The GP model accuracy is highly dependent on the hyperparameter vector $\boldsymbol{\theta}$ that is used on the covariance function. When operating in an unknown environment, it is unlikely that the correct hyperparameters for a given environmental distribution will be known a priori. Thus, a common approach is to learn the hyperparameter vector over time as measurements are collected. In this work, we employ the maximization of the log of the marginal likelihood [95] to obtain the best estimate of the hyperparameter vector $\boldsymbol{\theta}^*$ after the collection of n samples:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \ \log p(Y|P), \tag{3.7}$$

where

$$\log p(Y|P) = -\frac{1}{2}Y^T K_y^{-1} Y - \frac{1}{2}\log|K_y| - \frac{n}{2}\log 2\pi,$$

with the optimization performed using the gradient

$$\frac{\partial}{\partial\theta_i}\log p(Y|P) = \frac{1}{2}\mathrm{tr}\left(\left(\left(K^{-1}Y\right)\left(K^{-1}Y\right)^T - K_y^{-1}\right)\frac{\partial K_y}{\partial\theta_i}\right).$$

The hyperparameter vector is initialized based on expert knowledge of the environmental area and is then periodically re-estimated every ℓ samples (this work applies $\ell = 1$).

3.1.5 Acquisition Function

For a robotic system that collects environmental data sequentially, it is important to balance data collection between exploration and exploitation, i.e., sampling at locations with high uncertainty against locations with high predicted reward. Concentrating the sensing efforts within a specific area prematurely may miss ROIs in other areas of the environment, whereas aimlessly exploring the environment would forgo opportunities to collect important samples around the ROIs. The goal is to develop a policy that determines the optimal sequence of future sampling locations, taking into account both the GP model mean (predicted reward) through Eq. (3.5) and the GP model variance (predicted uncertainty) through Eq. (3.6), both of which are calculated using the current best estimate of the hyperparameter vector.

In this work, we adopt the GP upper confidence bound (GP-UCB) [117, 23] criteria for guiding sample selection. While entropy-based methods tend to reduce uncertainty over the full sampling area, UCB-based acquisition functions have been shown to provide better resolution of ROIs [79]. From a GP optimization perspective, the GP-UCB algorithm is designed to minimize the cumulative regret over the collection of N samples, i.e., the loss in reward from not selecting the optimal set of N samples. While sensor data can typically be collected at high rates, it becomes computationally infeasible to train the GP model on the entire dataset from the sensor feed. Downselecting this data to the optimal N samples makes the training process tractable. This strategy applies the principle of optimism in the face of uncertainty, where the selection of a sample position $\boldsymbol{p}_n \in \mathcal{W}$ at sampling step n is evaluated according to

$$L(\boldsymbol{x}_n, \boldsymbol{u}_n) = \underbrace{\mu_{n-1}(\boldsymbol{p}_n) + \sqrt{\beta_n}\sigma_{n-1}(\boldsymbol{p}_n)}_{\text{sampling reward}} \quad - \quad \underbrace{q(\boldsymbol{x}_n)}_{\text{navigation cost}} \quad , \tag{3.8}$$

where $\mu = \widehat{Y}_*$ and $\sigma^2 = \widehat{\Sigma}_*$ as described in Section 3.1.3. All sampling points along a trajectory are treated equally, meaning that the terminal reward is set to $\phi(\boldsymbol{x}_n(t_f)) = 0$.

The first component of the acquisition function is the sampling reward, which encourages the collection of informative samples. This reward depends only on the sampling locations and is based on the trade-off between potentially maximizing the GP model mean (choosing locations within ROIs characterized by maximal sensory measurements) and maximizing the GP model variance (exploring uncertain and/or unobserved areas of the environment workspace). Alternatively, if an ROI is described by minimal measurement values, as would be the case for detecting hypoxic zones, one would simply maximize the negative of the GP model mean, $-\mu$, to encourage sample collection within ROIs. The term β_n employed in Eq. (3.8) is the mean-variance trade-off parameter at sampling step n [117]:

$$\beta_n = 2 \log \left(\frac{|d| n^2 \pi^2}{6\delta} \right), \tag{3.9}$$

where d is the dimensionality of $\mathcal{W}$ and $\delta \in [0, 1]$ is a parameter that defines the probability that the regret bound is satisfied after N sampling steps, for $n = 1 \ldots N$ [117].

The second component of Eq. (3.8) is a standard non-negative navigation cost, which heavily penalizes trajectories that exit the environment workspace or approach a forbidden state set $\mathcal{O}$ (e.g., known obstacles or shorelines from nautical charts, the seafloor, etc.). Here, an infinite step function is applied to undesirable states as such:

$$q(\boldsymbol{x}_n) = \begin{cases} \infty & \text{if } \boldsymbol{p}_n \notin \mathcal{W} \text{ or } \boldsymbol{x}_n \in \mathcal{O} \\ 0 & \text{otherwise.} \end{cases} \tag{3.10}$$

While this work does not consider real-time obstacle detection, future work could extend the navigation cost by augmenting $\mathcal{O}$ with detected obstacles.

3.2 Adaptive Trajectory Planning

This section now proposes an algorithmic approach for adaptive environmental sampling and explores two informative path planning methods for solving the optimization of Eq. (1.6) and (1.7). Algorithm 1 details the overall framework of the proposed GP adaptive sampling (GPAS) method. In this framework the GP is initialized to a known prior or, if no known prior is available, it is set to a uniform constant field with a given variance. The first measurement y_0 at the starting location $\boldsymbol{p}_0$ is then added to the model, and subsequent sampling points $\boldsymbol{p}_n$ at each iteration are selected through the PATHPLANNING subroutine of Alg. 1.

For a realistic sampling mission duration, planning a trajectory over a long horizon is often intractable. Real-time computation can be achieved, however, by calculating a sub-optimal approximation of the full trajectory over a local planning horizon T. A receding horizon strategy is implemented here such that at every sampling step n (i.e., after each collected measurement), a trajectory $\boldsymbol{x}^*([t, t+T])$ is generated containing a sequence of H future sampling points $\{\boldsymbol{p}_i^*\}_{i=n}^{n+H}$, so that $T = H\delta_t$ where δ_t is a fixed sampling time step. Once sampling is performed at $\boldsymbol{p}_n^*$, the corresponding measurement y_n

Algorithm 1 GAUSSIAN PROCESS ADAPTIVE SAMPLING (GPAS)

```
procedure GPAS(GP, $x_0$, $t_f$)
    ▷ Collect measurement at starting location
    $y_0 \leftarrow$ MEASUREMENT($\boldsymbol{p}_0$)
    ▷ Add starting location and measurement to the GP
    $\mathrm{GP}_0 \leftarrow$ GP_INITIALIZE(GP, $\boldsymbol{p}_0, y_0$)
    Let $t \leftarrow 0$;  $n \leftarrow 0$
    repeat
        Set $t \leftarrow t + \delta_t$;  $n \leftarrow n + 1$
        ▷ Plan a sampling path up to planning horizon $T$
        $\boldsymbol{x}^*([t, t+T]) \leftarrow$ PATHPLANNING($\mathrm{GP}_{n-1}, \boldsymbol{x}(t)$)
        ▷ Collect measurement at first sampling location along path
        $y_n \leftarrow$ MEASUREMENT($\boldsymbol{p}_n^*$)
        ▷ Add sampling location and measurement to the GP
        $\mathrm{GP}_n \leftarrow$ GP_ASSIMILATE($\mathrm{GP}_{n-1}, \boldsymbol{p}_n^*, y_n$)
        ▷ Estimate GP hyperparameters with updated GP
        $\boldsymbol{\theta}_n^* \leftarrow$ ESTIMATEHYPERPARAM($\mathrm{GP}_n$)        ▷ Eq. (3.7)
    until $t = t_f$
end procedure
```

is assimilated into the GP model and the hyperparameters are updated. This process is repeated until the final mission time t_f. While this receding-horizon strategy is not globally optimal, it permits real-time execution and in situ field estimation on-board the sampling platform, which is our main objective.

Two approaches are presented to generate the sampling trajectory in the PATHPLANNING subroutine of Alg. 1. The first method is a modified branch-and-bound informative path planner (Section 3.2.1) and the second method is a cross-entropy informative path planner (Section 3.2.2). These two planning algorithms are presented and compared because of their respective strengths and weaknesses in different applications. The foundations of branch-and-bound techniques are well-suited for efficient informative path planning [9, 83], particularly on low-cost platforms with limited computational resources. However, generation of the candidate planning tree is often dependent on sub-optimal heuristics. Alternatively, cross-entropy-based methods, while more computationally demanding, are commonly used in reinforcement learning frameworks due to the desirable quality that they need not make any assumptions about the underlying (potentially black box) system. Thus, both algorithms offer value during informative path planning and the utility of each is analyzed in the following sections.

It should also be noted that the scale of the environmental distributions under study requires that the motion constraints of the sensing platform be taken into consideration during planning. More specifically, the objective is to capture scalar fields that exhibit variability with resolution on the scale of the vehicle's turning radius. In contrast to other methods [114, 72, 22] where planning is performed within a sampling grid of several miles, both of the methods presented here explicitly consider vehicle motion constraints with the intention of sampling highly localized ROIs in a 3D environment.

3.2.1 Branch & Bound Informative Path Planning (BB-IPP)

The computation time of a finite-horizon planner is exponential in the length of the horizon. Thus, exhaustive search methods that evaluate the reward function for every possible sampling path are only feasible for short planning horizons. Branch-and-bound algorithms offer an efficient alternative by pruning large portions of the search tree that are unlikely to contain the optimal solution based on an upper bound on the estimated reward.

Binney [9] explored branch and bound for informative path planning between two points on a coarsely discretized 2D sampling grid. There are three ways in which the method presented here differs: (i) we apply branch and

Algorithm 2 BRANCH & BOUND INFORMATIVE PATH PLANNING (BB-IPP)

```
procedure PATHPLANNING_BB(GP, $\boldsymbol{x}_{start}$)
    ▷ Set $\gamma^* \leftarrow 0$
    ▷ Generate a set of $Q$ query points, $\{\boldsymbol{p}_i\}_{i=1}^{Q} \in \mathcal{W}$
    ▷ Predict query point rewards $\{L(\boldsymbol{p}_i)\}_{i=1}^{Q}$ using Eq. (3.5), (3.6), and (3.8)
    ▷ Sort by increasing $L_i$ and assign the upper bound as the $\alpha$-percentile $L_{UB} \leftarrow L_{\lceil \alpha Q \rceil}$
    $[\boldsymbol{z}^*, \gamma^*] \leftarrow$ BB_RECURSION($\boldsymbol{x}_{start}, \emptyset, 0, 0$)
    return $\varphi_{\boldsymbol{z}^*}$
end procedure

procedure BB_RECURSION($\boldsymbol{x}$, $\boldsymbol{z}_{parent}$, $\gamma_{parent}$, $j$)
    if $j < H$ then
        ▷ Approximate maximum possible reward over the planning horizon
        $\gamma_{max} \leftarrow \gamma_{parent} + (H - j)L_{UB}$
    else
        $\gamma_{max} \leftarrow -\infty$
    end if
    if $\gamma_{max} > \gamma^*$ then
        ▷ Sample $\{\xi_i\}_{i=1}^{M}$ (consisting of primitives over $[t + j\delta_t, t + (j+1)\delta_t]$) according to Eq. (3.11)
        ▷ Extend trajectories, $Z \leftarrow \{(\boldsymbol{z}_{parent}, \xi_i)\}_{i=1}^{M}$
        ▷ Predict reward of each trajectory $\{\gamma_i\}_{i=1}^{M}$ using Eq. (3.5), (3.6), (3.8), and (1.5)
        ▷ Sort $Z$ by decreasing $\gamma_i$
        for each $\boldsymbol{z}_{child} \in Z$ do
            ▷ Update total reward at child, $\gamma_{child} \leftarrow \gamma_i$
            if $\gamma_{child} > \gamma^*$ then
                $\gamma^* \leftarrow \gamma_{child}$, $\boldsymbol{z}^* \leftarrow \boldsymbol{z}_{child}$
            end if
            $[\boldsymbol{z}^*, \gamma^*] \leftarrow$ BB_RECURSION($\varphi_{\boldsymbol{z}_{child}}(t + \delta_t))$, $\boldsymbol{z}_{child}$, $\gamma_{child}$, $j + 1$)
        end for
    end if
    return $\boldsymbol{z}^*, \gamma^*$
end procedure
```

bound to a search tree constructed of feasible motion primitives over a finite horizon, (ii) there is no fixed goal point towards which the algorithm presented here is driving, and (iii) as opposed to calculating an upper bound based on average variance reduction of a GP model, we calculate the upper bound based on an estimate of the ideal utility (Eq. (3.8)) from a set of unobserved query points, i.e., potential future sampling locations. The overall methodology for

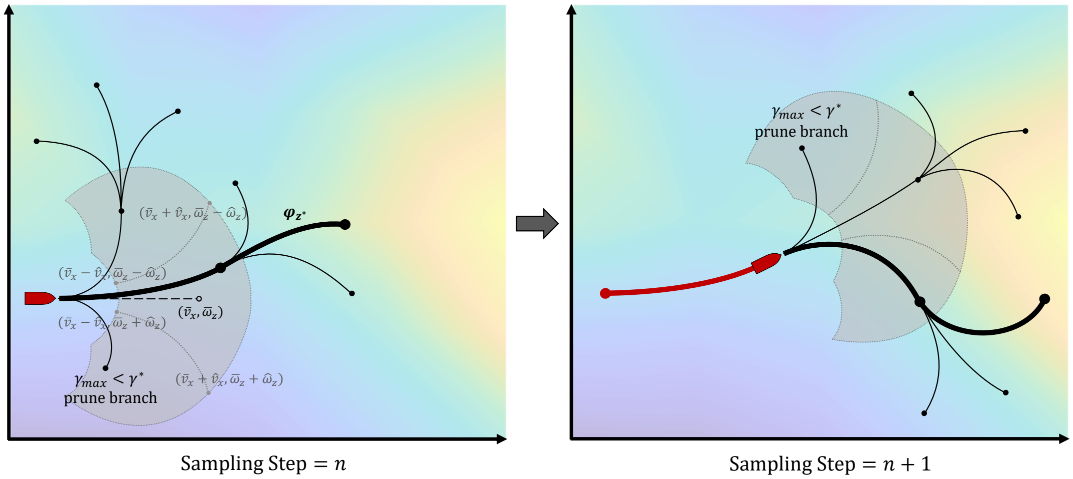

$(\bar{v}_x + \hat{v}_x, \bar{\omega}_z - \hat{\omega}_z)$
φ_{z^*}
$(\bar{v}_x - \hat{v}_x, \bar{\omega}_z - \hat{\omega}_z)$
$(\bar{v}_x, \bar{\omega}_z)$
$(\bar{v}_x - \hat{v}_x, \bar{\omega}_z + \hat{\omega}_z)$
$\gamma_{max} < \gamma^*$
prune branch
$(\bar{v}_x + \hat{v}_x, \bar{\omega}_z + \hat{\omega}_z)$
Sampling Step $= n$
$\gamma_{max} < \gamma^*$
prune branch
Sampling Step $= n + 1$

Figure 3.3: Illustrative example of the receding horizon BB-IPP algorithm driving towards a high-reward (yellow) ROI along the right edge of the sampling area. This 2D example uses $H = 2$ and trajectory parameterizations of $\boldsymbol{z} = (v_x, \omega_z)$ corresponding to speed and turn rate, respectively. For $v_x > 0$, the gray region represents the reachable space for one branching step over $t + \delta_t$.

branch-and-bound informative path planning (BB-IPP) is provided in Alg. 2 and illustrated with a 2D example in Fig. 3.3.

3.2.1.1 Branching Procedure

Algorithm 2 is based on a recursive best-first tree search adapted to the specific type of trajectory parametrization and heuristics employed. A tree with candidate trajectories over $[t, t + T]$ is constructed recursively starting from the current state, $\boldsymbol{x}_{start}$. States in the tree are extended at each level by a set of M feasible motion primitives over δ_t that form the edges of the tree. It is desired that the set of candidate trajectories covers the reachable space of the system while minimizing M. To achieve this, trajectory parameterizations are generated through *stratified rejection sampling* by partitioning the domain into several semantically different classes, i.e., “forward”, “left”, “right”, “down”, and “up”, to ensure balanced selection of primitives in each direction. Then each class is sampled uniformly, with primitives that violate the constraints (e.g., collisions) discarded. The main reason for stratifying the space is to achieve good coverage of the domain even with few samples, motivated by our real-time computation requirements. The quality of this proposed sample selection is analyzed in Section 3.2.1.2.

More formally, let $\xi = (v_x, v_z, \omega_z)$ be a primitive parameterization and Ξ be the parameter space over a single stage δ_t. This parameter space is partitioned into $\Xi = \{\Xi_1, \dots, \Xi_M\}$, where

$$\Xi_i = [\bar{v}_{x,i} - \hat{v}_x, \bar{v}_{x,i} + \hat{v}_x] \times [\bar{v}_z - \hat{v}_{z,i}, \bar{v}_{z,i} + \hat{v}_z] \times [\bar{\omega}_{z,i} - \hat{\omega}_z, \bar{\omega}_{z,i} + \hat{\omega}_z],$$

with $\bar{v}_x, \bar{v}_z, \bar{\omega}_z$ denoting the chosen nominal velocity, depth rate, and turn rate, respectively, while $\hat{v}_x, \hat{v}_z, \hat{\omega}_z$ can be regarded as perturbations around the nominal values. The nominal values and perturbations are chosen to fully cover the space without any overlap. The set of single-stage primitives $\{\xi_i\}_{i=1}^{M}$ is then sampled according to

$$\xi_i \sim \mathit{Uniform}(\{\xi \in \Xi_i \mid g(\Gamma(\varphi_\xi(\tau))) \leq 0,\ t \in [t, t + \delta t]\}). \tag{3.11}$$

The multi-stage trajectory parameterization over the planning horizon then follows as $z = (\xi_1, \dots, \xi_H)$, while the parameter sample space is defined as $\mathcal{Z} = \Xi^H$. The mapping from parameter space to a state space trajectory is achieved by the function φ as outlined in Section 3.1.2.

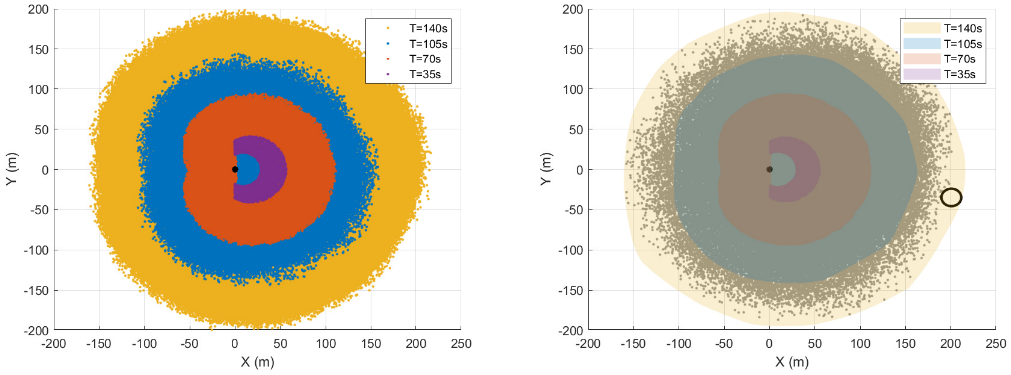
3.2.1.2 Branching Reachability Analysis

This section further examines the reachability characteristics of the branching procedure outlined in Section 3.2.1.1. The branching procedure is based on sampling motion primitives defined in velocity space; however, since the objective function is defined over the position workspace $\mathcal{W}$, a key requirement is the ability to explore that workspace with high probability in order to discover regions of high information content, especially since early stages greatly affect the sampling positions of later stages. Therefore, we establish a relationship between the velocity-space sampling process and the resulting coverage of $\mathcal{W}$ to ensure that the branching procedure does not omit key areas of the reachable workspace.

As a guideline, our goal is to ensure that with high probability we can visit any spatial region with a given radius R_{min} over the reachable space for a given planning horizon. A typical choice of this radius corresponds to the vehicle's turning radius of approximately $R_{min} = 15$ meters. Further, let $\mathcal{W}_{\boldsymbol{x}_0,T} \subset \mathcal{W}$ denote the reachable workspace starting at some state $\boldsymbol{x}_0$ over a horizon T and let the set of samples generated by the search algorithm be denoted by $\widehat{\mathcal{W}}_{\boldsymbol{x}_0,T} \subset \mathcal{W}_{\boldsymbol{x}_0,T}$. We require that with high confidence $R(\widehat{\mathcal{W}}_{\boldsymbol{x}_0,T}) < R_{min}$ where

$$R(\widehat{\mathcal{W}}_{\boldsymbol{x}_0,T}) = \sup_{\boldsymbol{p} \in \mathcal{W}_{\boldsymbol{x}_0,T}} \left(\min_{\boldsymbol{p}' \in \widehat{\mathcal{W}}_{\boldsymbol{x}_0,T}} \|\boldsymbol{p} - \boldsymbol{p}'\| \right)$$

is the radius of the largest ball that does not contain a future sampling location over the reachable workspace. We limit the analysis to spatial positions in the horizontal plane due to the significantly smaller length scale of the workspace depth component. Since the distribution of $R(\widehat{\mathcal{W}}_{\boldsymbol{x}_0,T})$ is difficult to compute

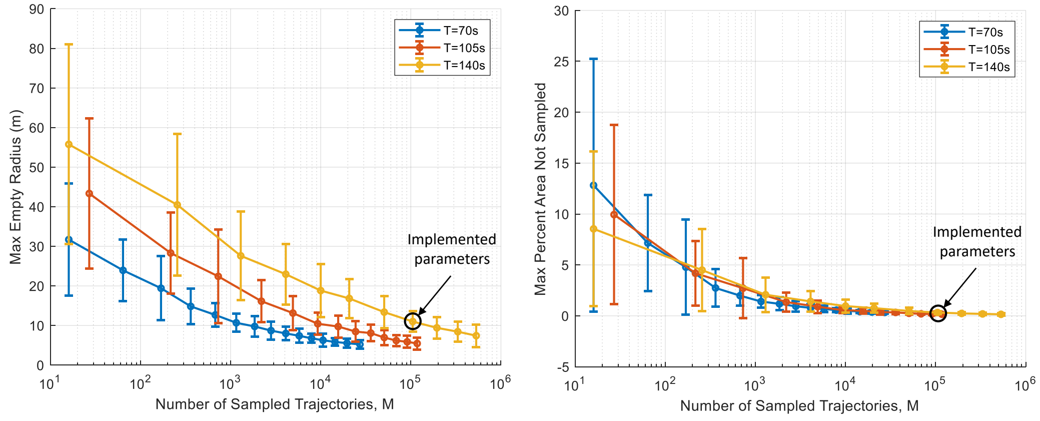


(a) **(b)**

Figure 3.4: (a) The approximate reachable workspace $\mathcal{W}_{x_0,T}$ over different planning horizons for $v_x \in [0.8, 1.6]$ m/s and $\omega_z \in [-0.09, 0.09]$ rad/s. The nonzero minimum speed means that the area immediately in front of the vehicle is not reachable for the shorter horizons. (b) Candidate sampling locations $\widehat{\mathcal{W}}_{x_0,T}$ (gray dots) evaluated for M trajectories over $T = 140$ sec. The largest empty circle for the samples is shown on the right.

analytically, i.e., it depends on a nonlinear transformation of a probability density function over the primitive sampling parameters, we instead numerically approximate its expected value and its confidence interval through offline sampling. This is accomplished by executing multiple simulations resulting in different $R(\widehat{\mathcal{W}}_{x_0,T})$ employed for sample-based confidence-interval analysis.

In each simulation, sampled locations are generated using M^H trajectories over a given planning horizon. Fig. 3.4 shows an example of approximated reachable spaces and the largest empty ball for a set of future sampling locations evaluated over $T = 140$ sec. Additionally, Fig. 3.5a shows statistics for the maximum empty ball calculated over 30 trials for various combinations of M^H and T. These results show that the parameters ultimately selected for the BB-IPP algorithm satisfy the desired exploration criteria with high probability. Specifically, after 100 simulations we obtain a sample mean for $R(\widehat{\mathcal{W}}_{x_0,T})$ of



(a) **(b)**

Figure 3.5: Monte Carlo analysis of the maximum empty radius (a) and maximum percentage of empty reachable space (b) as a function of the number of sampled trajectories for various planning horizons. Each data point shows the mean calculated over 30 trials with the error bars representing two standard deviations. The parameter combination implemented for the BB-IPP algorithm is highlighted.

11.47 meters and sample standard deviation of 0.98 meters. This corresponds to a confidence interval of

$$CI_{R(\widehat{\mathcal{W}}_{x_0,T})} = [11.21, 11.67].$$

Therefore, based on the chosen sampling parameters, with high confidence the branching procedure will not miss a region with radius $R_{\text{min}} = 15$ meters. Fig. 3.5b further emphasizes adequate coverage in the position space by showing that the largest empty ball is actually a very small percentage of the total area of the reachable space. Based on this analysis and further studies performed in Section 4.1, we pick parameters for T, M, and $\mathcal{Z}$ to achieve the desired reachability over a maximum planning horizon while also balancing computational expense.

3.2.1.3 Bounding Procedure

We have shown that the branching procedure produces desired reachability characteristics that ensure the proposed method is capable of exploring all regions of the reachable workspace. However, since the ultimate objective is not to visit a single location, a dense branching set is not necessary if it is determined in an early stage that a branch is unlikely to maximize *information*. In this sense, to avoid the exponential number of possible trajectories, it is critical to utilize a heuristic for the upper bound on the reward at any given sampling location. The goal is to choose an upper bound L_{UB} that effectively restricts the search space while not pruning branches that could contain the optimal solution.

The upper bound is computed by first generating a space-filling set of Q query points $\{\boldsymbol{p}_i\}_{i=1}^{Q}$, where $\boldsymbol{p}_i$ are candidate sampling locations within the planning horizon reachable space. These query points are evaluated using Eq. (3.8) to produce a set of rewards $\{L_i\}_{i=1}^{Q}$ that estimate the reward distribution. The upper bound on the reward L_{UB} is then assigned as the α-percentile of L_k where $L_k \in \{L_1 \geq \ldots \geq L_Q\}$. The value chosen for $\alpha \in [0,1]$ represents a trade-off between computation time and optimality: small values of α will prune many branches of the tree and reduce the computation time, but lead to a greater risk of pruning the optimal solution. Conversely, large values of α result in a strict L_{UB} threshold that explores more of the search tree. Empirical parameter tuning experiments indicate that best performance is achieved when $0.9 \leq \alpha \leq 1$. Once all branches have been explored, BB-IPP of Alg. 2 returns the optimal sampling trajectory.

3.2.2 Cross-Entropy Informative Path Planning (CE-IPP)

A second method based on cross-entropy optimization offers an alternative for use in the PATHPLANNING procedure of Alg. 1. The idea behind the cross-entropy method is to treat the optimization problem outlined in Eq. (1.6) as an estimation problem of rare events. In what follows, we present the simplified version of the cross-entropy method and refer interested readers to other related work for the detailed formulation [25, 55, 78].

Define a collection of indicator functions $\{I_{\{J(z)\geq\gamma\}}\}$ on $\mathcal{Z}$ for various thresholds or levels γ. Consider the case when $\mathcal{Z}$ has a probability density function $p(\cdot;\bar{v})$ belonging to some parametric family $\{p(\cdot;v), v \in \mathcal{V}\}$, where $\bar{v}$ is the true or nominal parameter. For instance, this could be a mixture of Gaussians. We can associate with Eq. (1.7) the following estimation problem:

$$\mathbb{P}_{\bar{v}}(J(\mathcal{Z}) \geq \gamma) = \mathbb{E}_{\bar{v}}[I_{\{J(z)\geq\gamma\}}] = \sum_{z} I_{\{J(z)\geq\gamma\}} p(z, \bar{v}). \tag{3.12}$$

The association comes from the fact that the probability $\mathbb{P}_{\bar{v}}(J(\mathcal{Z}) \geq \gamma)$ will be very small (rare event) when γ is close to γ^*. By the cross-entropy method, this rare event can be estimated by iteratively generating and updating a sequence of tuples $\{(\hat{\gamma}_j, \hat{v}_j)\}$ such that it will converge to a small region of the optimal tuple $\{(\gamma^*, \bar{v}^*)\}$.

Let v_0 be the initial parameter, for instance $v_0 = \bar{v}$, and employing a multilevel approach using a sequence, the tuples $\{(\hat{\gamma}_j, \hat{v}_j)\}_{j\geq 1}$ can be updated iteratively by:

- Let γ_j be the $(1-\rho)$-percentile of $J(\mathcal{Z})$ under v_{j-1}. An estimate of γ_j,

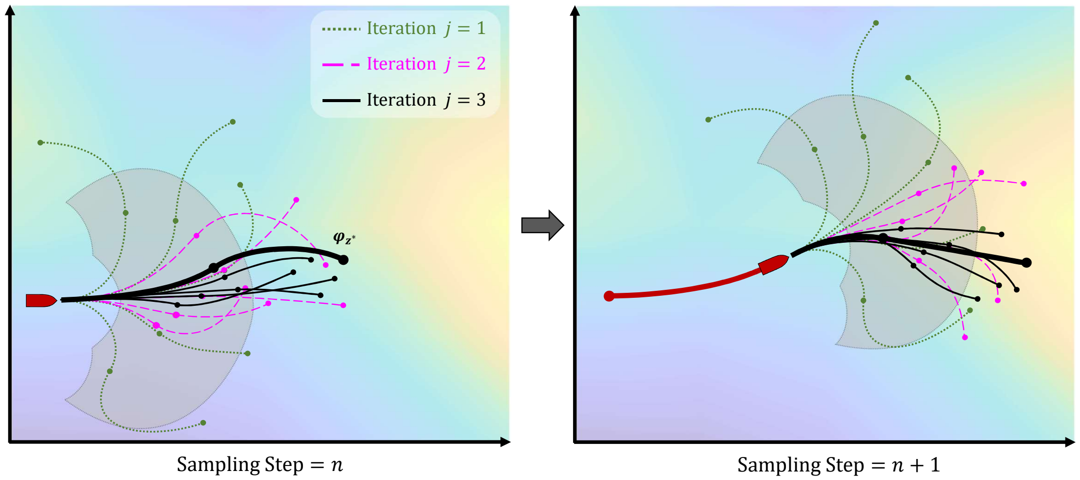


Figure 3.6: Illustrative example of the receding horizon CE-IPP algorithm driving towards a high-reward (yellow) ROI along the right edge of the sampling area. This 2D example uses $H = 2$ and trajectory parameterizations of $\boldsymbol{z} = (v_x, \omega_z)$ corresponding to speed and turn rate, respectively. For $v_x > 0$, the gray region represents the reachable space for one branching step over $t + \delta_t$.

denoted $\hat{\gamma}_j$, can be obtained by drawing M samples from $p(\boldsymbol{z}, v_{j-1})$ and assigning $\hat{\gamma}_j$ as the $(1 - \rho)$-percentile of $\{\gamma_i\}_{i=1}^{M}$.

- With fixed $\hat{\gamma}_j$ and v_{j-1}, the estimate of v_j, denoted $\hat{v}_j$, can be derived from work by Kobilarov [55]:

$$\hat{v}_j = \underset{v \in \mathcal{V}}{\text{argmax}} \frac{1}{|\epsilon_j|} \sum_{\boldsymbol{z}_i \in \epsilon_j} \ln p(\boldsymbol{z}_i, v), \tag{3.13}$$

  where ϵ_j is the elite set of the samples, i.e., samples $\boldsymbol{z}_i$ for which $\gamma_i \geq \hat{\gamma}_j$.

Instead of updating $\hat{v}_j$ directly with Eq. (3.13), we apply a simple smoothing filter:

$$\hat{v}_j = \kappa \hat{v}_j + (1 - \kappa)\hat{v}_{j-1}, \tag{3.14}$$

where κ is the smoothing parameter with $0.7 < \kappa < 1$. The filter serves two purposes: (i) smoothing the parameter update, and (ii) avoiding $\hat{v}_j$ from

Algorithm 3 CROSS ENTROPY INFORMATIVE PATH PLANNING (CE-IPP)

```
procedure PATHPLANNING_CE(GP,$\boldsymbol{x}_{start}$)
    ▷ Set $j \leftarrow 0$,$\hat{\gamma}_0 \leftarrow 0$, and $\hat{v}_0 \leftarrow v_0$
    repeat
        Set $j \leftarrow j + 1$
        ▷ Generate samples $\{\boldsymbol{z}_i\}_{i=1}^{M}$ from $p(\cdot; \hat{v}_{j-1})$, starting from $\boldsymbol{x}_{start}$
        ▷ Reject samples with constraint violations $g(\varphi_{\boldsymbol{z}_i}) > 0$
        ▷ Predict reward of each sample $\{\gamma_i\}_{i=1}^{M}$ using Eq. (3.5), (3.6), (3.8), and (1.5)
        ▷ Sort $\gamma_i$ in ascending order and compute the $(1-\rho)$-percentile $\hat{\gamma}_j \leftarrow J_{\lceil(1-\rho)M\rceil}$
        ▷ Update $\hat{v}_j$ using Eq. (3.13) over the elite set $\epsilon_j \leftarrow \{(\boldsymbol{z}_i|\gamma_i) \geq \hat{\gamma}_j\}$
    until $j = K$        ▷ $K$ is number of iterations
    ▷ Choose the sample with the maximum reward
    $\boldsymbol{z}^* \leftarrow \underset{\boldsymbol{z}_i}{\operatorname{argmax}}\ J(\boldsymbol{z}_i)$
    return $\varphi_{\boldsymbol{z}^*}$
end procedure
```

converging prematurely into a local minima, especially during the inital stage of the optimization process. A step-by-step process of the cross-entropy informative path planning (CE-IPP) is given in Alg. 3 and illustrated with a 2D example in Fig. 3.6. Once again, the sampled trajectory parameterizations for the AUV motion profile consist of a tuple sequence $\boldsymbol{z} = \{(v_x, v_z, \omega_z)_i\}_{i=1}^{H}$.

Chapter 4

Performance Evaluation of Gaussian Process Adaptive Sampling

4.1 Simulation Studies

Monte Carlo simulation studies were performed to evaluate the overall GPAS method and each informative path planner in sensing and localizing multiple ROIs. Quantitative and qualitative validation of the GPAS method is provided against conventional sensing approaches in a variety of realistic 3D environments. With regards to Chesapeake Bay water quality applications, ROIs are described here by hypoxic zones – areas of the water column characterized by a dearth of dissolved oxygen. Values for dissolved oxygen in the Chesapeake Bay typically range between 0 – 9 mg/L, with hypoxic zones designated by dissolved oxygen values less than 2 mg/L [43]. Thus, in order to maintain the assumption that measurements are normalized in the range 0 – 1, ROIs are defined as areas of the environment that exhibit normalized measurements below a threshold of $\frac{2\text{mg/L}}{9\text{mg/L}} \approx 0.22$. Hypoxic zones typically occur near the

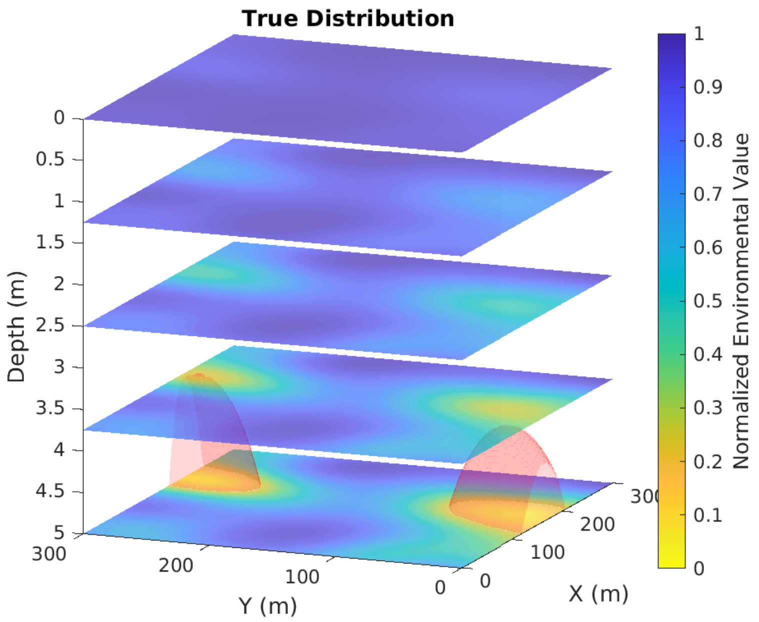


Figure 4.1: True environmental distribution used for simulation studies with a static field normalized in the range 0 – 1. ROIs are shown in red as the volume below the desired measurement threshold of 0.22.

seafloor because pycnocline layers of fresh and salt water prevent oxygen-rich surface water from mixing with deeper water. Decomposition of algae and other phytoplankton beneath the pycnocline further exacerbates the consumption of dissolved oxygen [43, 53]. To emulate this phenomenon, the ROIs in the virtual environmental fields used for the simulation studies are a linear function of depth and reach their maximum volume at the seafloor. Further, all environmental fields are assumed to be static based on the short time scale of AUV deployments within this application (on the order of hours).

Fig. 4.1 shows one such virtual environmental field used for the studies of Sections 4.1.1 and 4.1.2: a 300 meter × 300 meter × 5 meter volume with two ROIs near the seafloor. The volume of the ROIs in this environmental

Table 4.1: Summary of the performance metrics used to evaluate the planning approaches.

Metric	Description
Planning Time	The average computation time to select the optimal path at each sampling step n over the set of simulations. The simulations were run on a single laptop with a 2.8 GHz Intel i7-7700HQ processor with 16 GB of RAM. Each algorithm was implemented in Matlab on Ubuntu 18.04 for prototype analysis, although significant speed improvements could be achieved by implementing the algorithms in C++.
Reward Per Segment	The average optimal reward of the generated trajectory divided by the number of segments in the planning horizon, i.e., γ^*/H, over the set of simulations.
GP RMSE	The average root-mean-square error (RMSE) of the estimated GP model at the end of the sampling mission over the set of simulations.
GP Variance	The average variance of the estimated GP model at the end of the sampling mission over the set of simulations. The variance of the GP model is calculated as the mean of the covariance matrix diagonal.
Percent Mission in ROI	The average time spent within ROIs divided by the total mission time over the set of simulations.

field account for approximately 1.5% of the total sampling volume. No prior information of the environment is assumed and the initial density of the GP model is set to a uniform distribution across the scalar field. For each simulation, a different random seed was used for trajectory generation and the starting location of the AUV was randomly selected to be on the surface within a 25 meter × 25 meter box in the bottom left corner of the environmental area. Performance metrics used to evaluate the planning approaches are given in Table 4.1.

Additionally, low-cost AUV operations require periodic surfacing for GPS fixes in order to maintain navigation accuracy. Thus, in the simulations and field experiments that follow, a heuristic was implemented such that the vehicle surfaces for a GPS reading after traveling a user-specified distance. For instance, in order for the Iver3 AUV to maintain a dead reckoning accuracy of 5 meters (using its onboard compass, inertial measurement unit, and Doppler velocity log), the vehicle should resurface after traveling approximately 500 – 1000 meters [44] depending on the environmental conditions and the frequency of course and/or depth changes. Future work will examine methods for incorporating surfacing decisions into the path planning such as those explored by Kemna [52].

4.1.1 Algorithm Parameter Tuning

An initial Monte Carlo simulation study was performed to evaluate parameter variations of each informative path planner. Nominal parameter values for both planners that were not varied for the analysis are presented in Table 4.2. While the value of δ in Eq. (3.9) was set to 0.9, it should be noted that variations of this parameter may be insignificant [19]. Fifty simulations were executed for each parameter combination.

4.1.1.1 BB-IPP Parameter Tuning

Performance of the BB-IPP method was evaluated by varying the planning horizon T and the reward upper bound percentile α. Table 4.3 shows the resulting performance metrics for various combinations of these parameters. These results reveal that both the mean reward per trajectory segment and the mean planning time increase as the values of T and α increase. This is expected, as longer planning horizons and larger values of the reward upper bound percentile threshold correlate with evaluating a greater number of

Table 4.2: Algorithm parameters used for simulation studies.

Component	Parameter	Description
General	$N = 160$	Number of sampling points over mission ($N = t_f/\delta_t$)
	$t_f = 6000$ sec	Total sampling mission time (including surfacing events)
	$\delta_t = 35$ sec	Time step between each collected sample
	$T = 140$ sec	Planning horizon time length
	$\delta = 0.9$	Probability regret bound is satisfied after N sampling steps
BB-IPP	$\alpha = 0.95$	Upper bound percentile threshold
	$Q = 2000$	Number of query points used to estimate the utility distribution
	$\bar{v}_x \in \{1.0, 1.4\}$ m/s	Nominal speed trajectory parameterizations
	$\hat{v}_x = 0.2$ m/s	Perturbation magnitude from each $\bar{v}_x$
	$\bar{v}_z \in \{-0.08, 0, 0.08\}$ m/s	Nominal depth rate trajectory parameterizations
	$\hat{v}_z = 0.04$ m/s	Perturbation magnitude from each $\bar{v}_z$
	$\bar{\omega}_z \in \{-0.06, 0, 0.06\}$ rad/s	Nominal yaw rate trajectory parameterizations
	$\hat{\omega}_z = 0.03$ rad/s	Perturbation magnitude from $\bar{\omega}_z$
CE-IPP	$K = 9$	Maximum number of iterations
	$M = 300$	Number of trajectories evaluated at each K-th iteration
	$\rho = 0.1$	Percentile threshold for specifying elite sample set
	$\kappa = 0.9$	Parameter update smoothing filter value
	$v_x \in [0.8, 1.6]$ m/s	Range of available speeds from which to sample
	$v_z \in [-0.12, 0.12]$ m/s	Range of available depth rates from which to sample
	$r \in [-0.09, 0.09]$ rad/s	Range of available yaw rates from which to sample

possible trajectories. The remaining metrics of Table 4.3 including GP RMSE, GP variance, and mission time in ROIs do not indicate a strong relationship with the values of the BB-IPP parameters.

4.1.1.2 CE-IPP Parameter Tuning

For the CE-IPP method, performance was evaluated by varying the planning horizon T, the elite set percentile threshold ρ, and the number of trajectory samples generated per iteration M. The elite set percentile threshold ρ was chosen for study because it determines the rarity of trajectories that are used to learn subsequent importance samplers, thereby increasing the frequency of generating informative paths.

Table 4.4 shows the resulting performance metrics for various combinations of the CE-IPP parameters. These results exhibit the expected trend that the mean planning time increases as the values of T (longer trajectories) and M (greater number of possible trajectories) are increased. Surprisingly, however, while the mean reward per segment generally increases with longer planning horizons, greater values of M seem to result in a decrease in the mean reward per segment. This could possibly be explained by the fact that larger values of M result in a greater number of trajectories that compose the elite set of samples ϵ_j. In turn, the $(1-\rho)$-percentile of this set $\hat{\gamma}_j$ would be higher if the majority of the samples do not produce large rewards. This is supported by a slight increase in the mean reward per segment and mean mission percent in ROIs as the value of ρ decreases. The remaining metrics of Table 4.4 including GP RMSE and GP variance do not indicate a strong relationship with the

Table 4.3: Mean and standard deviation of the evaluation metrics (Table 4.1) for parameter variations of the BB-IPP algorithm. Fifty simulations were performed for each parameter combination. The bolded values are those that were chosen from the parameter tuning for further simulation and field experiments.

Parameters (T in sec)		**Reward Per Segment**	**Planning Time (sec)**	**GP RMSE**	**GP Variance**	**Percent Mission in ROI**
$T = 105$	$\alpha = 0.9$	1.28 ± 0.11	0.11 ± 0.006	0.024 ± 0.011	0.099 ± 0.082	11.6 ± 10.0
	$\alpha = 0.95$	1.31 ± 0.12	0.11 ± 0.007	0.026 ± 0.023	0.113 ± 0.090	13.1 ± 5.4
	$\alpha = 1.0$	1.35 ± 0.11	0.13 ± 0.016	0.026 ± 0.018	0.114 ± 0.240	13.2 ± 8.7
$T = 140$	$\alpha = 0.9$	1.31 ± 0.13	0.12 ± 0.006	0.025 ± 0.023	0.100 ± 0.088	12.5 ± 6.1
	$\alpha = 0.95$	**1.37 ± 0.09**	**0.12 ± 0.007**	**0.024 ± 0.013**	**0.095 ± 0.08**	**13.2 ± 8.9**
	$\alpha = 1.0$	1.40 ± 0.15	0.22 ± 0.064	0.026 ± 0.023	0.116 ± 0.093	12.0 ± 3.7
$T = 175$	$\alpha = 0.9$	1.31 ± 0.09	0.13 ± 0.006	0.025 ± 0.014	0.108 ± 0.161	10.4 ± 5.2
	$\alpha = 0.95$	1.36 ± 0.09	0.14 ± 0.009	0.026 ± 0.018	0.120 ± 0.234	12.4 ± 8.7
	$\alpha = 1.0$	1.41 ± 0.08	0.59 ± 0.295	0.024 ± 0.016	0.101 ± 0.071	11.8 ± 4.2

values of the CE-IPP parameters.

Overall, the parameters for each algorithm were selected to balance planning time, GP model fidelity, and the time spent in ROIs. The resulting parameter values chosen for subsequent simulation analysis and field experiments are given in Table 4.2. A comparison between the two planning methods shows that while both produced the same level of GP model fidelity, the CE-IPP method produces paths with slightly higher rewards and consistently spends a higher percentage of the mission sampling within ROIs. This comes at the expense of greater computation time, as the BB-IPP method is approximately 5 – 10x faster than the CE-IPP method, particularly when evaluating trajectories over a longer planning horizon. Neither method, however, is prohibitively expensive to preclude further examination, especially if the algorithms were to be optimized for runtime performance.

4.1.2 Performance Over the Sampling Mission

This section further analyzes the performance of each planner over the time history of the sampling mission. A baseline planning approach consisting of a pre-programmed undulating lawnmower (LM) path was also simulated to illustrate the advantages of the proposed GPAS method against conventional sensing approaches. In typical operations, a pre-programmed coverage mission such as this is executed without regard to the collected data. It is not until the data is analyzed after the vehicle returns that ROIs may be identified, at which point the vehicle would need to be redeployed to collect additional data in these areas. The adaptive scheme of the GPAS method addresses this

Table 4.4: Mean and standard deviation of the evaluation metrics (Table 4.1) for parameter variations of the CE-IPP algorithm. Fifty simulations were performed for each parameter combination. The bolded values are those that were chosen from the parameter tuning for further simulation experiments.

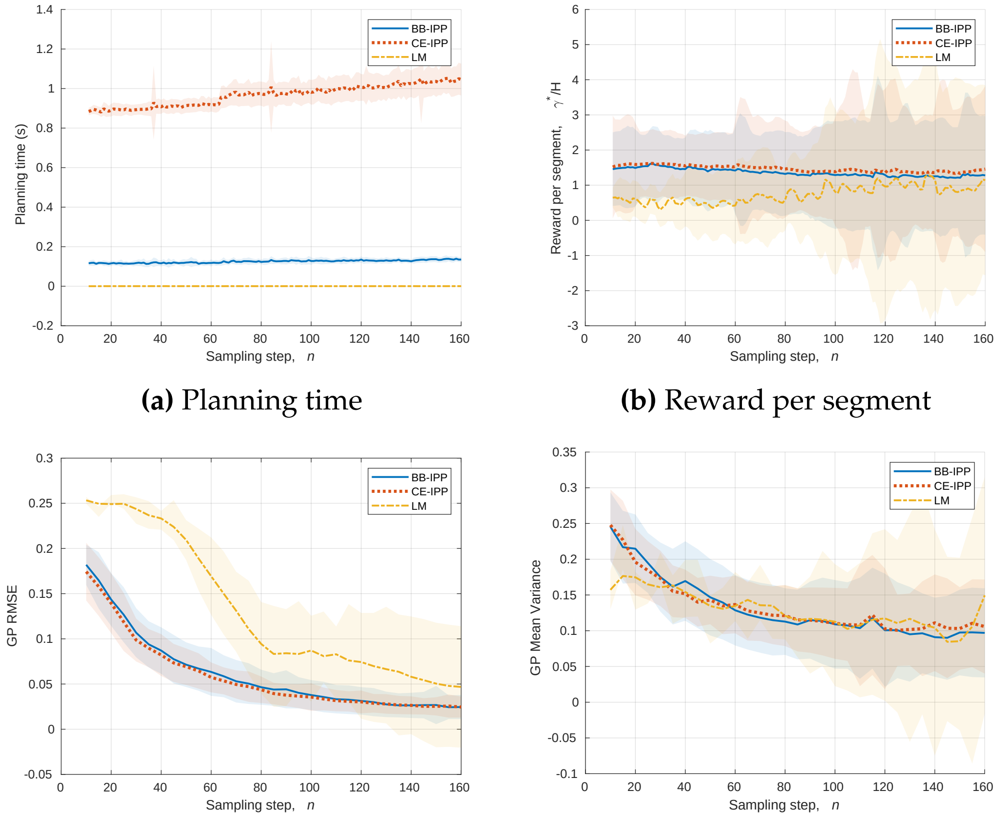
Parameters (T in sec)			**Reward Per Segment**	**Planning Time (sec)**	**GP RMSE**	**GP Variance**	**Percent Mission in ROI**
$T = 105$	$M = 300$	$\rho = 0.05$	1.40 ± 0.09	0.59 ± 0.05	0.029 ± 0.018	0.114 ± 0.010	17.8 ± 10.7
		$\rho = 0.1$	1.40 ± 0.10	0.58 ± 0.05	0.026 ± 0.014	0.184 ± 0.208	18.2 ± 10.0
		$\rho = 0.2$	1.39 ± 0.12	0.58 ± 0.05	0.027 ± 0.014	0.123 ± 0.238	17.7 ± 12.3
	$M = 500$	$\rho = 0.05$	1.40 ± 0.13	1.19 ± 0.13	0.024 ± 0.008	0.105 ± 0.100	20.5 ± 11.3
		$\rho = 0.1$	1.34 ± 0.08	1.18 ± 0.12	0.036 ± 0.021	0.110 ± 0.092	19.6 ± 12.1
		$\rho = 0.2$	1.33 ± 0.14	1.18 ± 0.12	0.022 ± 0.007	0.111 ± 0.211	18.3 ± 8.8
$\boldsymbol{T = 140}$	$\boldsymbol{M = 300}$	$\rho = 0.05$	1.41 ± 0.12	0.96 ± 0.05	0.026 ± 0.015	0.108 ± 0.120	20.88 ± 8.7
		$\boldsymbol{\rho = 0.1}$	$\mathbf{1.46 \pm 0.09}$	$\mathbf{0.96 \pm 0.05}$	$\mathbf{0.025 \pm 0.012}$	$\mathbf{0.106 \pm 0.126}$	$\mathbf{20.9 \pm 11.0}$
		$\rho = 0.2$	1.40 ± 0.21	0.96 ± 0.05	0.032 ± 0.023	0.115 ± 0.242	22.3 ± 14.8
	$M = 500$	$\rho = 0.05$	1.35 ± 0.11	1.95 ± 0.12	0.022 ± 0.006	0.093 ± 0.070	19.4 ± 8.9
		$\rho = 0.1$	1.40 ± 0.13	1.94 ± 0.13	0.022 ± 0.008	0.093 ± 0.067	18.5 ± 7.4
		$\rho = 0.2$	1.32 ± 0.12	1.95 ± 0.12	0.022 ± 0.006	0.098 ± 0.090	17.8 ± 12.1
$T = 175$	$M = 300$	$\rho = 0.05$	1.44 ± 0.09	1.21 ± 0.06	0.022 ± 0.008	0.105 ± 0.088	21.7 ± 12.7
		$\rho = 0.1$	1.44 ± 0.10	1.22 ± 0.06	0.023 ± 0.011	0.104 ± 0.083	20.7 ± 11.0
		$\rho = 0.2$	1.47 ± 0.11	1.22 ± 0.06	0.023 ± 0.014	0.101 ± 0.077	19.1 ± 12.5
	$M = 500$	$\rho = 0.05$	1.40 ± 0.12	2.39 ± 0.12	0.021 ± 0.008	0.101 ± 0.083	19.2 ± 9.2
		$\rho = 0.1$	1.38 ± 0.09	2.40 ± 0.12	0.028 ± 0.022	0.133 ± 0.092	18.7 ± 8.5
		$\rho = 0.2$	1.32 ± 0.12	2.40 ± 0.12	0.022 ± 0.011	0.103 ± 0.088	18.9 ± 9.7

problem by using the acquisition function of Eq. (3.8) to balance the trade-off between data collection in known ROIs (through GP model mean predictions) and exploration of unknown and/or uncertain areas of the environment (through GP model variance predictions).

4.1.2.1 Quantitative Performance

Fig. 4.2 shows the performance of each path planning algorithm as a function of the sampling step, averaged over a total of 150 simulations. While the planning times for each GPAS method are longer than the baseline LM path, it is evident that the trajectories planned by the GPAS method consistently produce higher rewards throughout the mission. Although not shown in Fig. 4.2, this is emphasized by the percentage of the total mission time spent in ROIs: 13.1% of the mission for the BB-IPP method, 20.9% of the mission for the CE-IPP method, and 2% of the mission for the baseline LM path, even though the volume of the ROIs account for only 1.5% of the total sampling area volume. Both the mean reward per segment and the mission percentage in ROIs indicate that the trajectories produced by GPAS significantly outperform the baseline LM path in collecting valuable measurements. In particular, the CE-IPP method produces slightly better rewards than the BB-IPP method and is able to allocate the most sampling time to the ROIs.

While these results indicate that the GPAS method is superior in collecting valuable measurements in ROIs, it is important to ensure that prediction over the entire environmental field is not sacrificed. Fig. 4.2c and 4.2d show the GP RMSE and GP variance as a function of the sampling step, respectively.



(a) Planning time

(b) Reward per segment

(c) GP RMSE

(d) GP variance

Figure 4.2: Metrics used to evaluate the performance of each path planning method over a set of 150 simulations (LM is the baseline undulating lawnmower path). Plots begin at the 10th sampling step so as to let the GP train with an initial set of measurements. The lines are the mean values and the shaded regions capture values within one standard deviation around the mean.

While the GP variance is similar for all planning methods, the lower RMSE values of the GPAS method compared to the baseline LM path show that the adaptive trajectories produce more accurate surrogate models over the course of the mission. This is intuitive, as the adaptive trajectories initially choose to explore areas of the environment with high uncertainty. As the mission progresses, a more accurate GP model means that additional effort can be focused on further improving the model in both ROIs (exploitation)

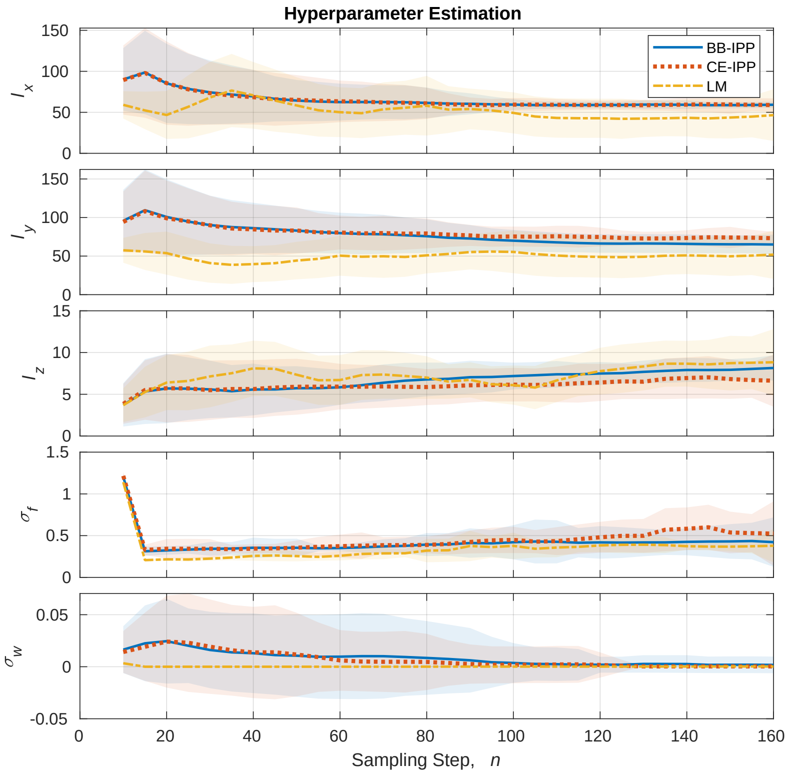


Figure 4.3: Hyperparameter estimates $\boldsymbol{\theta}^* = [l_x, l_y, l_z, \sigma_f, \sigma_w]^T$ for each path planning method over a set of 150 simulations (LM is the baseline undulating lawnmower path). The lines are the mean values and the shaded regions capture values within one standard deviation around the mean.

and uncertain areas (exploration).

Additionally, an ablation study was performed on the GP model to determine its ability to predict ROIs in the absence of direct measurements within these regions. For each simulation, measurements collected within ROIs were removed from the full set of collected measurements and the GP model was retrained on the subset of measurements. This reduced GP model was then used to predict the values of the removed ROI points as well as the values of the full measurement set, resulting in average RMSE values of 0.078 and

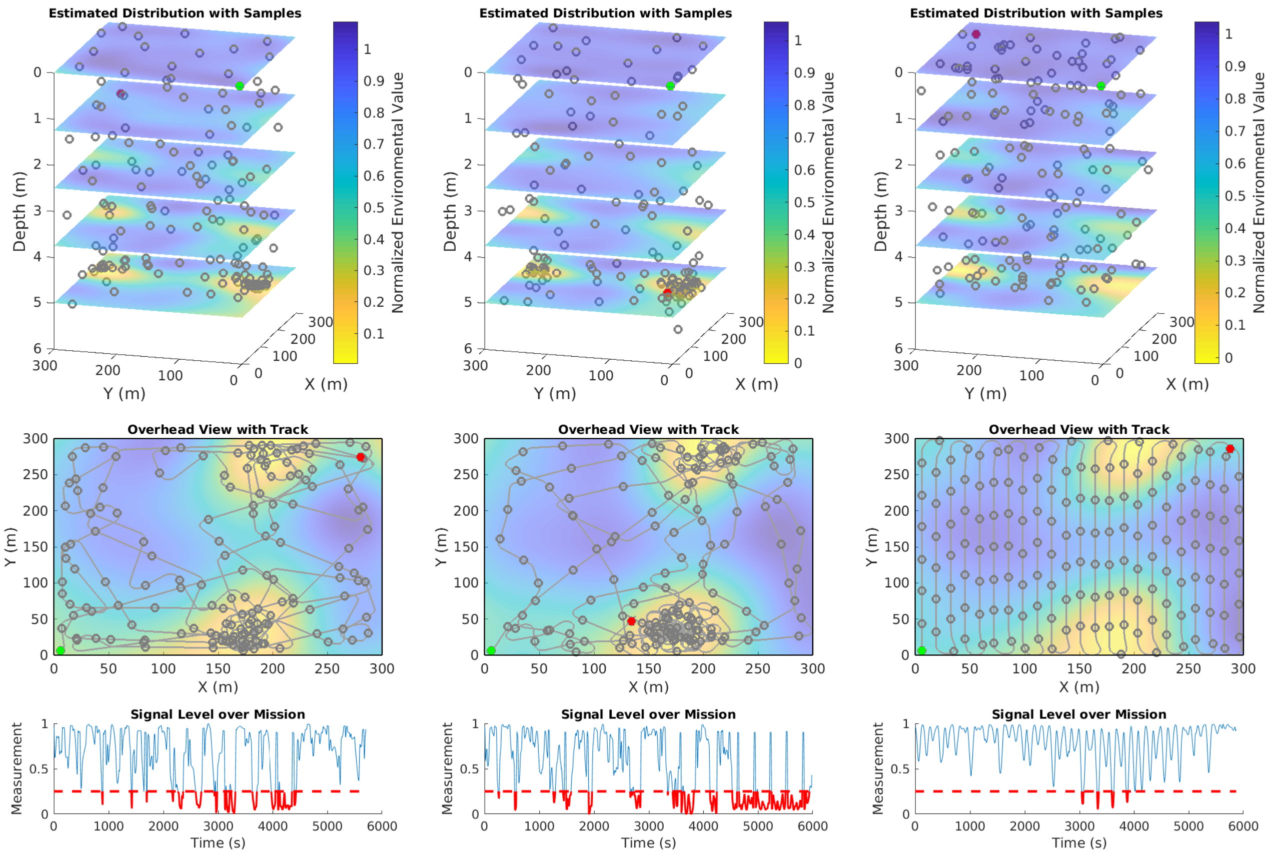
0.055, respectively. These RMSE values are only slightly higher than those observed from the GP model trained on the full measurement set. Further, an average of 83% of the removed measurements were still predicted below the ROI threshold, indicating that the GP model is capable of modeling and predicting these regions even in the absence of direct measurements.

Fig. 4.3 captures the hyperparameter estimates for each path planning method over the same set of simulations. All strategies produced relatively similar hyperparameter estimates that began to stabilize towards the end of the sampling mission. To compare the quality of these hyperparameter estimates, however, the RMSE was calculated using new GP models that were retrained with the final, averaged hyperparameters estimated by each path planning method. This resulted in average RMSE values of 0.038 for BB-IPP, 0.039 for CE-IPP, and 0.047 for the baseline LM method. These results are fairly consistent with the trends seen in Fig. 4.2c and provide further evidence that the GPAS method tends to produce models with overall lower RMSE values.

4.1.2.2 Example Trajectories

Fig. 4.4 shows the resultant trajectories and sampling locations of candidate simulations for each path planning method. These trajectories and sampling locations are overlaid on the estimated GP model of each simulation using the hyperparameter estimates at the end of the mission. Additionally, a time-series plot of the measurement signal is provided for each trajectory with the red portion signifying time spent within an ROI. Even though the trajectories planned by the GPAS method did not provide as complete area coverage as the baseline LM path, they exhibit a good exploration-exploitation trade-off by initially exploring the environment to learn a model of the underlying scalar field, and subsequently focusing samples within the ROIs.

The results shown in Fig. 4.4 also indicate that the GPAS method is capable of detecting and localizing multiple ROIs within a scalar field. Even though the signal level plots show that the first ROI is discovered early in the mission, the GPAS method continues to explore the rest of the field, as the variances around unexplored regions were still high. After the second ROI is discovered by both GPAS planning methods, the sampling is then focused alternatively between the two ROIs, a behavior that results from the GP-UCB criteria to minimize cumulative regret given the number of samples remaining in the mission, the current knowledge / uncertainty of the environment from collected data, and a limited planning horizon. As data is collected in one ROI, the uncertainty of that ROI decreases, and the utility of other areas of the environment (including other ROIs) increases as a result. These results show that the GPAS method is not prone to local minima even though the measurement signal level for both ROIs is comparable. In contrast, since the baseline LM path is preplanned with equal sampling step size, the samples collected along the trajectories are fixed. Furthermore, depending on the vehicle's starting position and the location of the ROI within a given environment, the detection of a ROI is random at best.



(a) BB-IPP

(b) CE-IPP

(c) Undulating lawnmower path

Figure 4.4: Simulation results for three different planning surveys. The top images shows the sampling points in 3D while the middle images show a bird's-eye view of the vehicle trajectory with the estimated distribution at the seafloor. The time series plot on the bottom shows the normalized measurement value over the simulation where measurements within an ROI are highlighted in red.

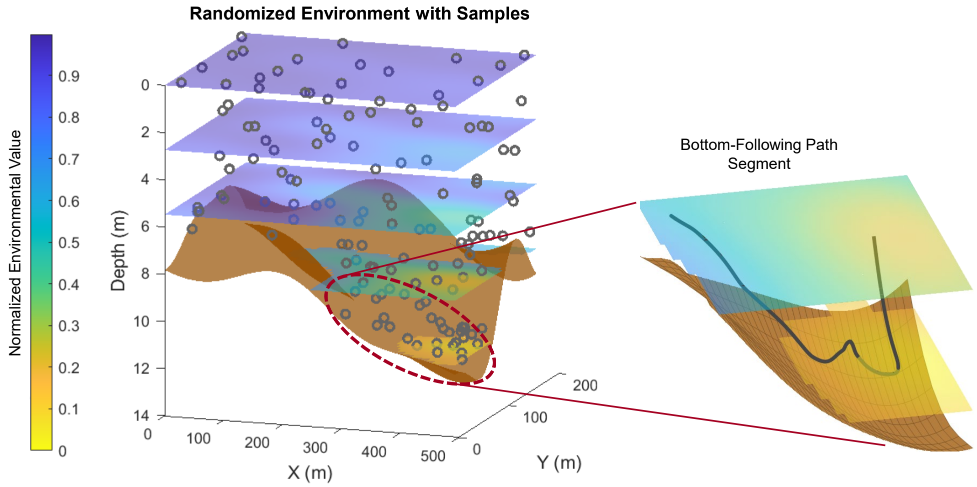


Figure 4.5: An example of a randomly generated environmental distribution and the resulting sampling points. Bathymetry is shown in brown where the right figure illustrates a portion of the trajectory exhibiting bottom-following at a safe minimum height.

4.1.3 Performance on Randomized Environmental Fields

While the simulation results explored so far highlight the advantages of the GPAS algorithm on a candidate test environment, it is important to study whether the results apply to any realistic sampling environment. This section gathers additional statistical evidence of the performance of the BB-IPP, CE-IPP, and baseline LM approaches on a set of 500 randomized, auto-generated sampling environments. The length, width, and depth of each environment was varied to create a range of sampling volumes between approximately $5 \times 10^4\ \mathrm{m}^3$ and $3.5 \times 10^6\ \mathrm{m}^3$. The static field distribution was randomly generated in the horizontal plane and, similar to Fig. 4.1, linearly scaled based on the depth.

In addition to randomly generating the environmental field, bathymetry was also generated for added realism. Consideration of local bathymetry is particularly important for operations in the Chesapeake Bay, as it is typical for hypoxic regions to form close to the seafloor. Bathymetry-aware planning is thus necessary to reduce the risk of vehicle loss when operating in environments characterized by large variations in bathymetry, significant bottom vegetation, and muddy soil. An example of a randomly generated sampling environment is shown in Fig. 4.5.

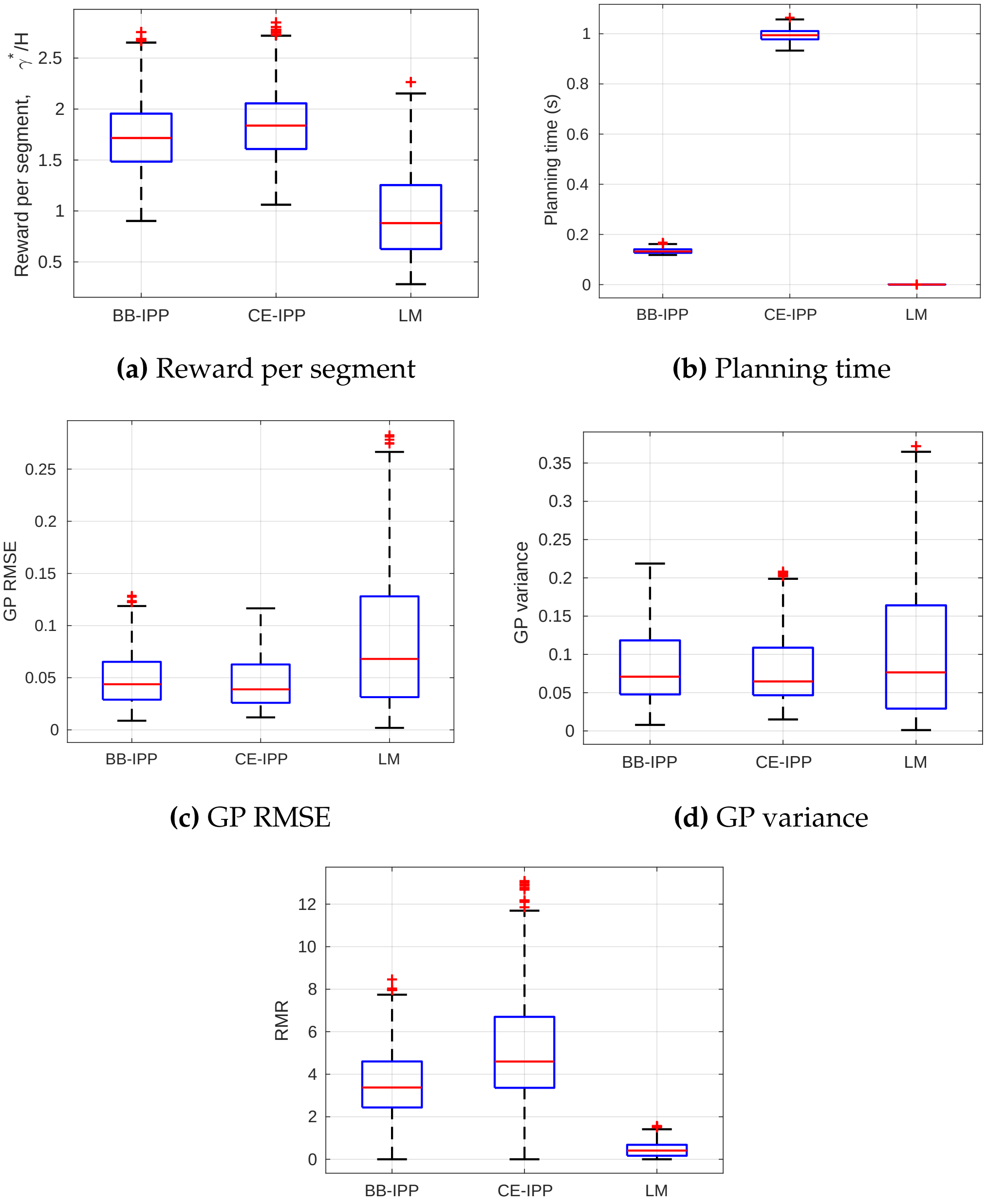
Slight modifications to the test conditions and algorithms were implemented to accommodate the new environments. The overall mission time for each environment was scaled according to the relative volume of each environment. Similarly, the baseline LM path was precomputed to scale according to the relative volume of each environment, not intersect with the seafloor, and cover the entire sampling volume within the mission time. For the adaptive planners, bathymetry was modeled throughout the sampling mission by incorporating measurements from simulated altimeter and pressure sensors. This bathymetry model was used to predict the vehicle's height from the seafloor along each trajectory considered during path planning. If any trajectory violated a safe minimum height from the seafloor, the depth rate v_z was saturated such that the vehicle never violated the safe height threshold. Any trajectory that violates the bathymetry constraints after saturating v_z is removed from consideration. In this way, the resulting motion primitives remain dynamically feasible while safely navigating variable bathymetry profiles as shown in Fig. 4.5. In addition to considering bathymetry during path planning, the waypoint following autopilot was extended to always keep the current state of the vehicle above the safe minimum height while transiting between waypoints.

Evaluation metrics for each planning method are shown in the form of box plots in Fig. 4.6, including a generalized ROI mission ratio (RMR) to account for variation between the randomized environments:

$$RMR = \frac{(t_{ROI}/t_f)}{(V_{ROI}/V)}, \tag{4.1}$$

where t_{ROI} is the time spent within ROIs, V_{ROI} is the volume of the ROIs, and V is the total volume of the test environment. This metric normalizes the percentage of time spent within ROIs based on the actual volume of the ROIs within different environments.

Overall, the performance observed on the randomized environments is consistent with that of Section 4.1.2, where the BB-IPP and CE-IPP algorithms outperform the baseline LM approach on all metrics except planning times (as expected). In particular, the RMR values illustrate that the GPAS algorithm generalizes to a variety of realistic, constrained sampling environments and is able to dedicate more sampling time within ROIs (approximately 3 – 5x for BB-IPP and 4 – 7x for CE-IPP) when compared to the baseline LM approach. Further, the GPAS method produced lower GP RMSE and variance values than the baseline LM coverage pattern, indicating that the adaptive sampling does not compromise the fidelity of the final GP model. The minimal spread of the planning times also indicates that the planning algorithms are agnostic to the environment being sampled and can repeatably produce new sampling points.



(a) Reward per segment

(b) Planning time

(c) GP RMSE

(d) GP variance

(e) ROI mission ratio (RMR)

Figure 4.6: Performance metric box plots for each path planning method over a simulation set of 500 randomly generated sampling environments. The GP RMSE and variance results were based on a GP model trained at the end of each mission.

4.2 Field Experiments

As a demonstration of the algorithm, two field experiments were carried out in the Severn River, Maryland using the JHU Oceanserver Iver3 AUV of Fig. 3.1. The goal was to use the GPAS method on an operational platform to fully explore the environment workspace while also prioritizing the collection of measurements within ROIs. The first experiment prescribed the AUV to take simulated measurements from the virtual environmental field of Fig. 4.1. This virtual field provides a perfect ground truth from which to evaluate the GP modeling. Additionally, this experiment is used to determine if the results from the simulation studies translate to an operational platform, thus giving confidence to the statistical results of the Monte Carlo testing in Section 4.1.3. The second experiment then used the GPAS algorithm to collect in situ dissolved oxyen measurements for the purpose of discovering and modeling hypoxic zones. The locations of the two field experiments are shown in Fig. 4.7.

4.2.1 Test Platform

The AUV is equipped with two on-board computers for autonomous control as well as various sensors for vehicle localization and navigation. When operating on the surface, the AUV primarily uses GPS and a magnetic compass to obtain its pose estimates. The AUV is also equipped with an inertial measurement unit (IMU), a Doppler velocity log (DVL), an altimeter, and a pressure sensor such that it can perform dead reckoning navigation when submerged. Table 4.5 lists the noise characteristics of these sensors. After achieving DVL

Table 4.5: JHU Iver3 AUV measurement sources and resolutions.

State	Source	Update Rate	Measurement Resolution
Roll, Pitch, Heading	OceanServer	1 Hz	$<$ 0.5 deg Heading RMSE when level; 1 deg Heading RMSE when $< \pm 30$ deg tilt
Z Trans	OceanServer	1 Hz	0.1 meter Std Dev
Translational Velocity	600kHz RDI Explorer DVL	4 Hz	0.01 m/s Std Dev

bottom lock, experimental testing [44] found that the submerged positional accuracy of the JHU Iver3 is approximately 0.5% distance traveled in nominal, fixed-depth conditions. This error has the potential to increase to approximately 1% distance traveled when operating in environments with high currents or when executing many course and/or depth changes.

In addition, the vehicle was outfitted with a JFE Advantech RINKO-FT dissolved oxygen sonde for measuring the environmental features of interest. Communication with the AUV is achieved through Wi-Fi when it is on the surface and through acoustic communication when it is submerged. Due to the limited bandwidth available through acoustic communication, only vehicle state updates and safety commands (e.g., abort commands) are sent between the operator and the vehicle. Thus, all processing regarding the path planning and GP modeling is performed by the vehicle's on-board computers.

The two on-board computers have separate functionality in a front-seat / back-seat configuration. The front-seat computer acts as the autopilot of the AUV. This computer processes sensor data to estimate the vehicle's state and accepts commands in the form of desired heading, desired speed, and desired

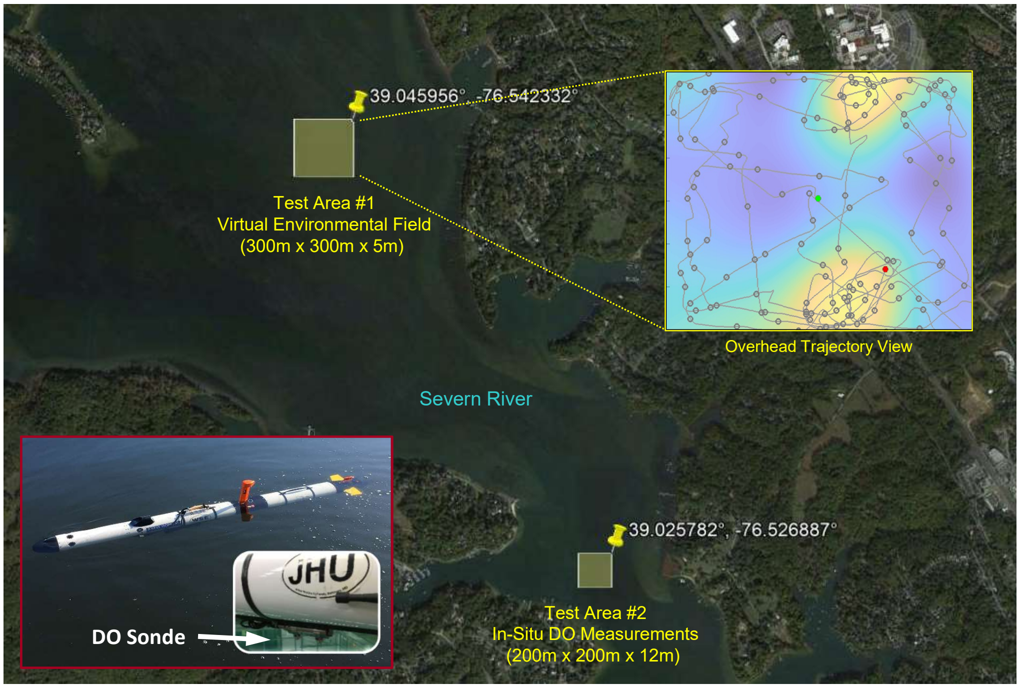


Figure 4.7: AUV field experiments were conducted in the Severn River, Maryland. The northernmost test area, referred to as TA1, ran the GPAS algorithm using virtual measurements from a simulated environmental field with multiple ROIs. The southernmost test area, referred to as TA2, ran the GPAS algorithm using in situ dissolved oxygen measurements.

depth. The back-seat computer hosts the GPAS and path planning algorithms. This computer receives vehicle state information from the front-seat and provides the navigation commands for the current sampling point. Additionally, the back-seat computer contains software to provide safety oversight including geofencing, abort overrides, safety thresholds, and heartbeat monitoring of all software components. The open-source Robot Operating System (ROS) [107] was used as the framework for managing information between each software component on both on-board computers.

4.2.2 Implementation Details

As seen in the results of the simulation studies, there exists a trade-off between performance and computational time between the two planning algorithms: the CE-IPP algorithm slightly outperforms in collecting higher rewards and spending more time in ROIs, while the BB-IPP algorithm requires less computational expense. This trade-off is important because the AUV on-board computers are significantly less powerful than standard laptop or desktop computers. Thus, due to the faster planning times and the relatively constrained sampling areas, the BB-IPP algorithm was chosen as the path planning subroutine implemented on the AUV for field experiments. The advantages of the more computationally expensive CE-IPP algorithm become more pronounced when performing sampling missions in large areas. Slightly longer processing times are acceptable on larger grids because the sampling waypoints can be spaced further apart. Future work in optimizing the implementation of these algorithms could overcome this limitation. Further, sampling platforms with increased computational capabilities (e.g., ASVs) could accept the more expensive CE-IPP algorithm.

Additionally, in a similar manner to Section 4.1.3, the planning algorithm was required to maintain a minimum height of 1.5 meters from the seafloor as a conservative safety condition to prevent vehicle loss, either through entanglement in bottom vegetation or getting stuck in loose mud. Enforcing this condition was done by continuously estimating the local bathymetry using the vehicle's onboard altimeter (to measure height from bottom) and pressure sensor (to measure depth). This safety condition, however, should not prohibit detection of ROIs, as technical reports of the testing area [87, 46] indicate that there is not significant variation in dissolved oxygen measurements within 2 meters of the seafloor (typically < 1 mg/L). Further, the results of the ablation study in Section 4.1.2 indicate that ROIs can still be accurately predicted by the resulting GP model using nearby measurements. As additional emergency recovery mechanisms are added, this safety condition could be relaxed to minimize height from the seafloor.

4.2.3 Experiment #1: Virtual Environmental Field

The first experiment was designed to determine if the results obtained in the simulation studies translated to an operational environment. To this end, the experiment utilized the virtual environmental field of Fig. 4.1 such that the GP model on-board the vehicle was trained via virtual measurements from this distribution based on the AUV's current position. This experiment was performed in the northernmost test area (TA1) of Fig. 4.7. This area was chosen because it is large enough to accommodate the virtual environmental field and consisted of fairly constant bathymetry that would not interfere with sample selection. Algorithm parameters were identical to those in Table 4.2 with the exception that $t_f = 5200$ sec and $\delta_t = 40$ sec. This produces trajectories of $N = 130$ sampling points and allows for additional time required for GPS surfacing events.

The results of the first experiment performed in TA1 are shown in Fig. 4.8 while the performance metrics are summarized in Table 4.6. The plots show that the resulting sampling points and vehicle trajectory are very similar

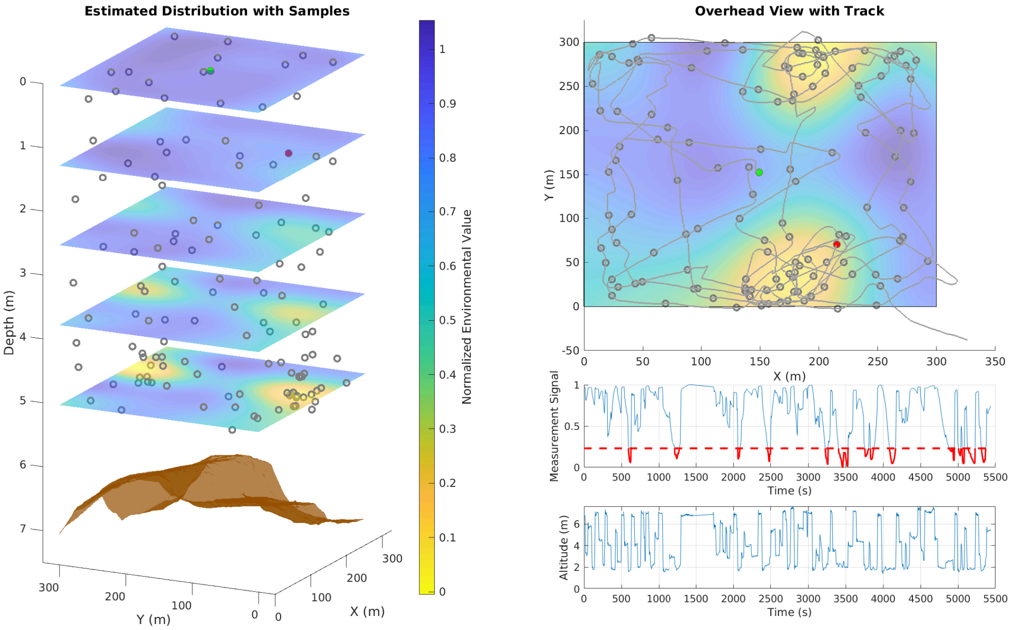


Figure 4.8: Results for the field experiment performed in TA1 using the virtual environmental distribution of Section 4.1. The left image shows the estimated GP model with sampling points in 3D. The measured bathymetry of the test area is shown in brown. The top right plot shows a bird's-eye view of the AUV trajectory with the estimated distribution at the seafloor. The time series plots in the bottom right show the normalized virtual measurements, where measurements within an ROI are highlighted in red, and the vehicle altitude.

to those observed in the simulation studies. Additionally, the performance metrics of Table 4.6 are consistent with the simulation results observed in Fig. 4.2 with the exception of longer planning times due to less processing power on the AUV on-board computers. Both of these facts demonstrate the practicality of the proposed method in a real world application while also giving further confidence to the statistical results from the randomized environment testing of Section 4.1.3.

One can observe in the TA1 results of Fig. 4.8 that the AUV unexpectedly navigated outside of the safety geofence due to a series of failed checksum

Table 4.6: Summary of performance metrics observed for each field experiment. Values for the GP RMSE and GP Variance are measured at the end of the sampling mission. GP RMSE was not calculated for the experiment in TA2 due to the lack of a ground truth dissolved oxygen source.

Experiment	Planning Time (sec)	GP RMSE	GP Variance	Percent Mission in ROI
TA1	0.67 ± 0.21	0.056	0.098	17%
TA2	0.63 ± 0.20	-	0.42 mg/L	48%

messages in the commanded inputs. This event occurred approximately 1400 sec into the mission and is evident by the break in the AUV's 2D trajectory outside the bottom right corner of the sampling area. At this time, safety mechanisms caused the AUV to travel back to the center of the mission area. The AUV resumed operations once back at the center of the sampling area approximately 1700 sec into the mission and successfully completed the remainder of the mission. Overall, this caused a 300 sec delay from the originally intended 5200 sec total mission time.

4.2.4 Experiment #2: In Situ Measurements From Dissolved Oxygen Field

The second experiment was performed in the southernmost test area (TA2) of Fig. 4.7. This experiment was designed to demonstrate the GPAS algorithm in collecting samples based on an environmental field estimated from in situ dissolved oxygen measurements. As described in Section 4.1, dissolved oxygen values in the Chesapeake Bay typically range between 0 – 9 mg/L. In order to evaluate sampling locations using the GP-UCB, measurements were normalized based on this range to be between 0 – 1. The dimensions of TA2 are 200 meters × 200 meters with depth ranges between 6 meters

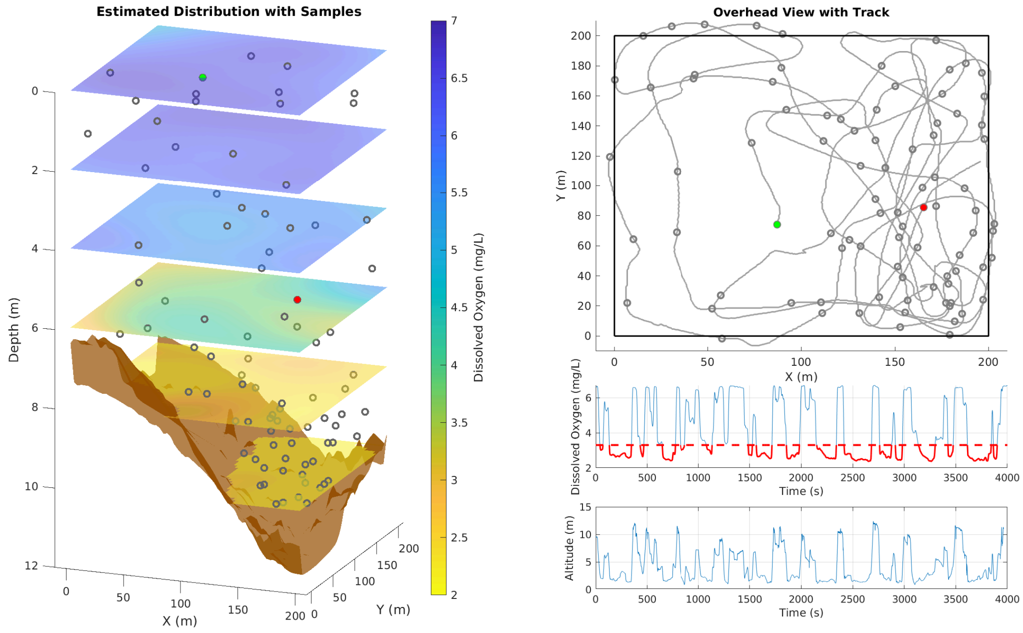


Figure 4.9: Results for the field experiment performed in TA2 using in situ dissolved oxygen measurements. The left image shows the estimated GP model with sampling points in 3D. The measured bathymetry of the test area is shown in brown. The top right plot shows a bird's-eye view of the AUV trajectory. The time series plots in the bottom right show the DO measurements, where measurements below the normalized ROI range (22nd percentile) are highlighted in red, and the vehicle altitude.

on the western edge and 12 meters on the eastern edge. This test site lies within a channel of the Severn River that is characterized by large changes in bathymetry and greater than average depths. These characteristics are of interest for measuring dissolved oxygen, as pockets of significant depth at the center of the channel have a tendency to produce hypoxic regions [90]. Algorithm parameters were identical to those in Table 4.2 with the exception that $t_f = 4000$ sec, $\delta_t = 40$ sec, and $N = 85$ to accommodate the smaller sampling volume.

The results obtained for the field experiment in TA2 are shown in Fig. 4.9 while the performance metrics are summarized in Table 4.6 (with the exception that GP RMSE could not be calculated in the absence of the ground truth dissolved oxygen field). The environmental distribution shown by the GP model in Fig. 4.9 is an estimate based on the collected dissolved oxygen measurements, which range between approximately 2.3 mg/L at the bottom of the channel along the eastern edge and 7 mg/L at the surface. The first takeaway from this experiment is that hypoxic conditions (< 2 mg/L) were not present at TA2 during the data collection event. This is primarily due to the fact that the field experiment was performed in early October, while hypoxic zones are typically most prevalent in July and August. The observed dissolved oxygen levels, however, were still well below the 3 mg/L threshold that is required to support most marine life including deep-water fish, crabs, and oysters [132]. Further, the observation that these measurements occurred in early October means that the region surrounding TA2 would be highly susceptible to hypoxia during peak summer months.

In the absence of hypoxic zones, an alternative evaluation of the GPAS effectiveness is to consider the samples collected in relation to the *available* measurement range at the test site. We continue to use the previous definition of an ROI as values below a normalized threshold of 0.22 (i.e., the 22nd percentile of the available measurement range), but now in relation to the available measurement range of 2.3 – 7 mg/L. According to this evaluation, the AUV spent 1946 sec within ROIs, accounting for approximately 48% of the total mission time. This translates to an *RMR* value of 2.4 when considering the volume of the ROI based on the collected data. In other words, the GPAS algorithm performed as expected with the information that was available during collection. This is evidenced in Fig. 4.9 by the concentration of samples collected along the eastern edge of the sampling area, especially towards the end of the sampling mission.

4.3 Summary of Findings

The results presented in this chapter indicate that the proposed GPAS methodology achieves the desired goal of producing environmental datasets that focus on regions of interest, while at the same time not sacrificing the final model fidelity of the full workspace. In fact, the Monte Carlo simulation results on randomized environments even show that the final GP models trained from GPAS trajectories are more accurate than those trained from conventional coverage trajectories.

The statistical performance gleaned from the simulation analysis was then supported by the results seen during AUV field tests. When operating on one of the same environmental fields used in simulation (in a virtual sense), the real AUV produced nearly identical results in its performance metrics. This fact gives added confidence that the performance seen in simulation is a good indicator of the performance that can be expected during deployments in similar environments. The final field test operating on in situ dissolved oxygen measurements serves as a demonstration into one such environment. Given enough trust in the system through additional simulation validation and field testing, the desire is that deployments such as the field test in TA2 could become a regular part of the strategy in monitoring the health of estuaries such as the Chesapeake Bay.

Part II

Good Seamanship Path Planning and Performance Evaluation for Autonomous Surface Vessels

Chapter 5

Background on Autonomous Surface Vessel Navigation

The focus of this thesis in Part II now shifts to another problem within marine robotics: improved navigation strategies for autonomous surface vessels (ASVs). While ASV navigation shares many similarities with nonholonomic path planning in other fields, it remains an open problem due to an array of domain-specific challenges and the many competing requirements of these systems. For example, ASVs must not only satisfy mission objectives and perform traditional collision avoidance, but they must also act in accordance with more abstract good seamanship[1] principles that human ship captains would exhibit. This includes predictable maneuvering, taking early and obvious actions, and obeying navigation protocols according to the International Regulations for Prevention of Collisions at Sea [41], referred to as COLREGS.

[1] As stated in Chapter 1, the term "seamanship" is used throughout this thesis because it is explicitly referenced in international maritime protocols [41] and contains intrinsic concepts that are understood by the practicing community. In an effort to avoid gender-biased language, however, terms such as seamanship should begin to be phased out and replaced with neutral terms (e.g., ship handling) in the future.

5.1 Review of COLREGS

The most recent edition of COLREGS [41] was initially published in 1972 and has been adopted as common practice since 1977. The full set of protocols consists of 38 rules organized into five different sections. The first two sections address the general applicability of the rules as well as proper ship handling etiquette. The final three sections address proper light and sound signaling, priorities given to different types of vessels, and exemptions given in special circumstances. These final three sections heavily depend on the ability to properly classify the vessel type and interpret its day shapes, lights, or sounds. While a complete ASV navigation suite would be expected to comply with all of the rules, the work in this thesis does not focus on maritime perception and classification (which in itself is a challenging research subject). Thus, only a subset of COLREGS from the first two sections is addressed, leaving extensions to the remaining rules open for future work.

Specifically, this research focuses on power-driven vessels and the COLREGS maneuvering protocols that all vessels must follow to reduce confusion when there is a collision risk. These expectations are covered in rules 2, 7, 8, 13 – 17 and designate vessels as either give-way (one which must yield to the other vessel) or stand-on (one which is expected to maintain course and speed while the other vessel yields).

- **Rule 2 – Responsibility:** The rule of responsibility essentially acts as a catch-all by saying that, regardless of what is dictated by the other COLREGS protocols, it is the responsibility of any vessel to act according to the “ordinary practice of seamen.” This rule of good seamanship is meant to recognize that special circumstances arise on the water and it is the responsibility of all vessels to act appropriately, regardless of which vessel has right-of-way.

- **Rule 7 – Risk of Collision:** This rule dictates that vessels must continuously monitor whether a collision risk exists with another vessel. Factors that influence whether a collision risk exists include the navigation conditions and the size / actions of the target ship.

- **Rule 8 – Action to Avoid Collision:** This rule again includes aspects of good seamanship, as it covers characteristics about proper vessel actions in a collision avoidance scenario. These characteristics include making large, obvious maneuvers that are readily apparent on radar, making early maneuvers to avoid close-quarter situations, making maneuvers that result in passing at a safe distance, and not impeding the safe passage of another vessel.

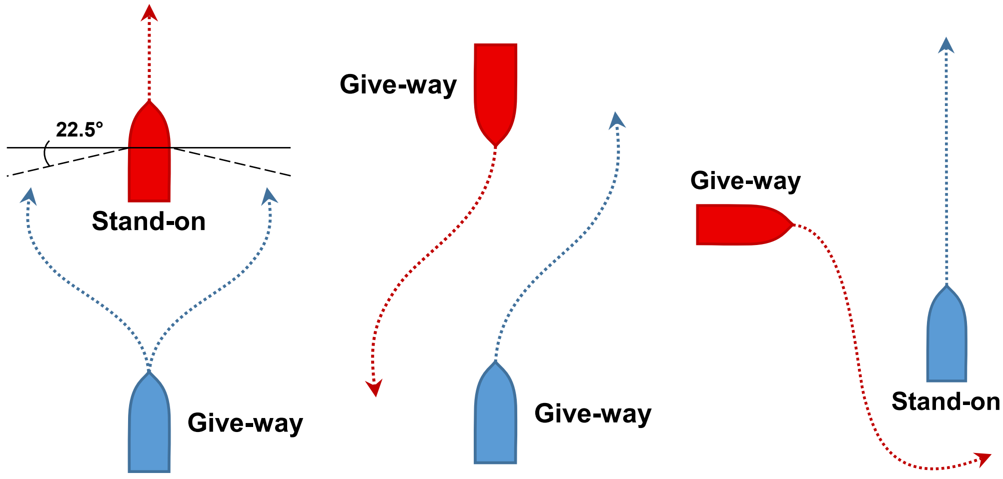
- **Rule 13 – Overtaking:** If a collision risk exists, a vessel is deemed to be overtaking when it approaches another vessel from a direction more than 22.5 deg abaft abeam such that the overtaking vessel would only be able to see the stern lights of the overtaken vessel (Fig. 5.1a). The overtaking vessel is expected to give way whereas the overtaken vessel is expected to stand on. If there is confusion as to whether ownship is overtaking, it should assume that it is.

- **Rule 14 – Head-on:** If a collision risk exists, a head-on encounter is described by vessels that approach each other on approximately reciprocal courses (Fig. 5.1b). The preferred de-escalation strategy in this scenario is for both vessels to give way to starboard such that they pass on each other's port side. However, special circumstances (such as a significantly offset head-on encounter) may deem it appropriate that the vessels pass each other on their starboard side.

- **Rule 15 – Crossing:** If a collision risk exists and the geometry of the encounter does not satisfy the criteria for overtaking or head-on scenarios, the encounter is a crossing scenario (Fig. 5.1c). If the target ship is approaching from ownship's starboard side, then ownship is expected to give way while the target ship is expected to stand on.

- **Rule 16 – Action by Give-Way Vessel**. This rule simply states that the give-way vessel must take early and appropriate action to stay well clear of the other vessel.

- **Rule 17 – Action by Stand-On Vessel** The primary expectation of the stand-on vessel is to maintain course and speed. If, however, the give-way vessel is noncompliant (i.e., does not take appropriate evasive action in a timely manner), the stand-on vessel is permitted and expected to take its own evasive action, so long as it does not turn to port. Turning to port in this instance could result in a head-on collision if the give-way vessel belatedly maneuvers to starboard as it should.



(a) Rule 13 – Overtaking **(b)** Rule 14 – Head-on **(c)** Rule 15 – Crossing

Figure 5.1: Diagrams of COLREGS rules 13 – 15 and the expected actions by each vessel.

5.2 Collision Risk Assessment

The rules presented above rely on the underlying assumption that a collision risk exists between the vessels; otherwise, no action would need to be taken. There have been many studies over the past two decades that assess the collision risk between two oncoming vessels, including general surveys of maritime risk assessment [66, 80]. Traditionally, metrics related to the closest point of approach (CPA) have been used to determine the collision risk of an encounter. The CPA is defined as the location where two moving objects with fixed velocity vectors reach their minimum separation distance. For example, the works of Bukhari et al. [12] and Perera et al. [102] both assess collision risk based on fuzzy inference of CPA-based indices. Meanwhile, Li and Pang [64] applied Dempster-Shafer theory to combine multiple CPA-based indices into an overall collision risk assessment.

Two important metrics derived from CPA are the time until CPA (t_{cpa}) and the distance at CPA (d_{cpa}). Defining the position vectors of each vessel as $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ and their velocity vectors as $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$, these metrics are calculated as follows:

$$t_{cpa} = \frac{(\boldsymbol{p}_2 - \boldsymbol{p}_1) \cdot (\boldsymbol{v}_1 - \boldsymbol{v}_2)}{||\boldsymbol{v}_1 - \boldsymbol{v}_2||^2}, \tag{5.1}$$

$$d_{cpa} = ||(\boldsymbol{p}_1 + t_{cpa}\boldsymbol{v}_1) - (\boldsymbol{p}_2 + t_{cpa}\boldsymbol{v}_2)||. \tag{5.2}$$

In this formulation, a negative value of t_{cpa} indicates that CPA occurred in the past. Collision risk and the applicability of COLREGS can then be determined based on whether t_{cpa} and d_{cpa} are less than the desired safety thresholds, the values of which are largely the preference of the ship captain and could vary based on the situation at hand.

A parallel line of research has focused on performing maritime safety assessment using the concept of ship domain. The traditional definition of ship domain as defined by Goodwin [38] is "the surrounding effective waters which the navigator of a ship wants to keep clear of other ships or fixed objects." With regards to safety assessment, various authors have applied ship domain in a binary fashion (i.e., any object within the vessel's domain is considered a threat to navigational safety) [35, 18], while others have adopted a more continuous risk assessment, known as a fuzzy ship domain, based on the object's degree of penetration into the vessel's domain [124, 134, 104]. He et al. [45] use multiple ship domains that depend on the COLREGS situation to determine collision avoidance maneuvers. An excellent review of different ship domain models and their applications is presented by Szlapczynski [126].

5.3 ASV Navigation Related Work

Until COLREGS are adjusted to explicitly consider ASVs, it is understood that ASVs and humans alike must abide by the current standard. The challenge in developing ASV software to comply with these protocols, however, is that they were originally written for human-operated vessels. The result is that portions of the protocols were intentionally left vague so as to leave room for interpretation and common sense decision-making (from a human perspective). Further, there are many aspects of good seamanship not explicitly written in these protocols, particularly regarding multi-vessel encounters. In these circumstances, some COLREGS protocols may not be applicable because they would produce conflicting expectations for each vessel. A robust ASV system must be capable of COLREGS compliance while remaining flexible enough to not blindly adhere to a rule-based system in more complicated scenarios.

Many of the early approaches to COLREGS-compliant ASV navigation focused on reactive planning methods, meaning that control actions are generated in a local sense from immediate sensor information as opposed to employing motion planning to generate a future trajectory. For ASVs, this typically takes the form of generating a desired velocity vector. Benjamin et al. [5] was one of the first works to introduce behavior-based COLREGS compliance using interval programming (IvP). Within an IvP framework, the COLREGS behavior produces an objective function that encourages COLREGS compliance, which is then combined with objective functions from other behaviors (e.g. waypoint following) and optimized to produce the desired velocity vector. Other work by Naeem et al. [98] offers an alternative based on line-of-sight waypoint guidance that is augmented with manual heading biases to account for different COLREGS encounters. Extensions to velocity obstacles [31] are also popular for ASV navigation due to the ability to handle many dynamic obstacles at reasonable computational expense. Kuwata et al. [58] incorporated COLREGS with velocity obstacles by augmenting the restricted velocity space with regions that violate COLREGS. Similar concepts have also been adopted using reciprocal velocity obstacles [57].

More recent work in COLREGS-compliant ASV navigation has focused primarily on deliberative planning approaches. Eriksen provides a recent collection of works [28, 27, 29] detailing each component in an overall ASV navigation system, including a planning approach that evaluates an exhaustive set of motion primitive-based trajectories. The work by Bergman et al. [6] uses a similar approach within a two-stage planner, where motion primitives are used as the trajectory seed for direct optimal control methods. Modifications to tree-based designs are also described in a COLREGS-modified RRT planner [16] and model predictive control-based approaches [49, 42]. Rather than using a discrete set of motion primitives at each stage of the planning tree construction, other methods incorporate adaptive sampling techniques to scale the length of the primitives based on the estimated future risk [123, 113]. Recent lines of research have also begun the challenging problem of modeling the probabilistic intent of other vessels during planning [129].

A limitation of the approaches listed above, however, is their ability to generalize in complicated multi-vessel scenarios. As discussed above, while

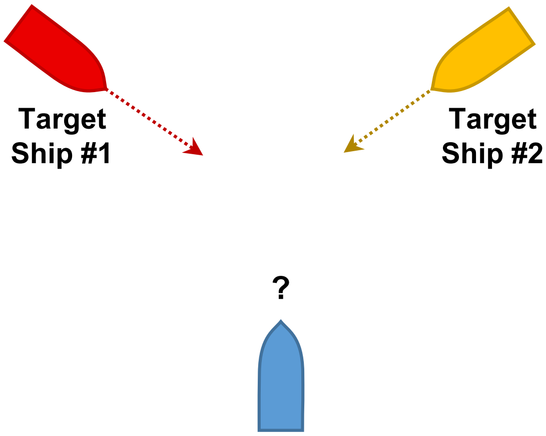


Figure 5.2: Example of a multi-vessel scenario with conflicting COLREGS rules.

COLREGS provide guidance in single-vessel scenarios, the protocols cannot be blindly applied to each vessel individually in multi-vessel encounters. For example, consider the scenario shown in Fig. 5.2. If COLREGS rules were applied to each target ship individually, ownship would be expected to stand on to target ship #1 but also give way to target ship #2. These conflicting expectations mean that COLREGS rules designed for single-vessel scenarios do not generalize easily to multi-vessel scenarios. In reality, the primary rules that remain in multi-vessel situations are those that specify good seamanship practices (e.g., Rules 2, 7, 8). Thus, it is important to consider what exactly constitutes good seamanship such that ASV software can exhibit these principles when faced with complicated avoidance situations. The approach presented in Chapter 6 aims to make progress toward this goal of an ASV navigation system that generalizes to all types of on-water encounters.

Chapter 6

Encoding Good Seamanship Principles Within a Path Planning Framework

This chapter now begins to detail a hybrid planning approach that adheres to COLREGS protocols in single-vessel encounters, but also incorporates a more general risk-based framework for multi-vessel encounters. Once again, the problem statement introduced in Section 1.3 is used to guide the planner development, albeit slightly reformulated to now minimize a cost rather than maximize a reward, i.e.:

$$\boldsymbol{z}^* = \underset{\boldsymbol{z} \in \mathcal{Z}_{con}}{\operatorname{argmin}} \, J(\boldsymbol{z}), \tag{6.1}$$

$$\gamma^* = \min_{\boldsymbol{z} \in \mathcal{Z}_{con}} J(\boldsymbol{z}), \tag{6.2}$$

where then $\pi^* = \varphi(\boldsymbol{z}^*)$. This is a notational choice for convenience that allows the costs to be defined as positive values.

The dynamic surface vessel model and its parameterization for trajectory

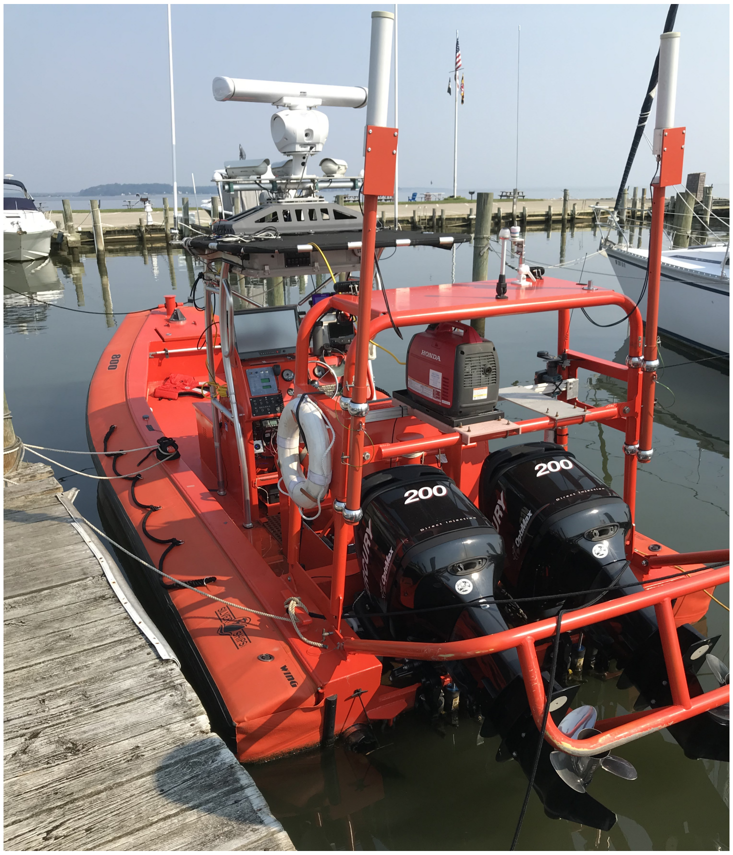


Figure 6.1: Autonomous surface vessel used in field experiments for which the hydrodynamic model was derived.

generation are first provided, followed by the derivation of how good seamanship principles can be quantified to evaluate the quality of proposed paths. Finally, the strategy employed to perform this path planning is described.

6.1 Surface Vessel Dynamic Model

First, we introduce the hydrodynamic surface vessel model of the system under test – a simplified 2-DOF model that captures the unique acceleration and turning characteristics of high-speed surface craft while minimizing the number of required model coefficients. Thus, it is assumed that the vessel operates in a plane where roll, pitch, sway, and heave effects are negligible (at least for the purposes of simulating ASV navigation at mission-length timescales). This model was developed for the 29-ft rigid-hull inflatable boat shown in Fig. 6.1 and would primarily be intended for smaller vessels. The state $\boldsymbol{x} = (p_x, p_y, \psi, u, r) \in \mathbb{X}$ includes the position $\boldsymbol{p} = (p_x, p_y) \in \mathbb{R}^2$, yaw ψ, surge velocity u, and yaw rate r. The control inputs $\boldsymbol{u} = (\delta_T, \delta_R) \in \mathbb{U}$ define the normalized thrust input and normalized yaw rate input, respectively. These control inputs are scaled and constrained in order to produce high-level motion characteristics at steady-state. The normalized thrust input $\delta_T \in [0, 1]$ represents the range between zero and maximum propeller speed, and the normalized yaw rate input $\delta_R \in [-1, 1]$ represents the range between maximal deflections of the vehicle rudder. The equations of motion $\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u})$ are given as the following:

$$\dot{p}_x = u \cos \psi, \tag{6.3a}$$

$$\dot{p}_y = u \sin \psi, \tag{6.3b}$$

$$\dot{\psi} = r, \tag{6.3c}$$

$$\dot{u} = \overline{X}_{uu} u^2 + \overline{X}_u u + \bar{k} \delta_T, \tag{6.3d}$$

$$\dot{r} = \overline{N}_{ur}(u) u r + \overline{N}_r(u) r + \bar{c}(u) u^2 \delta_R, \tag{6.3e}$$

where $(\bar{\cdot})$ indicates that the hydrodynamic coefficients have been made independent of the vehicle's mass and inertia.

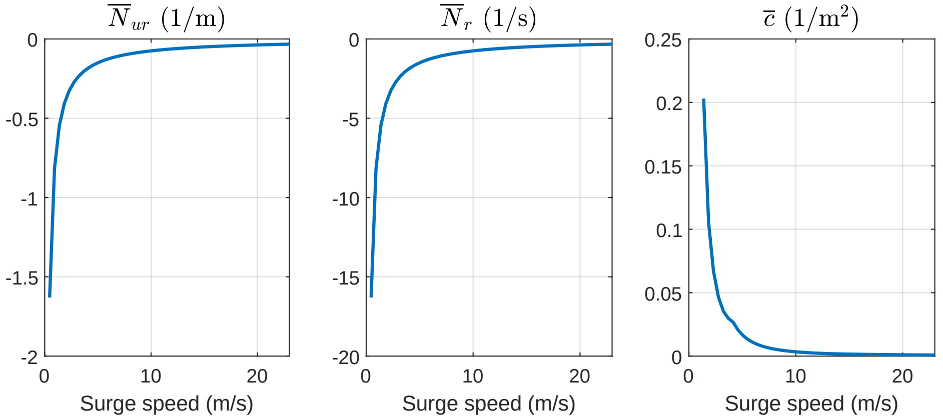


Figure 6.2: Values for the yaw acceleration model coefficients based on vessel surge speed.

Table 6.1: Surface vessel hydrodynamic model coefficients

Coefficient	$\overline{X}_{uu}$	$\overline{X}_u$	$\overline{k}$	$\overline{N}_{ur}(u)$	$\overline{N}_r(u)$	$\overline{c}(u)$
Value	-0.0114	-0.2283	11.29	(Fig. 6.2)	(Fig. 6.2)	(Fig. 6.2)
Units	1/m	1/s	m/s^2	1/m	1/s	1/m^2

Surface craft hydrodynamics are complicated by the fact that the motion characteristics are dependent on whether the vessel is plowing (pushing through the water at slow speeds) or planing (riding on top of the water at high speeds). The equations of motion are able to capture the main effects of each regime by scheduling the yaw acceleration model coefficients on the vessel surge speed. The values of these coefficients as a function of surge speed are shown in Fig. 6.2 and the remaining coefficients are given in Table 6.1.

6.2 Surface Vessel Motion Parameterization

For high-level ASV planning, trajectories are generated using a sequence of motion primitives according to the parameterization introduced in Section 1.3.2. The resulting primitive curves can account for kinematic constraints and be solved in closed form, enabling the planner to consider a large number of future state evolutions.

More formally, each primitive assumes constant velocity and turn rate, and the primitive parameter is given as $\xi = (\delta_u, \delta_\psi)$ corresponding to desired changes in speed and heading, respectively. The full, multi-stage trajectory parameter over the planning horizon is then given as $z = (\xi_1, ..., \xi_n) \in \mathcal{Z}$ for a sequence of n primitives. At a time $t^* = t + \delta_t$, the mapping $\varphi : \mathcal{Z} \to \mathbb{X}$ then takes the following form:

$$u(t^*) = u(t) + \delta_t \left(\min\big(\max(\tfrac{\delta_u}{\delta_t}, a_{min}), a_{max}\big)\right), \tag{6.4a}$$

$$r(t^*) = \min\big(\max(\tfrac{\delta_\psi}{\delta_t}, r_{min}), r_{max}\big), \tag{6.4b}$$

$$\psi(t^*) = \psi(t) + \delta_t r, \tag{6.4c}$$

$$p_x(t^*) = \begin{cases} p_x(t) + \frac{u(t^*)}{r}\big(\sin(\psi(t) + \delta_t r) - \sin\psi(t)\big) & \text{if } r \neq 0, \\ p_x(t) + \delta_t u(t^*)\cos(\psi(t)) & \text{otherwise,} \end{cases} \tag{6.4d}$$

$$p_y(t^*) = \begin{cases} p_y(t) - \frac{u(t^*)}{r}\big(\cos(\psi(t) + \delta_t r) - \cos\psi(t)\big) & \text{if } r \neq 0, \\ p_y(t) + \delta_t u(t^*)\sin(\psi(t)) & \text{otherwise.} \end{cases} \tag{6.4e}$$

where linear acceleration $a \in [a_{min}, a_{max}]$ and turn rate $r \in [r_{min}, r_{max}]$ have been saturated within the system's allowable bounds. This formulation assumes that the transients from potential velocity discontinuities are small when compared to the horizon over which surface vessel interactions occur, particularly when using conservative values for the system's allowable control bounds.

6.3 Quantifying Good Seamanship

Developing ASV planning and evaluation strategies that generalize to complicated multi-vessel scenarios requires that they embody the notions of good seamanship. These practices not only include the COLREGS maneuvering protocols primarily intended for single-vessel encounters, but also more general themes regarding safe and predictable navigation on the water that transcend rules 13 – 17. This section attempts to quantify good seamanship practices into a mathematical framework for ASV planning and evaluation purposes.

An appropriate quantification of good seamanship should capture the essence of COLREGS maneuvering compliance without subjecting the analysis to rule-based conditions that may not be appropriate in multi-vessel encounters, e.g., when there are conflicting give-way and stand-on expectations. In these instances, it is understood that COLREGS are interpreted more as guidelines to good seamanship as opposed to a set of rules that must be strictly followed. As such, this methodology could supplement existing COLREGS decision-making, where specific COLREGS rule classification loses value due to potentially conflicting rules for each target ship. Good seamanship is quantified here based on the combination of two categories: (i) how the ASV reduces the overall collision risk to all vessels involved in an encounter and (ii) whether the ASV takes early, appropriate action. The first category

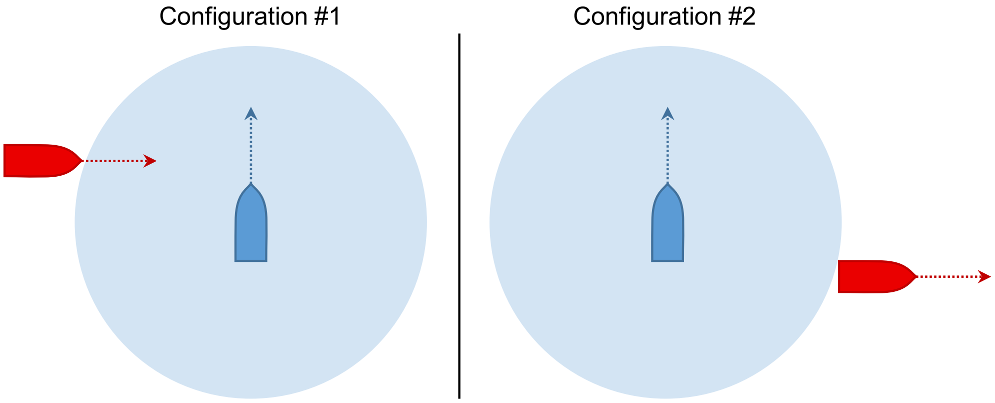


Figure 6.3: Two configurations with the same d_{cpa} value. It is clear, however, that the scenario on the left presents a much higher risk at CPA than the scenario on the right.

is rooted in the collision avoidance requirements of all mariners while the second category captures the essence of responsibility intended by COLREGS.

6.3.1 Collision Risk Quantification

6.3.1.1 Ship Domain

This research makes use of the concept of ship domain (as described in Section 5.2) to quantify the overall collision risk between multiple vessels. An appropriate definition of the ship domain geometry can overcome several of the weaknesses associated with CPA-based risk assessment. This is because CPA is naturally a circular domain representation and does not discriminate between different geometries with the same d_{cpa} value. This idea is illustrated through two different configurations shown in Fig. 6.3. Both configurations have the same d_{cpa} value, however, it is clear that the scenario on the left presents a much higher risk when compared with the scenario on the right. Conversely, ship domain geometry can be described by any polygon, typically

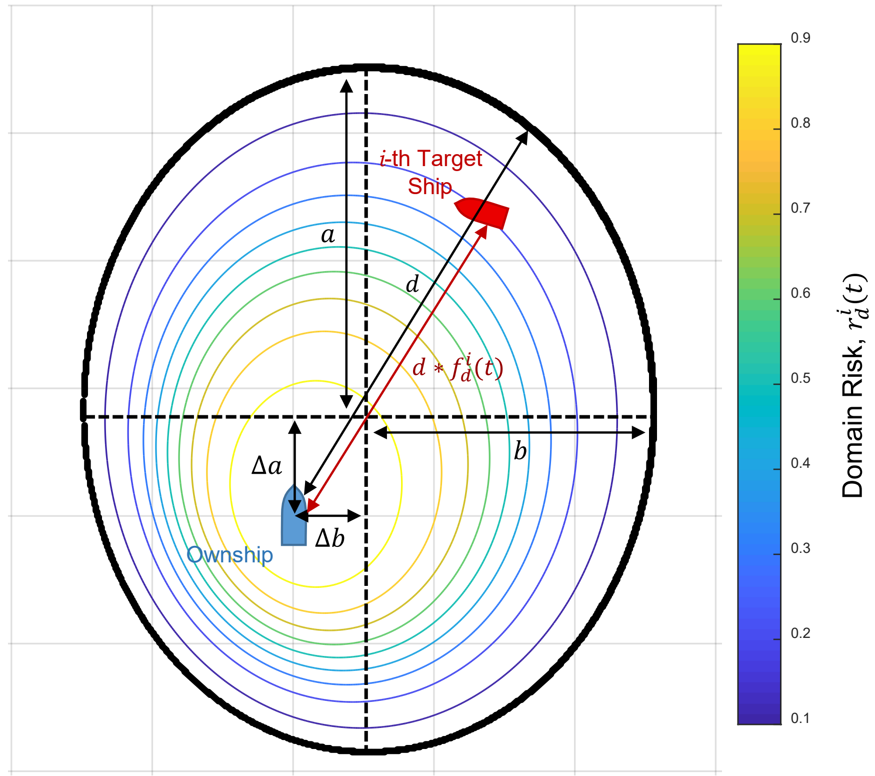


Figure 6.4: Decentralized ellipse ship domain, where d is the distance to the edge of ownship's domain in the direction of the target ship. The fuzzy ship domain, which translates domain violation $f_d(t)$ into a risk value, is shown by the inner contour plots.

one that emphasizes keeping the fore and starboard sectors of the vessel clear, as illustrated in Fig. 6.4.

Existing ship domains proposed in the literature generally define the geometry based on three different methods: analytical [134, 68], empirical from ship movement data [18], and those based on artificial intelligence from expert knowledge [104, 140]. Because ship domain has been used in various applications, different authors adopt slightly varying definitions of ship domain and there is no universally accepted geometry. We utilize ship domain with the following assumptions:

- The ship domain is designed for an open water setting, although it could be modified for use in traffic separation schemes or waters with constricted maneuvering.
- Following Szlapczynski [126], the domains of all vessels in an encounter should be kept clear, i.e., ownship should not violate the domain of any target ship and likewise each target ship should not violate ownship's domain.

This work adopts a decentralized ellipse as the ship domain (Fig. 6.4) similar in form to that proposed by Szlapczynski [125], albeit with different dimensions. The proposed methodology for quantifying good seamanship, however, does not depend on the specific geometry of the ship domain; thus, different geometries could be substituted into the procedures provided hereafter. While various authors have proposed complex polygonal domain geometries, many of these can be approximated by a decentralized ellipse. Further, the governing equations of a decentralized ellipse can be solved analytically [125] while still creating a domain that emphasizes COLREGS maneuvering compliance, e.g., by favoring port-to-port maneuvers or those that cross astern of the target ship. As shown in Fig. 6.4, the geometry of this domain is described by the ellipse axis lengths a and b, and displacements from the ellipse center Δa and

Δb. Values for these parameters are approximated according to the following:

$$a(t) = \tfrac{1}{2}\big(R_{f,d}(t) + R_{a,d}(t)\big), \tag{6.5}$$

$$b(t) = \tfrac{1}{2}\big(R_{s,d}(t) + R_{p,d}(t)\big), \tag{6.6}$$

$$\Delta a(t) = R_{f,d}(t) - a(t), \tag{6.7}$$

$$\Delta b(t) = R_{s,d}(t) - b(t), \tag{6.8}$$

where $R_{f,d}(t)$, $R_{a,d}(t)$, $R_{s,d}(t)$, and $R_{p,d}(t)$ are radii for the fore, aft, starboard, and port sectors of the domain, respectively. Values for these radii are configurable and can be affected by factors such as vessel size, vessel speed, environmental conditions, the skill of the mariner, etc. Following the work of Kijima [54] and Wang [134] with slight modifications for a more conservative domain size, the values of these radii are chosen based on the vessel's evasive maneuvering characteristics, including the length of the vessel in nautical miles L, speed in knots $u_{kts}(t)$, advance A_D, and tactical diameter D_T:

$$R_{f,d}(t) = \left(1 + 1.34\sqrt{k_{A_D}(t)^2 + \big(0.5k_{D_T}(t)\big)^2}\right)2L, \tag{6.9}$$

$$R_{a,d}(t) = \left(1 + 0.67\sqrt{k_{A_D}(t)^2 + \big(0.5k_{D_T}(t)\big)^2}\right)2L, \tag{6.10}$$

$$R_{s,d}(t) = \big(0.2 + k_{D_T}(t)\big)2L, \tag{6.11}$$

$$R_{p,d}(t) = \big(0.2 + 0.75k_{D_T}(t)\big)2L, \tag{6.12}$$

where

$$k_{A_D}(t) = A_D(t)/L \approx 10^{(0.3591 \log_{10} u_{kts}(t)+0.0952)}, \tag{6.13}$$

$$k_{D_T}(t) = D_T(t)/L \approx 10^{(0.5441 \log_{10} u_{kts}(t)-0.0795)}. \tag{6.14}$$

The equations above depend only on $u_{kts}(t)$ and L, which are assumed to be known for ownship and estimated for a target ship through Automatic Identification System (AIS) or other perception systems. In the event that these radii are too conservative for some users, a scale factor λ_r can easily be applied to each domain dimension, i.e., $R_{(\cdot)}(t) = \lambda_r R_{(\cdot)}(t)$.

The primary drawback of a decentralized ellipse is that it is equally acceptable to pass on either side of a vessel in overtaking scenarios. Some works define different domain geometries for these different COLREGS situations [18, 104], but this leads to the possibility of discontinuities when using ship domain for projection calculations. We address this drawback by introducing an equalizer term l_e on the lateral dimensions of the domain based on the relative heading between ownship and the target ship. An overtaking scenario occurs when a vessel approaches another vessel from a direction more than 22.5 deg abaft abeam. Letting ψ_r represent the relative heading between the vessels, this roughly translates to an associated $|\psi_r| \leq 67.5$ deg. The lateral equalizer term is then defined using smoothing bounds $\psi_r^+ = 72.5$ deg and $\psi_r^- = 62.5$ deg to prevent discontinuities in the domain geometry:

$$l_e = \min\left(\max\left(\frac{|\psi_r| - \psi_r^+}{\psi_r^- - \psi_r^+}, 0\right), 1\right). \tag{6.15}$$

The final port and starboard radii then become

$$R_{lat}(t) = \frac{1}{2}\left(R_{s,d}(t) + R_{p,d}(t)\right), \tag{6.16}$$

$$R_{s,d}(t) = (1 - l_e)R_{s,d}(t) + l_e R_{lat}(t), \tag{6.17}$$

$$R_{p,d}(t) = (1 - l_e)R_{p,d}(t) + l_e R_{lat}(t). \tag{6.18}$$

6.3.1.2 Collision Index

The example described in the previous section through Fig. 6.3 highlights the limitations of using t_{cpa} and d_{cpa} for collision risk assessment. To alleviate these problems, Szlapczynski [124] introduced a new measure for collision risk derived from the degree of penetration within a ship domain. At every time instant t, collision risk can be assessed based on a scale factor $f_d(t)$ of the largest domain-shaped area that is free from other vessels, i.e., after scaling a ship's domain by $f_d(t)$, the other vessel will be on the boundary of the scaled domain. Values for $f_d(t) < 1$ can then represent the degree of penetration by a target ship into ownship's domain as shown in Fig. 6.4. Szlapczynski [125] provides analytical formula for calculating $f_d(t)$ on a decentralized ellipse ship domain, however, $f_d(t)$ could also be determined for arbitrary domain geometries [124]. Thus, the reader is free to substitute any ship domain for use with this methodology.

Following and updating some of Szlapczynski's developments [125], we now provide the formula for calculating $f_d(t)$ on a decentralized ellipse for completeness. This derivation calculates $f_d(t)$ of the target ship's domain with respect to ownship position, i.e., how much is the target ship's domain scaled

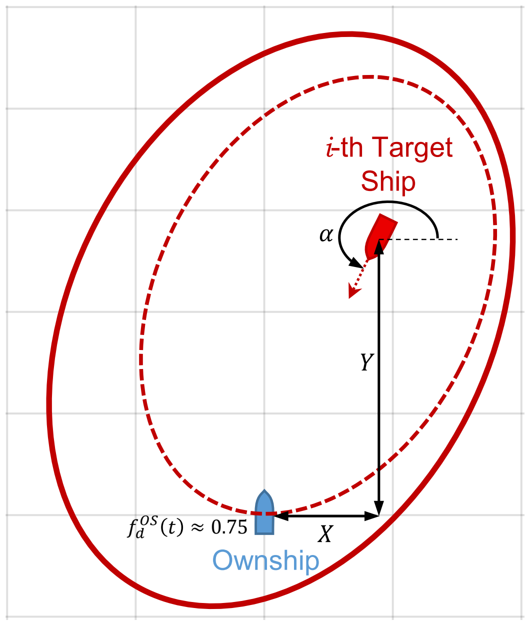


Figure 6.5: Diagram depicting variables used in the derivation of $f_d(t)$.

to get to ownship position. As shown in Fig. 6.5, let X and Y be the lateral and longitudinal relative position of the target ship with respect to ownship's local vehicle frame. Also let V_X and V_Y be the relative velocity of the target ship along these dimensions and α be the yaw angle of the target ship in ownship's local vehicle frame. The center of the rotated f_d-scaled ellipse is given by

$$X_e(t) = X + hf_d + V_X t, \tag{6.19}$$

$$Y_e(t) = Y + kf_d + V_Y t, \tag{6.20}$$

$$h = \Delta a \cos\alpha + \Delta b \sin\alpha, \tag{6.21}$$

$$k = \Delta a \sin\alpha - \Delta b \cos\alpha. \tag{6.22}$$

We aim to solve for $f_d(t)$ using the parametric equation for the f_d-scaled ellipse:

$$\frac{\big(X_e(t)\cos\alpha + Y_e(t)\sin\alpha\big)^2}{f_d(t)^2 a^2} + \frac{\big(X_e(t)\sin\alpha - Y_e(t)\cos\alpha\big)^2}{f_d(t)^2 b^2} = 1. \tag{6.23}$$

By expanding and organizing terms, we can reformulate Eq. (6.23) as

$$A_1 X_e(t)^2 + B_1 X_e(t) Y_e(t) + C_1 Y_e(t)^2 = f_d(t)^2, \tag{6.24}$$

where

$$A_1 = \frac{\cos^2\alpha}{a^2} + \frac{\sin^2\alpha}{b^2}, \tag{6.25}$$

$$B_1 = 2\sin\alpha\cos\alpha\big(\frac{1}{a^2} - \frac{1}{b^2}\big), \tag{6.26}$$

$$C_1 = \frac{\sin^2\alpha}{a^2} + \frac{\cos^2\alpha}{b^2}. \tag{6.27}$$

Plugging Eq. (6.19) and (6.20) into Eq. (6.24) then lets us formulate a quadratic equation for $f_d(t)$:

$$A_2 f_d(t)^2 + (B_{21} + B_{22}t) f_d(t) + C_{21} + C_{22}t + C_{23}t^2 = 0, \tag{6.28}$$

where

$$A_2 = A_1 h^2 + B_1 hk + C_1 k^2 - 1, \tag{6.29}$$

$$B_{21} = (2A_1 X + B_1 Y)h + (2C_1 Y + B_1 X)k, \tag{6.30}$$

$$B_{22} = (2A_1V_X + B_1V_Y)h + (2C_1V_Y + B_1V_X)k, \tag{6.31}$$

$$C_{21} = A_1X^2 + B_1XY + C_1Y^2, \tag{6.32}$$

$$C_{22} = 2(A_1XV_X + C_1YV_Y) + B_1(XV_Y + YV_X), \tag{6.33}$$

$$C_{23} = A_1V_X^2 + B_1V_XV_Y + C_1V_Y^2. \tag{6.34}$$

Finally, solving for $f_d(t)$ in Eq. (6.28) using the standard quadratic equation gives

$$f_d^{1,2}(t) = \frac{-(B_{21} + B_{22}t) \pm \sqrt{D_1t^2 + E_1t + F_1}}{2A_2}, \tag{6.35}$$

where

$$D_1 = B_{22}^2 - 4A_2C_{23}, \tag{6.36}$$

$$E_1 = 2B_{21}B_{22} - 4A_2C_{22}, \tag{6.37}$$

$$F_1 = B_{21}^2 - 4A_2C_{21}. \tag{6.38}$$

When assuming that the relative velocity vector between the vessels remains constant into the future, Eq. (6.35) gives the domain scale factor at any future time for a decentralized ellipse geometry. While the quadratic solution of Eq. (6.35) gives two possible values for $f_d(t)$, the single desired solution is that which is both positive and real. To determine the domain scale factor for ownship's domain, the same calculation from above is repeated from the perspective of the target ship. The superscripts $(\cdot)^{OS}$ and $(\cdot)^{i}$ are henceforth used to refer to values calculated from ownship' perspective and the *i*-th target ship's perspective, respectively.

In order to translate the domain scale factor into a measure used for collision risk evaluation, we define a domain risk index, $r_d \in [0, 1]$, using a logistic function according to the following:

$$r_d(t) = \frac{1}{1 + e^{\kappa (f_d(t) - f_0)}}, \tag{6.39}$$

where κ and f_0 are parameters that define the shape of the logistic curve. Equation (6.39) was chosen based on findings that indicate the safety perceived by mariners is roughly proportional to the logarithm of the vessel separation distance [103]. This work uses $\kappa = 10$ and $f_0 = 0.5$ based on tuning experiments and the resulting domain risk is shown by the inner contour plots of Fig. 6.4.

Several works only consider violations of ownship's domain [134, 104, 68] or the target ship's domain [35, 125] when assessing current and future collision risk. We believe these interpretations are incomplete, as the actual collision risk between two vessels should be identical regardless of each vessel's perspective. Thus, we define the combined mutual domain risk between ownship and the *i*-th target ship as such:

$$r_d^{OS,i}(t) = r_d^{OS}(t) + r_d^{i}(t)\left(1 - r_d^{OS}(t)\right), \tag{6.40}$$

where again, r_d^{OS} is calculated from the perspective of ownship and r_d^{i} is calculated from the *i*-th target ship perspective.

Subsequently, the collision index between the ASV and the *i*-th target ship at time t is defined as the maximum value of the mutual domain risk over a future time horizon T:

$$\Theta_C^i = \max_{\tau \in [t, t+T]} r_d^{OS,i}(\tau). \tag{6.41}$$

This optimization can be easily solved numerically by assuming constant speed and heading for each vessel. Figure 6.6 shows a projection of both vessels at the future time of maximum mutual domain risk, deemed t_C^i. The values Θ_C^i and t_C^i are meant to provide analogous risk metrics to d_{cpa} and t_{cpa}, albeit generalized through the ship domain geometry to capture additional features that contribute to the overall risk.

6.3.2 Appropriate Action Quantification

6.3.2.1 Ship Arena

Similar to the concept of ship domain is the notion of ship arena, defined as the area around ownship where a mariner should begin maneuvering if a collision risk exists. The ship arena naturally encompasses a larger area than the ship domain, as any evasive action should be planned and executed well before violations of each ship's domain. Additionally, by defining a geometry that prefers earlier action to vessels in ownship's fore and starboard sectors, the ship arena can capture the give-way and stand-on expectations governed by COLREGS.

This work again chooses a decentralized ellipse to define the ship arena, albeit with different dimensions of $R_{f,a}$, $R_{a,a}$, $R_{s,a}$, and $R_{p,a}$. It is typical

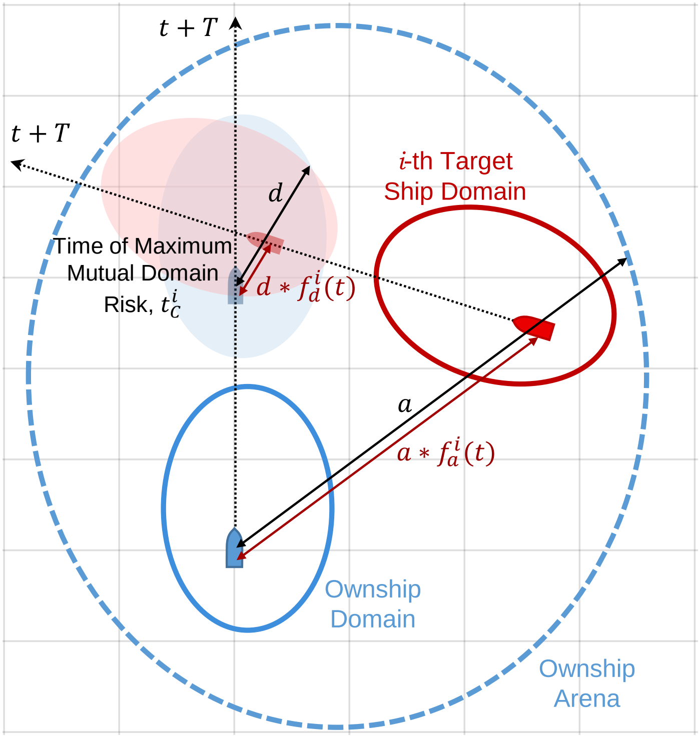


Figure 6.6: Illustration of overall risk methodology. Θ_C^i is calculated as the maximum mutual domain risk with the i-th target ship over a future time horizon $t + T$. Θ_A^i is calculated based on the i-th target ship's degree of penetration within ownship's arena.

for mariners to internally set predefined distances at which they begin to consider maneuvering actions. In give-way situations, taking early action is preferred, sometimes right when a target ship is reliably detected on radar or AIS. Conversely, in stand-on situations, mariners are expected to maintain course and speed until the target ship has been deemed noncompliant, i.e., the target ship does not perform its own evasive action. Evasive action by ownship in these noncompliant stand-on situations would then occur at distances much shorter than the detection or give-way range. It should be emphasized that the dimensions of exactly when a ship should begin maneuvering are strictly based on the preferences of the mariner. There is a dearth of literature on more principled methods to make these parameter selections; thus, values for $R_{f,a}$, $R_{a,a}$, $R_{s,a}$, and $R_{p,a}$ can be set arbitrarily based on the size of the vessels and their perception capabilities. Future research should certainly guide the community into more accepted standards for defining the ship arena based on the vessel classification (e.g. sailing vessel, etc.) and relative velocity, as these values all affect when evasive action should be considered.

6.3.2.2 Action Index

Once the ship arena is defined, the action index with respect to the i-th target ship, Θ_A^i, is calculated in a similar fashion to r_d:

$$\Theta_A^i = \frac{1}{1 + e^{\kappa(f_a^i(t) - f_0)}}, \tag{6.42}$$

where now the value $f_a^i(t)$ is the ship arena scale factor such that the i-th target ship lies on the boundary of ownship's scaled ship arena. This representation can be thought of as the degree to which ownship should take action and can appropriate quantify delayed avoidance maneuvers. When compared to the calculation of Θ_C^i, it should be noted that Θ_A^i is evaluated for each target ship only from the perspective of ownship.

6.3.3 Overall Risk Index

We can now define an overall risk index for the i-th target ship as

$$\Theta_S^i = \Theta_C^i \Theta_A^i. \tag{6.43}$$

As visualized in Fig. 6.6, the combination of Θ_C^i and Θ_A^i is important for evaluating seamanship – even if a target ship is within the ASV ship arena, the overall risk should be low if there is no future collision risk. Similarly, the overall risk should be low if a future collision risk exists, but the target ship is far off into the distance.

This overall risk index is able to better capture periods of high risk in the ASV trajectory when compared to CPA-based methods. Specifically, the time and location of maximum collision risk is not necessarily the same as the time and location of CPA. For a vessel crossing the bow of another vessel, it is likely that the point of maximum collision risk occurs well before CPA. Figure 6.7 illustrates this scenario with two snapshots of vessel geometry: one at the point of maximum risk and another at CPA. It is clear that the overall risk between the two vessels is significantly less in the CPA configuration even though the vessels are closer together.

For a multi-vessel scenario involving $i = 1, \ldots, N$ target ships, the risk associated with each target ship can be combined to capture the risk associated with the overall scenario, Φ_S. As opposed to using the average value of all Θ_S^i, this work proposes the overall scenario risk should be calculated as the union of individual risk indices through the following recursion:

$$\begin{aligned} &\textsc{While}\ i \leq N \\ &\Phi_S = \begin{cases} \Theta_S^i & \text{if } i = 1 \\ \Theta_S^i + \Phi_S \left(1 - \Theta_S^i\right) & \text{if } 1 < i \leq N \end{cases} \end{aligned} \tag{6.44}$$

The logic behind this formula is that the overall scenario risk should be *at least* as large as the highest risk from the *i*-th target ship, with additional risk from

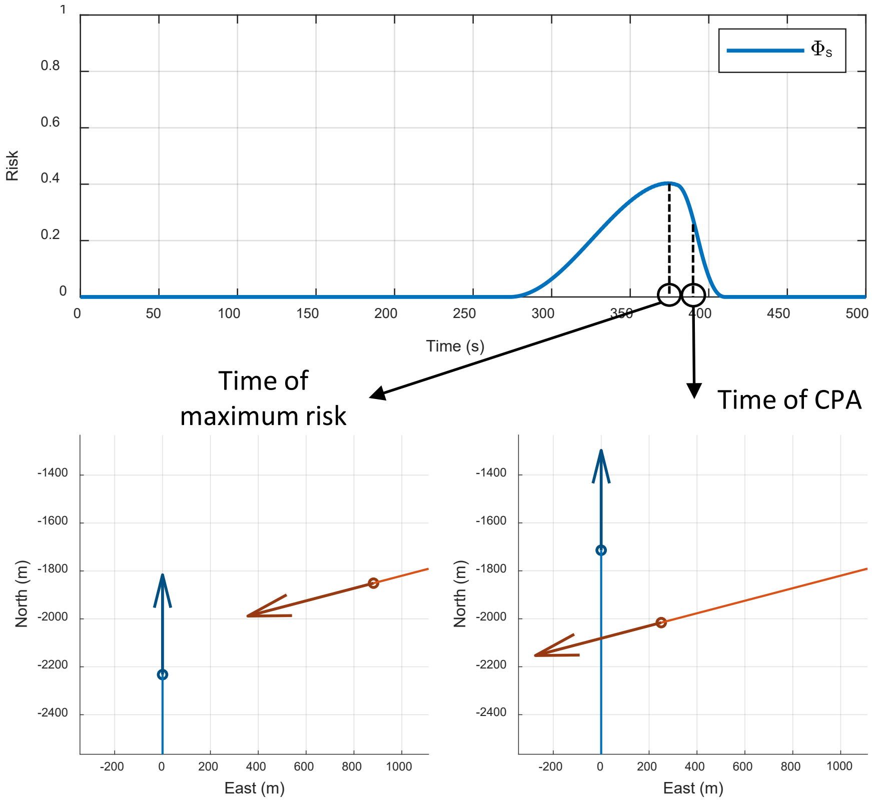


Figure 6.7: Overall risk curve for a two-vessel crossing scenario. Snapshots of the vessel configuration are given at the time of maximum overall risk (left) and the time of CPA (right).

other vessels only augmenting the value of Φ_S. The value provided through Φ_S provides a quantitative metric for evaluating the overall risk of a given multi-vessel configuration or sequence of events. By naturally incorporating the essence of COLREGS through domain geometry rather than rule-based heuristics, this metric generalizes the core seamanship principles that remain important even in complicated multi-vessel scenarios.

6.4 Path Planning Approach

This section now proposes an approach to ASV path planning by naturally extending COLREGS compliance to multi-vessel encounters through the good seamanship metrics of Section 6.3. The two key components of the overall decision-making are modules relating to situational awareness and path planning. The situational awareness module categorizes an encounter with respect to relevant hazards and vessels in the environment including the COLREGS type, requirements of ownship, etc. The planning module then consists of a three-layered architecture with (i) a high-level, primitive-based branch-and-bound planner, (ii) a mid-level clothoid-based RRT* [8, 50], and (iii) a low-level trajectory follower. The high-level branch-and-bound planner uses the output from the situational awareness logic to prescribe a sequence of H state nodes separated by timestep δ_t over a receding planning horizon $T = \delta_t H$. In the descriptions that follow, we denote $(\cdot)_n$ to refer to a future state node calculated at time $t + n\delta_t$ in the n-th stage of the sequence. It is desired that these nodes constitute a coarse trajectory that exhibits good seamanship and other aspects of COLREGS compliance, mission efficiency, and safety considerations (Fig. 6.8). This state node sequence is connected by the mid-level clothoid-based RRT* to produce a continuous and feasible trajectory at a much finer time scale, which is then tracked using line of sight guidance techniques [14, 60, 33].

Similar to the planning approaches discussed in Part I, the focus in this thesis is on the high-level planning that would handle COLREGS compliance and good seamanship on the timescales seen in surface craft interactions (i.e., larger than the system dynamics). Determining this high-level path remains

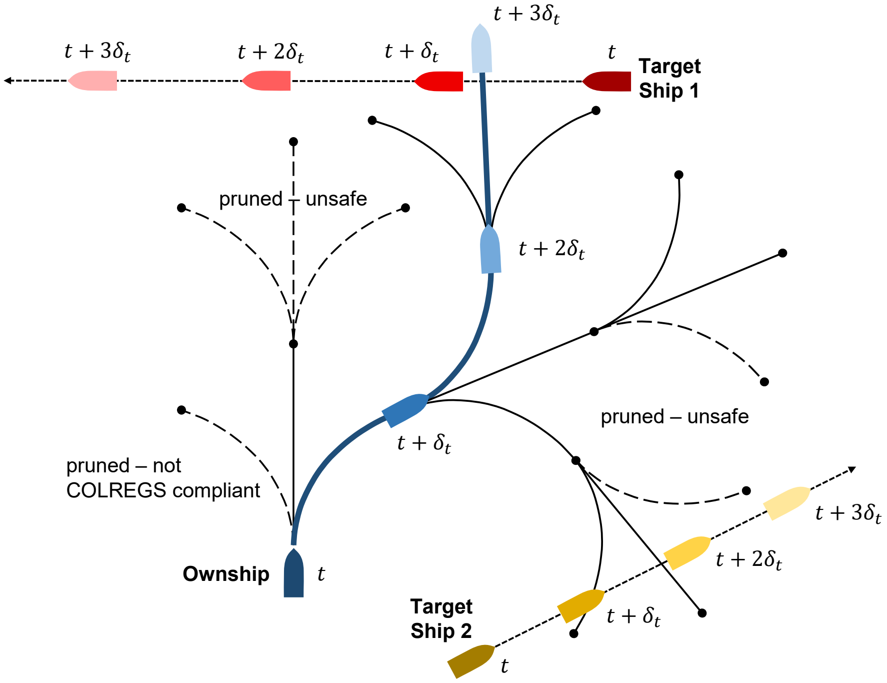


Figure 6.8: Primitive-based receding horizon planning that encodes good seamanship principles for multi-vessel scenarios.

the most open challenge in ASV navigation, as the trajectory tracking methods used in the mid- and low-level planners are fairly well established, or could be easily substituted for more recent methods such as MPC-based tracking for ASVs [11].

6.4.1 Situational Awareness

The module performing situational awareness is meant to incorporate all perception information into a simplified world model $\mathcal{S}$ on which path planning can operate. Following the procedure of Alg. 4, we define $\mathcal{V}$ to be the set of all perceived target ships and $\mathcal{V}^*$ to be the set of *relevant* target ships considered

Algorithm 4 SITUATIONAL AWARENESS

```
procedure OVERALL_SITUATION($\boldsymbol{x}, \mathcal{V}$)
    for each $V^i \in \mathcal{V}$ do
        if $\Theta^i_C > \tilde{\Theta}_C$ & $t^i_C < \tilde{t}_C$ then
            $\mathcal{V}^* \leftarrow (\mathcal{V}^* \cup \{V^i\})$
            $(\mathcal{C}_{type}, \mathcal{C}_{exp})^i \leftarrow$ COLREGS type and expectation   ▷ determined from [137]
            if $\Theta^i_S > \tilde{\Theta}_S$ then
                $\mathcal{C}^i_{type} \leftarrow$ EMERGENCY;  $\mathcal{C}^i_{exp} \leftarrow$ GIVE-WAY  ▷ Ownship *in extremis*
            end if
        end if
    end for
    if size($\mathcal{V}^*$) $= 0$ then
        $\mathcal{S}_{type} \leftarrow$ CLEAR; $\mathcal{S}_{exp} \leftarrow$ ANY
    else if any($\mathcal{C}^i_{type}$) $=$ EMERGENCY then
        $\mathcal{S}_{type} \leftarrow$ EMERGENCY;  $\mathcal{S}_{exp} \leftarrow$ GIVE-WAY
    else if size($\mathcal{V}^*$) $= 1$ then
        $\mathcal{S}_{type} \leftarrow$ COLREGS;  $\mathcal{S}_{exp} \leftarrow \mathcal{C}^i_{exp}$
    else
        $\mathcal{S}_{type} \leftarrow$ CONGESTION
        if any($\mathcal{C}^i_{exp}$) $=$ GIVE-WAY then
            $\mathcal{S}_{exp} \leftarrow$ GIVE-WAY
        else
            $\mathcal{S}_{exp} \leftarrow$ STAND-ON
        end if
    end if
    return $\mathcal{S} \leftarrow \{\mathcal{V}^*, \mathcal{S}_{type}, \mathcal{S}_{exp}\}$
end procedure
```

during planning, i.e., those for which Θ^i_C and t^i_C satisfy user-specified thresholds $\tilde{\Theta}_C$ and $\tilde{t}_C$, respectively. COLREGS is categorized for each relevant target ship and $\mathcal{S}$ is defined according to the number of relevant target ships as well ownship's expectation for each relevant target ship.

6.4.2 Cost Function

ASV navigation requires many different considerations when evaluating the quality of proposed paths. We wish to follow a nominal trajectory π_{nom} as closely as possible while minding traditional aspects such as collision avoidance, as well as domain-specific aspects such as COLREGS compliance and the good seamanship principles introduced in Chapter 5. The cost function $J : \mathcal{Z} \to [0,1]$ for evaluating potential trajectories thus consists of a weighted average of mission cost J_M, safety cost J_S, and COLREGS cost J_C as follows:

$$J = \frac{w_M J_M + w_S J_S + w_C J_C}{w_M + w_S + w_C}, \tag{6.45}$$

where w_M, w_S, and w_C are weights for the mission, safety, and COLREGs criteria, respectively.

6.4.2.1 Mission Cost

Mission cost at an n-th stage node is calculated as the cumulative normalized distance from the nominal trajectory over the planning horizon, i.e.,

$$L_M(\boldsymbol{x}_n, \boldsymbol{u}_n) = \frac{||\boldsymbol{p}_n - \boldsymbol{p}_{n,nom}||}{||\boldsymbol{p}_0 - \boldsymbol{p}_{n,nom}|| + u\delta_t n}. \tag{6.46}$$

The denominator in this equation uses the maximum distance that could be traveled from the nominal trajectory over the future time length as a normalization factor. The mission cost for a trajectory up to the n-th stage is then the sum of individual node mission costs:

$$J_M = \sum_{i=1}^{n} L_{M,i}, \tag{6.47}$$

where a complete trajectory would be a total of $n = H$ nodes.

6.4.2.2 Safety Cost

Safety cost is determined from a combination of risk factors – both from static obstacles in the form of a cost map and from dynamic vessels using the seamanship-based risk metric defined in Section 6.3. Let $\mathcal{C} : \boldsymbol{p} \rightarrow [0, 1]$ be a cost map of position states over the local environment. This cost map is built based on information publicly available in electronic navigational charts (ENCs) to include known hazards (buoys, bridges, etc.) as well as bathymetry data and shoreline boundaries. Real-time perception of additional static obstacles can then easily augment this cost map beyond the information available in ENCs. Then, using developments from Section 6.3, the final safety cost is defined as the union of risks from both static obstacles and the overall, multi-vessel scenario risk:

$$L_S(\boldsymbol{x}_n, \boldsymbol{u}_n) = \Phi_{S,n} + \Phi_{S,n}(1 - \mathcal{C}(\boldsymbol{p}_n)). \tag{6.48}$$

The use of Φ_S in the safety cost captures the risk from all relevant vessels in a manner that also prioritizes good seamanship practices. The safety cost for a trajectory up to the n-th stage is then the sum of individual node safety costs:

$$J_S = \sum_{i=1}^{n} L_{S,i}, \tag{6.49}$$

where a complete trajectory would be a total of $n = H$ nodes.

6.4.2.3 COLREGS Cost

The COLREGS component of the cost function is intended to capture other features of good seamanship that are difficult to incorporate within the ship domain approach. Portions of this cost are informed by the work of Woerner [136, 137], which proposed a framework to evaluate the COLREGS compliance of single-vessel encounters. While Woerner's framework was designed for

Algorithm 5 COLREGS COST METHODOLOGY

```
procedure COLREGS_COST($\pi, \mathcal{S}$)
    Set $Q_C \leftarrow 1$
    if $\mathcal{S}_{exp}$ = GIVE-WAY & $\mathcal{S}_{type} \neq$ EMERGENCY then
        $Q_C \leftarrow p_D Q_C$    ▷ Penalize for delayed actions ($p_D$ from Alg. 5 of [137])
        $Q_C \leftarrow p_M Q_C$    ▷ Penalize non-obvious maneuvers ($p_M$ from Alg. 7 and 8 of [137])
        $Q_C \leftarrow p_I Q_C$    ▷ Penalize action indecision
        $Q_C \leftarrow p_S Q_C$    ▷ Penalize speed-only actions (prefer course changes)
    else if $\mathcal{S}_{exp}$ = STAND-ON then
        $Q_C \leftarrow$ COLREGS score for Rule 17    ▷ (Alg. 9 of [137])
    end if
    return $J_C \leftarrow 1 - Q_C$
end procedure
```

use on post-processed trajectories, we have adapted the work here to evaluate the quality of a proposed path (from a COLREGS perspective). Several additional penalties that generalize to multi-vessel encounters are also applied as outlined in Alg. 5. When ownship is not in an emergency state, these include principles such as ensuring that control actions produce readily apparent maneuvers, creating a predictable trajectory by reducing changes in control actions (i.e., minimizing indecision), and preferring course changes to speed changes.

The two components of Alg. 5 not based in the work of Woerner [137] are the definitions of p_I and p_S, referring to an indecision penalty and speed change penalty, respectively. For the former, we define an activation function

$$\delta\xi_n = \begin{cases} 0 & \text{if } (\xi_n - \xi_{n-1}) = (0,0), \\ 1 & \text{else,} \end{cases} \tag{6.50}$$

to represent changes between sequential motion primitive parameters. The penalty used to minimize action indecision (and thus make maneuvers more predictable to other target ships) is then calculated as

$$p_I = 1 - \frac{\max\left(\sum_{n=2}^{H} \delta\xi_n - 1, 0\right)}{3\left(1 + \sum_{n=2}^{H} \delta\xi_n\right)}. \tag{6.51}$$

The second term p_S is that which encourages course changes over speed changes when choosing a trajectory. Changes in course are much more perceivable to the target ship, both visually and over radar, and further reduce potential confusion during an encounter. This penalty is set to a simple constant of $p_S = 0.9$ that is applied to primitives where $\delta_u \neq 0$ and $\delta_\psi = 0$.

6.4.3 Branch & Bound Path Planning

Let high-level planning occur in the parameter space $\mathcal{Z}$ such that a trajectory π consists of state nodes derived from a sequence of motion primitive parameters $\boldsymbol{z}$, i.e., $\pi = \{\boldsymbol{x}_n\}_{n=1}^{H} = \varphi(\boldsymbol{z})$. The computation time of planning over this finite horizon is exponential based on the value of H, thus making brute force search methods infeasible for the planning horizons required over surface craft encounters. Branch-and-bound planning alleviates this burden by pruning regions of the parameter space that are unlikely to contain the optimal solution based on estimates of the expected costs in those regions. The branch-and-bound planner of Alg. 6 implements a receding horizon strategy such that a final trajectory is computed up to the local planning horizon T, which is then periodically re-planned at a nominal refresh rate of 0.25 Hz, or when critical events trigger a new plan (such as when $\mathcal{S}_{type} = \text{EMERGENCY}$ or the size of $\mathcal{V}^*$ changes). Efficiency is increased by “warm starting” the tree generation by setting π^* and γ^* as the values obtained from following π_{nom}.

Algorithm 6 Branch & Bound Primitive Planning

```
procedure BB_Planner($\boldsymbol{x}_{start}$, $\pi_{nom}$, $\mathcal{S}$)
    Initialize $\gamma_{LB}$, $\delta_t$, $H$, $M$
    Initialize $[\pi^*, \gamma^*]$ based on the cost of following $\pi_{nom}$
    Evaluate cost $\gamma_{u=0}$ of zero-speed trajectory $\pi_{u=0}$
    if $\gamma_{u=0} < \gamma^*$ then
        $\gamma^* \leftarrow \gamma_{u=0}$;  $\pi^* \leftarrow \pi_{u=0}$
    end if
    $[\pi^*, \gamma^*] \leftarrow$ BB_Recursion($\emptyset, 0, 1$)
     return $\pi^*$
end procedure

procedure BB_Recursion($\boldsymbol{z}_{parent}$, $\gamma_{parent}$, $n$)
    Estimate $\gamma_{LB}$    ▷ Eq. (6.56)
    if $(\gamma_{parent} + \gamma_{LB}) < \gamma^*$ & $n \leq H$ then
        Sample primitives: $\{\xi_k\}_{k=1}^{M}$    ▷ Eq. (6.52) and (6.53)
        Extend trajectory parameterizations: $Z \leftarrow \{\boldsymbol{z}_k\}_{k=1}^{M} \leftarrow \{(\boldsymbol{z}_{parent}, \xi_k)\}_{k=1}^{M}$
        Generate state trajectories: $\Pi \leftarrow \{\pi_k\}_{k=1}^{M} \leftarrow \varphi(Z)$    ▷ Starting from $\boldsymbol{x}_{start}$
        Calculate cost $\{\gamma_k\}_{k=1}^{M}$ of each $\pi_k \in \Pi$:    ▷ Eq. (6.45), $\pi_{nom}$, $\mathcal{S}$
        Sort $\pi_k$ by increasing $\gamma_k$
        for each $\pi_k \in \Pi$ do
            if $\gamma_k < \gamma^*$ then
                $\gamma^* \leftarrow \gamma_k$;  $\pi^* \leftarrow \pi_k$
            end if
            $[\pi^*, \gamma^*] \leftarrow$ BB_Recursion($\boldsymbol{z}_k$, $\gamma_k$, $n+1$)
        end for
    end if
     return $\pi^*$, $\gamma^*$
end procedure
```

6.4.3.1 Branching Procedure

The edges of the branch-and-bound tree at each n-th stage consist of a set of M primitive curves with constant velocity and turn rate, each parameterized using $\{\xi_k\}_{k=1}^{M}$ as defined in Section 6.2. Using a full dynamic model such as Eq. (6.3) would require several integration steps to generate each segment, resulting in significant increases to computation time. Thus, for high-level planning at this stage, we utilize the kinematic trajectory parameterization of Eq. (6.4) that can be solved in closed form, while more rigorous dynamic constraints are handled by the other layers of the overall system.

Values for the trajectory parameters are chosen from discrete sets at each level of the tree:

$$\delta_u \in \begin{cases} \{-\frac{2}{3}u_{nom}, -\frac{1}{3}u_{nom}, 0\} & \text{if } n = 1, \\ \{0\} & \text{otherwise,} \end{cases} \tag{6.52}$$

$$\delta_\psi \in \tfrac{\pi}{180}\{-60, -45, -30, 0, 30, 45, 60\} \; \forall \, n. \tag{6.53}$$

The strategy behind Eq. (6.52) is to choose a single speed at the beginning of the trajectory, which minimizes M and allows the depth of the tree H to be maximized based on the system's available computing resources. Additionally, as shown in Alg. 6, rather than including a speed change primitive that brings ownship to rest, a zero-speed trajectory $\pi_{u=0}$ is evaluated as a single instance before tree generation. This is because the vessel is underactuated and only one zero-speed trajectory is possible, regardless of the sequence of desired heading changes. The choice of Eq. (6.53) also ensures that if a heading change is chosen, that a minimum of 30 deg is used to remain readily apparent as required by COLREGS. This threshold is not explicitly written in the COLREGS protocols, but it has become commonly accepted by the U.S. Coast Guard [136, 137].

The choices of δ_t, H, and M are also crucial in generating desirable maneuvers. To ensure that control actions are considered both before and after the relevant vessel interactions while minding computational constraints, this work chooses

$$\delta_t = \max\big(\min_i(t_C^i)/\ell, \, \tilde{\delta}_t\big), \tag{6.54}$$

where $\tilde{\delta}_t$ is a nominal minimum value for the time step. This formula scales

δ_t based on the time until maximum mutual domain risk, such that longer planning horizons are accommodated if the interaction is far away. The choice of ℓ sets the number of control actions considered until t_C^i.

6.4.3.2 Bounding Procedure

The lower bound on the overall cost is calculated as $J_{LB} : \mathcal{Z} \rightarrow [0, 1]$, and is used as a heuristic to prune branches unlikely to contain the optimal solution such that an exponential number of trajectories is not explored. Due to the nature of the cost function in Section 6.4.2, it is entirely possible for γ^* to equal zero if π_{nom} were to be tracked exactly while not encountering any relevant vessels or hazards. Thus, the lower bound must assume that the safety and COLREGS portions of the final cost are zero at unobserved state nodes. The remaining portion of the lower bound is estimated as the mission cost-to-go for the current branch to return back to the nominal trajectory. As illustrated in Fig. 6.9, we define $\hat{\boldsymbol{p}}_{n+1}$ to be the position projected towards π_{nom} at the $(n+1)$-th stage by a distance $u\delta_t$:

$$\hat{\boldsymbol{p}}_{n+1} = \begin{cases} \boldsymbol{p}_{n+1,nom} & \text{if } ||\boldsymbol{p}_{n+1,nom} - \boldsymbol{p}_n|| \leq u\delta_t, \\ \boldsymbol{p}_n + \frac{(\boldsymbol{p}_{n+1,nom} - \boldsymbol{p}_n)}{||\boldsymbol{p}_{n+1,nom} - \boldsymbol{p}_n||} u\delta_t & \text{otherwise.} \end{cases} \tag{6.55}$$

Successive projections are then made from the previous projection up until the end of the planning horizon. The lower bound on the cost over the remaining planning horizon (for the branch under evaluation) is then estimated as the total mission cost of these direct position projections:

$$J_{LB} = \sum_{n}^{H-1} \frac{w_M ||\hat{\boldsymbol{p}}_{n+1} - \boldsymbol{p}_{n+1,nom}||}{(w_M + w_S + w_C)(||\boldsymbol{p}_0 - \boldsymbol{p}_{n+1,nom}|| + u\delta_t(n+1))}. \tag{6.56}$$

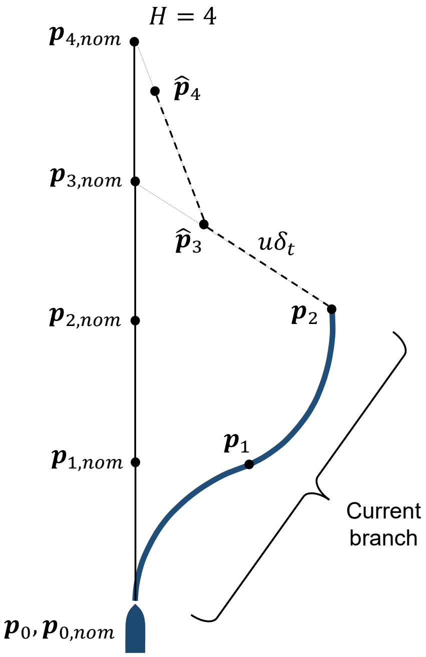


Figure 6.9: Example of the procedure used to estimate the J_{LB} heuristic, where $H = 4$ was set arbitrarily for illustration purposes.

Letting γ_{parent} be the cost accumulated on the current trajectory branch and γ_{LB} be the value resulting from Eq. (6.56), if $(\gamma_{parent} + \gamma_{LB}) \geq \gamma^*$, then the current branch is pruned, as it is highly unlikely to produce the optimal solution given the estimated future cost.

Chapter 7

Towards Verifying Autonomy Decision-Making through Simulation-Based Testing

As discussed in Chapter 1, building trust in autonomous systems means that algorithms such as the ASV planner of Chapter 6 need to be rigorously tested. This chapter takes a brief detour from ASV navigation to present an algorithmic approach for continuous performance analysis of autonomous systems using large-scale simulation data. First, adaptive scenario generation is used to identify unknown failure modes in the system under test (SUT), followed by performance regression analysis to determine if the system has degraded in performance against either previous software versions or other baseline approaches. The proposed framework is generalized such that it can be applied to any black-box SUT, and will then be used in Chapter 8 to analyze the performance of the good seamanship ASV planning strategy.

7.1 Introduction

7.1.1 Autonomy Testing & Evaluation

Given the need to trust autonomous systems in safety-critical domains, it is well understood that rigorous testing and evaluation (T&E) methods are needed to ensure robust performance [4]. The challenge of properly validating the decision-making aspects of an autonomous system, however, is extremely difficult. It is typical that even a moderately complex autonomy will have many interacting components that obfuscate predicting its actions under all possible stimuli. Modern autonomous systems are also inherently stochastic in nature, either through probabilistic decision-making or updates to their control policy based on learned information. These challenges force evaluators to treat the autonomous system as a black box – one that also exhibits a highly nonlinear and/or discontinuous performance surface.

Autonomy T&E encompasses a large swath of research summarized by Corso et al. [20]. To date, much of this literature focuses on fault detection and software robustness [17, 84] or formal methods [71] as opposed to evaluating the actual decision-making of the system. In realistic, complex autonomy, it is not sufficient to confirm that the system performs appropriately in isolated scenarios. Rather, a much larger testing space (i.e., the space of all possible test conditions under consideration) is required to evoke emergent behaviors that may only appear under certain combinations of stimuli.

As emphasized by Alexander et al. [1], evaluating the decision-making of an autonomous system in operational environments requires a diverse test suite that provides situation coverage of the testing space. Field experiments in realistic settings are the ideal avenue to perform autonomy T&E; however, they are expensive and time-consuming. Simulation-based testing offers an alternative that can produce large datasets under a wide array of test conditions for statistical performance analysis (under the assumption that the simulation is a reasonably good predictor of the true SUT behavior). Techniques for obtaining these datasets primarily focus on scenario generation methods [110]. For example, Li et al. [65] used exhaustive test generation methods over sensor ranges to discover performance transitions in urban driving environments, while Zou et al. [141] used multi-objective optimization techniques to identify failures in multi-UAV conflict resolution algorithms.

Scenario generation based on importance sampling is of particular interest here, as several veins of research [131, 130, 20] have shown this to be an effective method to identify unknown failure modes within high-dimensional testing spaces. These methods auto-generate scenarios that balance the exploration-exploitation trade-off of both characterizing autonomy performance under a wide array of test conditions, while preferentially generating scenarios that hone in on areas of interest (often those that stress the decision-making and lead to deficient performance). Work by O'Kelly et al. [100] employs this idea by using the cross-entropy method to detect rare-event failures in autonomous driving scenarios. In fact, previous work by the author in collaboration with Mullins [91, 92, 93] has also developed intelligent scenario generation techniques to identify the failure modes and performance boundaries of an autonomous system. The concepts in these works lay the foundation that will be further developed in Section 7.2.

7.1.2 Performance Regression Testing

A significant technical gap within autonomy T&E exists when it comes to *comparing* the current performance of the SUT against its previous versions, i.e., performance regression testing. While traditional regression testing is common in software development for code analysis and fault detection [139], autonomous systems introduce a new challenge where it also becomes necessary to ensure that software changes do not adversely affect the holistic performance and behavior of the autonomy in unexpected ways, particularly when changes are constantly made during active development.

When using simulation-based testing to study holistic performance, the question becomes how to analytically compare the performance and failure regions of one dataset to another. Figure 7.1 illustrates this notion of performance regression, where example performance landscapes for two different autonomy versions (A and B) are shown side-by-side. These scatter plots represent simulated scenarios for various test parameter combinations. The color shows the performance score assigned to each scenario based on the autonomy's actions. Two failure regions (red and gray) are highlighted between the datasets. The red failure region exists in version A, but is fixed based on software changes made for version B. The gray region does not exist in version A, but rather appears as a new, unexpected failure mode in version B. In this sense, the gray region represents a region of performance regression characterized by a decrease in performance.

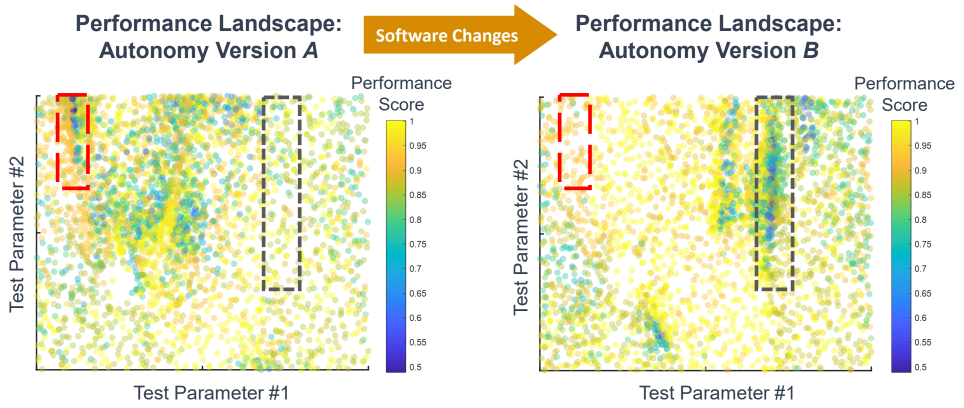


Figure 7.1: Simulation performance datasets for two versions of autonomy software. The red box highlights performance improvement whereas the gray box highlights performance regression.

The underlying research objective highlighted by the example of Fig. 7.1 is to apply statistical learning techniques to analytically compare two (potentially high-dimensional) performance landscapes. Such an analysis helps ensure that more "mature" releases of autonomy software reduce the failure space and do not produce new, unexpected failure modes. Alternatively, this process could also be used to conduct a principled performance comparison between two different autonomy strategies (or two different autonomy products) tested within the same context.

While simulation-based testing is an effective way to generate performance datasets, there is more limited literature on performance regression techniques. Work by Rocklage et al. [111] considered auto-generating scenarios for regression tests in the context of self-driving car maneuvering. The problem posed here, however, is the more general comparison of performance datasets, one which shares similarities with the change detection field. Applications within this field usually fall into detecting changes within time-series information [2]

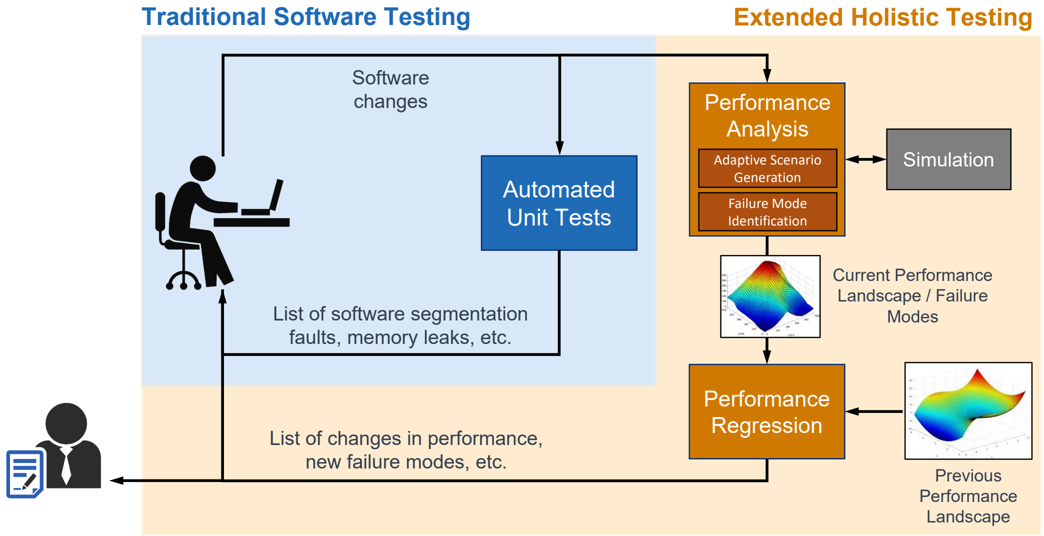

Traditional Software Testing
Extended Holistic Testing
Software changes
Automated Unit Tests
Performance Analysis
Adaptive Scenario Generation
Failure Mode Identification
Simulation
Current Performance Landscape / Failure Modes
List of software segmentation faults, memory leaks, etc.
Performance Regression
List of changes in performance, new failure modes, etc.
Previous Performance Landscape

Figure 7.2: A design for augmented continuous integration for testing autonomy software. In addition to traditional unit testing that checks for software quality issues, automated simulation-based testing is used to evaluate the holistic performance of the system. Additionally, statistical learning is used to detect performance regression, i.e., areas where the SUT may have degraded in performance when compared to its previous versions.

or between images [86, 115]. Generalized methods that detect differences in the underlying probability distributions between datasets also exist [116, 99].

The remainder of this chapter is thus focused on (i) a performance analysis methodology utilizing adaptive scenario generation to produce performance datasets that identify failure modes in the system (such as those in Fig. 7.1), and (ii) developing a methodology to conduct performance regression analysis between two of these (potentially non-identical) datasets. The overall goal is to use both of these pieces within an augmented continuous integration framework for testing autonomy software, shown in Fig. 7.2. This form of holistic performance testing is envisioned as a complement to the traditional practice of unit-based software testing, which would typically only capture software errors such as memory leaks or segmentation faults. An augmented continuous integration pipeline such as Fig. 7.2 would not only check for software quality, but also automatically test the decision-making of the autonomy, i.e., the "brain" of the system, to ensure that failure regions are identified and properly tracked.

7.2 Performance Analysis Methodology

The focus of the performance analysis presented here is on the holistic decision-making aspects of the autonomy that are evoked through simulation-based testing. From a development standpoint, the desired output of the analysis is the discovery of test parameters that produce unexpected failure modes, defined as regions of the testing space characterized by undesirable performance and/or behavior in the autonomy. For autonomy T&E, discovering these failure modes allows developers to fix the issues or limit the operational profile of the system.

A high-level depiction of the analysis methodology (influenced by [93]) is shown in Fig. 7.3, where adaptive scenario generation is used to emphasize data collection around failure modes, followed by unsupervised clustering methods to separate unique failures conditions within the dataset. This framework introduces two key inputs of the process: (i) determining the test parameters and their values to be varied between simulations, referred to as the testing space, and (ii) the performance criteria on which the autonomy is evaluated based on its actions, referred to as the performance space.

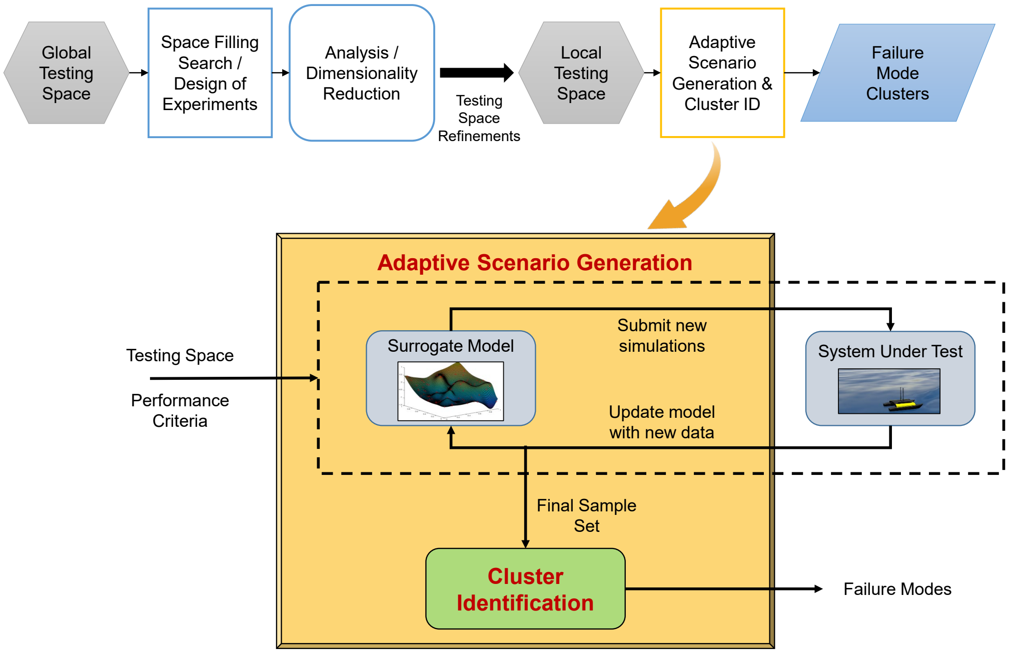

Global Testing Space
Space Filling Search / Design of Experiments
Analysis / Dimensionality Reduction
Testing Space Refinements
Local Testing Space
Adaptive Scenario Generation & Cluster ID
Failure Mode Clusters
Adaptive Scenario Generation
Testing Space
Performance Criteria
Surrogate Model
Submit new simulations
System Under Test
Update model with new data
Final Sample Set
Cluster Identification
Failure Modes

Figure 7.3: High-level flow diagram of the performance analysis methodology to identify failure modes, influenced by the work in [93].

7.2.1 Problem Setup

Formally, the autonomous SUT is treated as a function $\mathcal{F} : \mathcal{X}^D \rightarrow \mathcal{Y}$ that maps a D-dimensional testing space $\mathcal{X}^D = \mathcal{X}_1 \times \ldots \times \mathcal{X}_d$ to a scalar performance space $\mathcal{Y} \subset \mathbb{R}$. A scenario $\boldsymbol{\eta} = [\eta_1, \ldots, \eta_D]^T \in \mathcal{X}^D$ is a specific instantiation of the testing space, where $\eta_k \in \mathcal{X}_k$ is the value of each test parameter. The performance score $y = \mathcal{F}(\boldsymbol{\eta}) \in \mathcal{Y}$ is the output of each simulated scenario.

Ultimately, we wish to identify regions of the testing space characterized by low performance scores (i.e., failure modes) as $\mathcal{Q} = \{\mathcal{X}^D \,|\, \mathcal{Y} < y^*\}$, where y^* is a performance threshold that indicates failure scores of interest. However, because the performance landscape of a black-box autonomy exists in a high-dimensional (and often highly nonlinear) space, it is difficult to characterize

$\mathcal{Q}$ analytically. Thus, the structures of $\mathcal{Q}$ and $\mathcal{Y}$ are estimated using samples collected from the SUT in the form of simulation datasets.

Let sets of N scenarios $X = \{\boldsymbol{\eta}_i\}_{i=1}^{N}$ and their resulting performance scores $Y = \{y_i\}_{i=1}^{N}$ be combined to form a study $S = \{X, Y\}$. Further, the algorithms that follow are assumed to operate on the normalized testing space $\bar{\mathcal{X}}^D = [0,1]^D$ and normalized performance space $\bar{\mathcal{Y}} = [0,1]$, where the $(\bar{\cdot})$ operator indicates data that has been normalized to this range based on its maximum and minimum possible values. The objective then is to characterize failure modes by identifying a set of scenario clusters $Q_* = \{Q_i\}_{i=1}^{q}$ that are uniquely determined by both the severity of the failure mode and their location within the testing space.

7.2.2 Adaptive Scenario Generation

In the presence of a high-dimensional testing space, it is not feasible to perform an exhaustive set of simulations to identify salient features of the system's performance landscape, especially when limited simulations can be collected due to runtime constraints and the ability of the simulation to be parallelized. Thus, as described in Section 7.1, adaptive scenario generation based on importance sampling can be used to collect high-information data that balances the exploration-exploitation trade-off, ensuring that the simulation set covers the entire testing space while also giving preference to scenarios in areas of interest. Within the context of this analysis, these areas of interest would be unexpected failure modes of the system.

Previous work outlined by Mullins [91, 93, 92] (and contributed to by the author) details an approach for adaptive scenario generation that is extended by this thesis. While the original details are left to previous work [93], the basic paradigm is illustrated in Fig. 7.3, where batches of scenarios are submitted to the SUT, and the resulting performance scores are used to fit a surrogate model $\mathcal{M} : \bar{\mathcal{X}}^D \rightarrow \hat{\mathcal{Y}}$ that predicts performance in untested scenarios. Here, $\hat{\mathcal{Y}}$ represents the predicted normalized performance space. These predictions are used within an acquisition function $L(\bar{\boldsymbol{\eta}})$ to estimate the utility of proposed samples. The scenarios submitted to the SUT during the next iteration of data collection (i.e., the loop of Fig. 7.3) are then chosen with probability proportional to their expected utility [138].

The key pieces of this process that are extended here are the surrogate model and associated acquisition function. The choice of surrogate model is important because autonomous systems can exhibit unpredictable emergent behavior that results in highly nonlinear and/or discontinuous performance (as exampled in Fig. 7.1). Further, the surrogate model must also be able to accommodate high-dimensional testing spaces (i.e., tens of dimensions) and large sample sets. For these reasons, this work adopts quantile random forests (QRFs) [85] as the surrogate modeling technique, although alternative methods such as deep neural networks could be easily substituted. QRFs are well-suited to handle the challenges described above through their non-parametric structure, while also providing prediction bounds that accommodate the uncertainty associated with autonomy performance.

In general, for a response variable Y, predictor variable X, and underlying conditional distribution $\mathbb{P}(Y \leq y | X = \boldsymbol{\eta})$, quantile regression methods not only attempt to estimate the conditional mean, but also the conditional α-quantiles, defined as

$$\mathbb{Q}_{\alpha}(\boldsymbol{\eta}) = \inf\{y : \mathbb{P}(Y \leq y | X = \boldsymbol{\eta}) \geq \alpha\}. \tag{7.1}$$

Quantile random forests achieve estimates $\hat{\mathbb{Q}}_{\alpha}(\boldsymbol{\eta})$ of these α-quantiles using the inherent bagging structure of random forests. In essence, rather than simply storing the weighted mean of all observations for a given leaf node (as is the case with traditional random forests), QRFs extend this by storing the *values* of all observations from each tree to compute an estimate of the full conditional distribution,

$$\hat{\mathbb{P}}(Y \leq y | X = \boldsymbol{\eta}) = \sum_{i=1}^{N} w_i(\boldsymbol{\eta}) \mathbb{I}_{\{Y_i \leq y\}}, \tag{7.2}$$

where w_i are the typical weights calculated when training the random forest and $\mathbb{I}$ is the indicator function (see [85] for details). The α-quantile estimates can then be calculated by using Eq. (7.2) in Eq. (7.1). Most importantly, $\hat{\mathbb{Q}}_{\alpha}(\boldsymbol{\eta})$ can be used to provide confidence bounds on the predictions made by the QRF model. Using a 95% prediction interval, the predictions become

$$[\hat{y}^{lb}, \hat{y}, \hat{y}^{ub}] \mathrel{\widehat{=}} [\hat{\mathbb{Q}}_{0.025}(\boldsymbol{\eta}), \hat{\mathbb{Q}}_{0.5}(\boldsymbol{\eta}), \hat{\mathbb{Q}}_{0.975}(\boldsymbol{\eta})], \tag{7.3}$$

where the superscripts $(\cdot)^{lb}$ and $(\cdot)^{ub}$ represent the lower and upper prediction bounds.

Now that the performance of untested scenarios can be predicted, we next define the acquisition function that estimates the utility of proposed samples. To accommodate high-dimensional testing spaces (large D) and high-volume sample sets (large N), the acquisition function is defined as

$$L(\bar{\boldsymbol{\eta}}) = w_D \left(\frac{\sum_{i=1}^{K} ||\bar{\boldsymbol{\eta}} - \bar{\boldsymbol{\eta}}_i||}{K} \right) + w_F \left(1 - \hat{y}\right) + w_V \left(\hat{y}^{ub} - \hat{y}^{lb}\right), \tag{7.4}$$

where w_D, w_F, and w_V are weights that emphasize exploration of the testing space, failure mode coverage, and collecting data in regions of high performance uncertainty, respectively. The exploration portion of Eq. (7.4) calculates the average distance between a *proposed* sample and its K-nearest *collected* samples. Higher utility is assigned to proposed scenarios with large average distances, as this indicates a region of the testing space that is sparsely sampled. Empirical testing has shown that an approximate value of $K \in [4, 6]$ and values of $w_D = 1$, $w_F = 5$, and $w_V = 4$ provide the desired results for reasonable values of D (i.e., less than 12).

The final term of Eq. (7.4) (collecting data in regions of high performance uncertainty) is important in order to gain confidence in the results of those regions. From a T&E perspective, high performance uncertainty may also indicate a region of large performance gradient. Identifying these large gradients, termed here as performance boundaries, is useful because they contain information regarding critical decisions made by the autonomy, which in turn can lead to the discovery of new failure modes. When compared to purely randomized scenario generation, importance-based methods ultimately produce a sample set that better represents the interesting areas of the performance landscape. This process allows significant insight into the decision-making of the system by isolating the scenarios that result in behavior changes. Such insight is extremely valuable when evaluating black-box systems, particularly those with unpredictable performance surfaces.

7.2.3 Failure Mode Identification

Figure 7.3 shows that the output of the the adaptive scenario generation process is a final sample set S that characterizes the performance landscape of the SUT, meaning it is expected that S contains samples from all areas of the testing space with varying performance scores. It is difficult to draw conclusions from S in its raw form, as it is typically high-dimensional data composed of thousands of scenarios. Thus, in our goal of understanding the failure modes of the system, we employ unsupervised learning that clusters the samples in both (i) the degree of performance deficiency and (ii) the region of the testing space where failure occurs. In this way, each cluster represents a unique failure mode in a form that is more readily digested by the analyst.

7.2.3.1 Performance Space Clustering

The data contained in S is first clustered on the $\mathcal{Y}$ space such that common performance themes can be binned into similar categories. First, S is filtered to remove any samples that have performance scores above the threshold indicating failure scenarios of interest: $\bar{Y}_Q = \{\bar{Y} \mid \bar{Y} < \bar{y}^*\}$. Performance space clustering then operates on the $\bar{Y}_Q$ set.

The algorithm used for unsupervised clustering is dependent on the structure of the data and the fact that the number of clusters is not known a priori. The clustering algorithm implemented for performance space clustering uses Gaussian mixture models (GMMs) fit with the iterative Expectation-Maximization algorithm [82], although alternative algorithms could also be used. The assumption of Gaussian structure to the performance clusters imposes limitations on their shape, but also adds useful analysis properties by giving estimates of the cluster center (mean) and spread (covariance).

A set of j performance clusters is defined as $P_j = \{p_i\}_{i=1}^{j} = GMM(\bar{Y}_Q, j)$, where each p_i is a unique performance cluster and GMM is the clustering operation [82]. To overcome the assumption inherent to GMMs that the number of clusters is known, multiple models are fit, where each assumes a different number of clusters $j \in [1, j_{max}]$. The final performance cluster set is then determined as

$$P_* = \max_{s} \{P_j\}_{j=1}^{j_{max}}, \tag{7.5}$$

where s is the averaged silhouette score [112] of the clustered samples.

7.2.3.2 Testing Space Clustering

The second clustering step takes samples within each performance cluster and further groups them in the $\mathcal{X}^D$ space. This aids in diagnosing regions of the testing space that cause different forms of failure. In other words, similar failure modes belonging to the same performance cluster could occur at multiple different locations within the testing space and it is useful to separate each of these clusters.

Once again, the choice of clustering algorithm is important given the new domain $\mathcal{X}^D$. Imposing structure requirements on the data (such as in GMMs) in the testing space proved to be too limiting. Additionally, there are potentially many more individual clusters within this space, amplifying the challenge that the number, size, and locations of failure modes are not known a priori. Thus, the testing space clustering makes use of density-based methods (specifically DBSCAN [30]) to alleviate some of these problems, which can accommodate an unknown number of clusters and imposes loose requirements on their shape.

Let $\bar{X}_p$ be the scenarios for which their respective performance scores belong to each $p \in P_*$, i.e., $\bar{X}_p = \{\bar{X} \,|\, \bar{Y}_Q \in p\}$. Then, in a similar fashion to the performance clustering step, a set of scenario clusters is calculated as $Q = DBSCAN(\bar{X}_p, \sigma)$, where each $q \in Q$ is now a unique cluster of scenarios, $DBSCAN$ is the clustering operation [30], and σ denotes the clustering algorithm's hyperparameters. Because these hyperparameters significantly affect the clustering process, several cluster sets are calculated over variations to the hyperparameters. The scenario cluster set for a given p is once again that which maximizes the silhouette score over the tested hyperparameters:

$$Q_p = \max_s \{Q\}_\sigma. \tag{7.6}$$

This clustering process is then repeated using scenarios from each $p \in P_*$ such that the final set of failure modes is identified as $Q_* = \{Q_p\}_{p=1}^{p_{max}}$.

7.2.3.3 Feature Scaling

When analyzing realistic SUTs, the well-known curse of dimensionality dilutes the effectiveness of clustering if applied directly to high-dimensional data; thus, feature importance scaling is applied during testing space clustering to reduce the size of the testing space based on the importance of each test parameter. This is achieved by using the QRF surrogate model trained on the final sample set S. The importance β_k of each k-th parameter is determined by how much the out-of-bag mean-squared error of the model changes through different parameter permutations [40]. Normalized parameter importance is given as $\bar{\beta}_k = \beta_k / \sum_{\ell=1}^{D} \beta_\ell$. The normalized importance values are then used as weights to scale the Euclidean distance d between two scenarios $\boldsymbol{\eta}_m$ and $\boldsymbol{\eta}_n$ as:

$$d(\boldsymbol{\eta}_m, \boldsymbol{\eta}_n) = \sqrt{(\boldsymbol{\eta}_m - \boldsymbol{\eta}_n)^T B (\boldsymbol{\eta}_m - \boldsymbol{\eta}_n)}, \tag{7.7}$$

where $B = \text{diag}(\bar{\beta}_1 \dots \bar{\beta}_D)$. This weighted distance function is used during testing space clustering and has the effect of minimizing unimportant test parameters.

7.3 Identifying Performance Regression in Autonomous Systems

The performance analysis of Section 7.2 is capable of producing high-information datasets that evaluate the system's failure modes and statistical performance. This information is highly useful, but without proper context, it can be difficult to determine whether the system is performing as expected. This section thus outlines the second component of the evaluation pipeline in Fig. 7.2: an approach to compare these datasets from a regression standpoint, either datasets generated between versions of the same system, or datasets generated between different systems evaluated on the same parameters.

7.3.1 Problem Setup

Extending the formal problem definition of Section 7.2.1, this section tackles the problem of comparing two different SUTs, $\mathcal{F}_A$ and $\mathcal{F}_B$, where $\mathcal{F}_A \neq \mathcal{F}_B$. Assuming both SUTs are evaluated on the same testing space $\mathcal{X}^D$, we define the change in performance over the entire testing space as $\Delta\mathcal{Y} = \mathcal{Y}_B - \mathcal{Y}_A = \mathcal{F}_B(\mathcal{X}^D) - \mathcal{F}_A(\mathcal{X}^D)$. Given these definitions, we aim to identify the subset of regression regions characterized by a decrease in performance: $\mathcal{X}^- = \{\mathcal{X}^D \,|\, \Delta\mathcal{Y} < 0\}$.

Once again, an exact characterization of the regression regions is intractable; thus, we estimate them using the performance datasets S_A and S_B collected from $\mathcal{F}_A$ and $\mathcal{F}_B$, respectively. While the techniques of Section 7.2 are preferred for generating performance datasets around failure modes of the system, the regression analysis could also be applied to datasets generated from more simplistic randomized Monte Carlo methods. Further, while we assume that both SUTs are evaluated on the same testing space, the resulting studies need not be identical. For the purposes of performance regression analysis, we aim to estimate the regression regions by identifying a set of regression clusters R_* that are uniquely determined by both the severity of the performance decrease between S_A and S_B, and also their location within the testing space.

The overall approach taken to identify and characterize the regression clusters shares many similarities with the methods of Section 7.2, albeit applied to augmented datasets of predicted performance regression. Fig. 7.4 shows a flowchart of the process. The algorithms that follow again operate on the normalized testing space $\bar{\mathcal{X}}^D$ and normalized performance space $\bar{\mathcal{Y}}$.

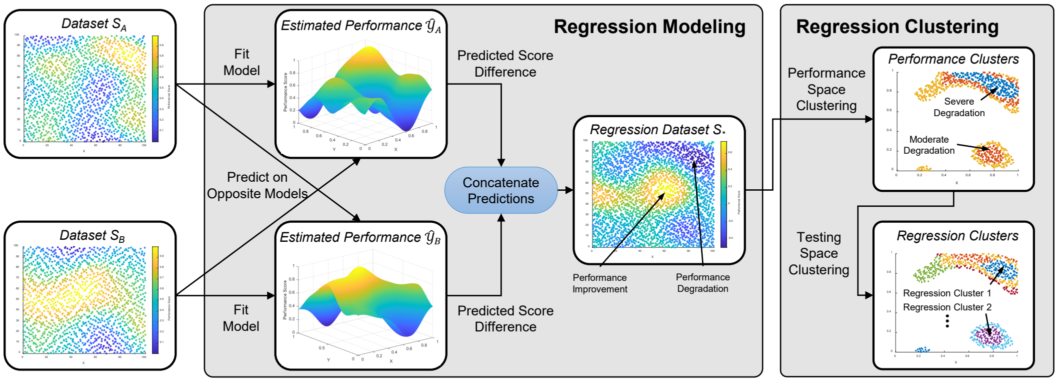


Figure 7.4: Overall performance regression analysis approach. Surrogate modeling is used to normalize each study to the other and predict whether samples have regressed. Unsupervised clustering is then used to extract unique regression clusters in both the severity of the performance decrease and its location within the testing space.

7.3.2 Regression Modeling

Given that $\bar{S}_A \neq \bar{S}_B$, the first step of the performance regression analysis is to once again fit QRF surrogate models $\mathcal{M}_A$ and $\mathcal{M}_B$ to each dataset $\bar{S}_A$ and $\bar{S}_B$, respectively. Regression regions are then estimated based the predicted response of each model to the opposite dataset:

$$[\hat{Y}^{lb}_{AB}, \hat{Y}_{AB}, \hat{Y}^{ub}_{AB}] = \mathcal{M}_B(\bar{X}_A), \tag{7.8}$$

$$[\hat{Y}^{lb}_{BA}, \hat{Y}_{BA}, \hat{Y}^{ub}_{BA}] = \mathcal{M}_A(\bar{X}_B), \tag{7.9}$$

$$\Delta\hat{Y}_A = \hat{Y}_{AB} - Y_A, \tag{7.10}$$

$$\Delta\hat{Y}_B = -(\hat{Y}_{BA} - Y_B). \tag{7.11}$$

Here, $\hat{Y}_{AB}$ and $\hat{Y}_{BA}$ are the predicted scores from studies A and B on models B and A, respectively. The upper and lower prediction bounds are translated to predict the bounds on the performance regression of each dataset as

$$\Delta\hat{Y}_A^{lb} = \hat{Y}_{AB}^{lb} - Y_A, \tag{7.12}$$

$$\Delta\hat{Y}_A^{ub} = \hat{Y}_{AB}^{ub} - Y_A, \tag{7.13}$$

$$\Delta\hat{Y}_B^{lb} = -(\hat{Y}_{BA}^{ub} - Y_B), \tag{7.14}$$

$$\Delta\hat{Y}_B^{ub} = -(\hat{Y}_{BA}^{lb} - Y_B). \tag{7.15}$$

A combined study set $\bar{S}_* = \{\bar{X}_*, \Delta\hat{Y}_*\}$, where $\bar{X}_* = \bar{X}_A \cup \bar{X}_B$ and $\Delta\hat{Y}_* = \Delta\hat{Y}_A \cup \Delta\hat{Y}_B$, is then created to predict the change from $\mathcal{F}_A$ to $\mathcal{F}_B$. The set of samples representing performance regression can be estimated as $\bar{S}_*^- = \{\bar{S}_* \,|\, \Delta\hat{Y}_* < 0\}$. This combined set augments the number of samples ($N_* = N_A + N_B$) for improved fidelity in cluster analysis. Alternatively, the predicted regression bounds $\Delta\hat{Y}_*^{lb} = \Delta\hat{Y}_A^{lb} \cup \Delta\hat{Y}_B^{lb}$ and $\Delta\hat{Y}_*^{ub} = \Delta\hat{Y}_A^{ub} \cup \Delta\hat{Y}_B^{ub}$ could also be used to define the combined study set. Using the lower bound offers a conservative approach and prioritizes extracting regions with the greatest possible regression. The upper regression bound, meanwhile, lets the analysis prioritize performance regression that is occurring with high confidence, meaning that even the upper bound is predicted to have regressed.

7.3.3 Regression Clustering

The final set of regression clusters R_* is determined using the same clustering procedure defined in Section 7.2.3. The primary difference here, however, is that the process is applied to the augmented dataset $\bar{S}_*^-$, which now contains predicted performance change as opposed to the original scores. Ultimately, the identification of both Q_* and R_* allow for a complementary analysis during development. Q_* gives the user insight into the current deficiencies of the system from an absolute perspective, while R_* gives the user a sense of how the system is progressing from a relative perspective.

7.4 Preliminary Analysis on Test Functions

Before applying the performance analysis and regression frameworks to a real autonomous system in Chapter 8, a preliminary evaluation of the surrogate modeling strategy is performed to obtain a baseline of its effectiveness. This evaluation is done using a Monte Carlo study of auto-generated test functions from which ground truth performance landscapes can be calculated. Unsupervised cluster evaluation is omitted because establishing ground truth clusters is much more subjective in nature.

Each randomized test function is meant to be representative of the types of performance surfaces seen in autonomy testing datasets. Specifically, the test functions generate $\mathcal{Y}$ to consist of several "plateaus" within $\mathcal{X}^D$. The quantity, performance score, and location of each plateau are all randomized. The boundary between adjacent plateaus is linearly smoothed and measurement

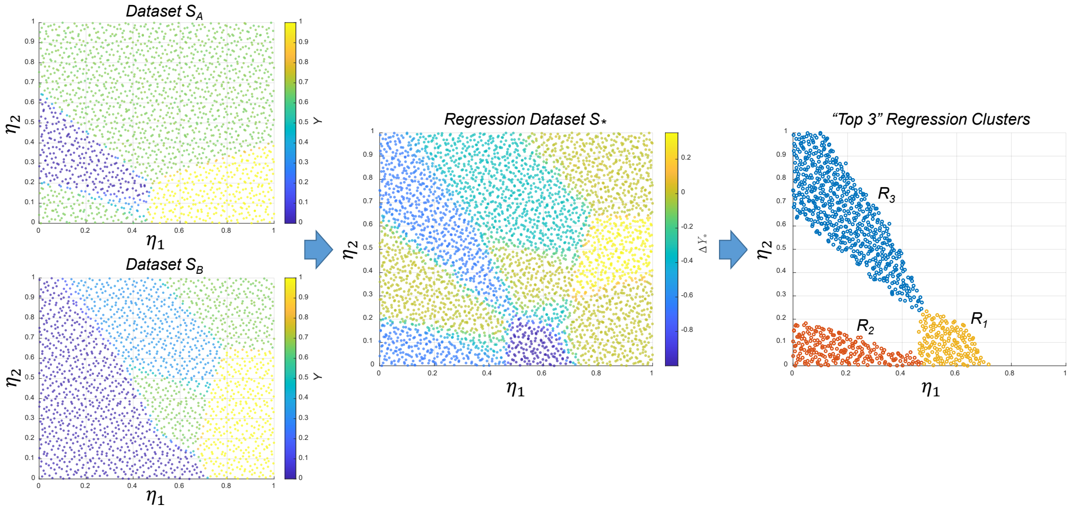


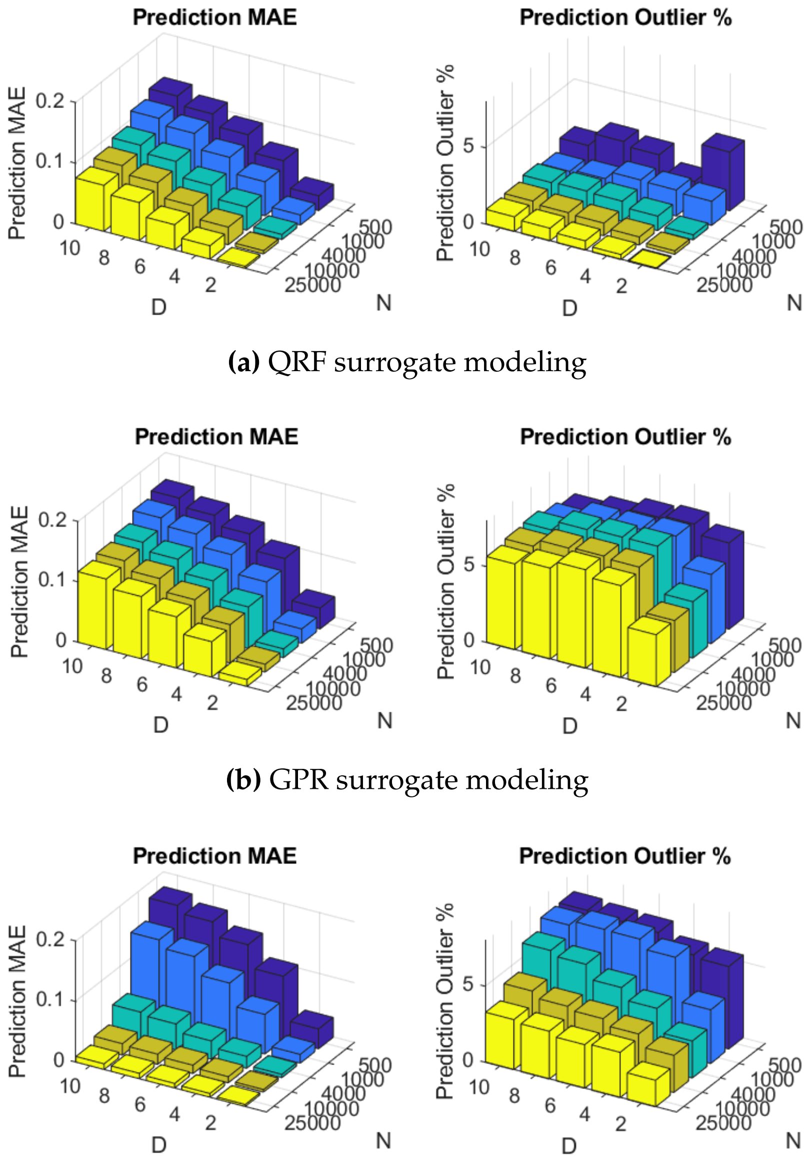
Figure 7.5: Example of the regression analysis applied to two randomly generated test functions (left). The resulting regression dataset and three high-priority clusters are shown by the middle and right plots, respectively.

noise is added to each sample such that $y = y_{truth} + \mathcal{N}(0, \sigma^2)$, where $\sigma = 0.02$. For each study, two test functions are randomly generated and assigned as $\mathcal{F}_A$ and $\mathcal{F}_B$ (examples shown in Fig. 7.5), allowing us to study the modeling performance in the context of the performance regression methodology. The results obtained by predicting performance regression then naturally translate to predicting raw performance (as would be used for adaptive scenario generation) through the operations of Section 7.3.2.

In order to compare different modeling approaches, the QRF model of Section 7.2 was compared against a Gaussian process regression (GPR) model and a five-layer neural network (DNN), all of which used hyperparameters that were optimized to minimize prediction error. The performance of each model is then evaluated across various combinations of D and N to determine scalability for high-dimensional testing spaces and large datasets. Ten studies were run for each combination of D and N (each with new test functions) and the results were averaged.

Figure 7.6 shows the performance of each model in its predictions of the regression dataset S_*. When comparing the predictions to the ground truth regression between pairs of randomized test functions, the first evaluation metric shows the five-fold cross validation mean absolute error (MAE) of $\Delta\hat{Y}_*$. Each model exhibits the expected trend that MAE increases for large values of D and small values of N. The QRF model has lower MAE scores than the GPR model over all combinations of D and N, but is slightly outperformed by the DNN model for large sample sizes, indicating that there may be conditions to apply different modeling techniques.

The second evaluation metric shown in Fig. 7.6 is the percentage of $\Delta\hat{Y}_*$ predictions that lie outside their prediction bounds (i.e., the 95% prediction interval $[\Delta\hat{Y}_*^{lb}, \Delta\hat{Y}_*^{ub}]$). For the GPR and DNN models, the 95% prediction intervals of Eq. (7.8) and (7.9) are calculated as $[\hat{Y}^{lb}, \hat{Y}^{ub}] = \hat{Y} \pm 1.96\sigma$, where the standard deviation σ for the GPR model is calculated through its inherent covariance structure, and for the DNN model is approximated using the model's five-fold cross validation root-mean-square error. The percentage of outliers for all models is roughly consistent with a 95% confidence interval, however, the QRF exhibits the lowest outlier percentage, where typically only $1 - 2\%$ of its predictions lie outside the prediction interval. While the prediction interval is centered on the mean for the GPR and DNN models, the prediction interval of the QRF model is asymmetrical around the mean, which likely results in the increased uncertainty robustness.



(a) QRF surrogate modeling

(b) GPR surrogate modeling

(c) DNN surrogate modeling

Figure 7.6: Regression modeling performance on the prediction of $\Delta\hat{Y}_*$ for randomized test functions.

Overall, the QRF surrogate modeling offers a good balance of reducing MAE while also minimizing the number of predictions that lie outside the 95% predicted regression intervals. It is evident, however, that each surrogate modeling technique may offer some benefits based on the characteristics of the dataset and could be easily substituted into the performance analysis framework.

Chapter 8

Performance Evaluation for Autonomous Surface Vessel Navigation

We now tie Chapters 6 and 7 together by running performance analysis on the proposed good seamanship ASV planning strategy. With regards to T&E of ASV systems, perhaps the biggest issue preventing full adoption is trusting that these systems will behave as expected when operating in environments with other vessels. As detailed in Chapter 5, safely navigating in real-world environments demands that these systems perform a combination of mission completion, general obstacle avoidance, and COLREGS compliance, all of which are objectives that could potentially conflict with each other. To establish trust in the decision-making of an ASV, it is necessary to perform a full statistical evaluation of the system over the range of its normal operating environment using the methods developed in Chapter 7.

The system under test is the 29-ft rigid-hull inflatable boat introduced in Section 6.1 and pictured in Fig. 6.1. All parameters for both planning and evaluation are tuned for this vessel. The values for parameters of the ASV planner not previously derived or defined are given in Table 8.1.

First, the testing spaces and performance spaces are defined with respect to the ASV navigation problem. These components are then applied using the performance analysis of Chapter 7 to evaluate the good seamanship ASV planning strategy. The performance of the planner is compared to other baseline navigation approaches in both single-vessel and multi-vessel avoidance scenarios. We then examine the extracted failure modes of the system and perform a case study on the usefulness of the performance regression framework when attempting to remedy these failure modes. Finally, the results of field experiments in the Chesapeake Bay are described for testing the good seamanship planner over a wide array of COLREGS geometries.

8.1 Analysis Setup

8.1.1 ASV Testing Space Design

The testing space $\mathcal{X}^D$ is the first component that needs to be defined for effective simulation-based T&E. Proper parameterization of the testing space requires careful thought – the goal is to capture all relevant scenarios while minimizing its dimensionality. Here, the application of interest is evaluating ASV decision-making in the context of COLREGS compliance and multi-vessel good seamanship.

Figure 8.1 illustrates the test parameters from which three testing spaces are designed within this context. These parameters capture all geometries of relevant collision encounters with other vessels, as well as the possibility to

Table 8.1: ASV planner parameters used for simulation studies and field experiments.

Parameter	**Description**
$[r_{min}, r_{max}] = [-0.12, 0.12]$ rad/s	Allowable turn rate bounds of the ASV
$[a_{min}, a_{max}] = [-4.0, 1.0]$ m/s^2	Allowable acceleration bounds of the ASV
$L = 4.77 \times 10^{-3}$ nmi	Length of ownship
$\lambda_r = 7.5$	Ship domain radius scaling factor
$\kappa = 10$	Ship domain/arena logistic function shape parameter
$f_0 = 0.5$	Ship domain/arena logistic function shape parameter
$R_{f,a} = 1200$ m	Ship arena fore radius
$R_{s,a} = 1200$ m	Ship arena starboard radius
$R_{a,a} = 650$ m	Ship arena aft radius
$R_{p,a} = 650$ m	Ship arena port radius
$\tilde{\delta}_t = 7$ sec	Minimum time step between branch & bound planner nodes
$\ell = 4$	Number of control actions considered until time of maximum mutual domain risk
$H = 6$	Number of state nodes in branch & bound planner trajectory sequence
$\tilde{\Theta}_C = 0$	Collision index threshold for determining whether target ship is in COLREGS
$\tilde{t}_C = 70$ sec	Time until maximum mutual domain risk threshold for determining whether target ship is in COLREGS
$\tilde{\Theta}_S = 0.5$	Overall risk index threshold used for determining emergency state
$w_M = 1$	Cost function mission weight
$w_S = 15$	Cost function safety weight
$w_C = 3$	Cost function COLREGS weight

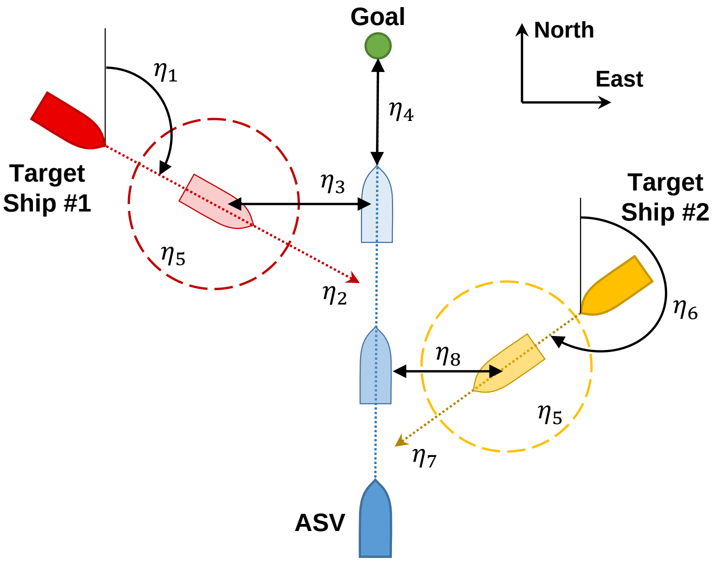


Figure 8.1: Illustration of the test parameters used for each testing space. Ranges and descriptions are given in Table 8.2.

evoke emergent behavior through competing mission objectives and varying compliance of the target ships. For all testing spaces, the ASV starts at a fixed location and travels due North at a nominal speed of 6 m/s towards a goal waypoint (in this sense, π_{nom} is simply the straight line trajectory between the start and goal points). Naturally, there are many more factors that would be of interest in the overall ASV navigation problem than those captured in Fig. 8.1. For example, the locations of static obstacles and environmental effects are all parameters that could influence the behavior of the system. While this thesis focuses primarily on behaviors induced due to different collisions encounters with other vessels, a full test suite should certainly include additional testing spaces to capture these factors.

Table 8.2: Test parameter ranges and descriptions for the ASV testing spaces. The intercept time in the following descriptions is defined as the time at which the target vessel and ASV are collinear along the North-South axis assuming constant speed and heading.

Parameter	**Description**
$\eta_1 \in [0, 360]$ deg	Nominal relative heading for target ship #1
$\eta_2 \in [0, 16]$ m/s	Nominal relative speed for target ship #1
$\eta_3 \in [-200, 200]$ m	Lateral offset of target ship #1 from ASV at the intercept time
$\eta_4 \in [0, 1000]$ m	ASV distance from goal waypoint at intercept time
$\eta_5 \in [0, 400]$ m	Avoidance range used by target ships for their own navigation
$\eta_6 \in [0, 360]$ deg	Nominal relative heading for target ship #2
$\eta_7 \in [0, 16]$ m/s	Nominal relative speed for target ship #2
$\eta_8 \in [-200, 200]$ m	Lateral offset of target ship #2 from ASV at the intercept time

8.1.1.1 Single-Vessel Standard Testing Space

The first testing space on which the good seamanship planner is evaluated involves only the first three test parameters of Fig. 8.1: $\mathcal{X}_{1v}^3 = \mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{X}_3$. Thus, the $\mathcal{X}_{1v}^3$ testing space only generates single-vessel encounters where the target ship maintains constant speed and heading based on the scenario. This simple 3D testing space is meant to evaluate the ASV planner's compliance with COLREGS over a wide array of geometries.

8.1.1.2 Single-Vessel Extended Testing Space

The second testing space extends $\mathcal{X}_{1v}^3$ by also including the fourth and fifth test parameters: $\mathcal{X}_{1v}^5 = \mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{X}_3 \times \mathcal{X}_4 \times \mathcal{X}_5$. The η_4 test parameter modifies how close the collision encounter occurs to the ASV's goal waypoint, introducing a competing priority that could affect its avoidance of the target ship. Additionally, the target ship does not simply maintain constant speed and heading during the encounter. Rather, it follows a simple collision avoidance strategy based on COLREGS-compliant velocity obstacles [58], where the η_5 test parameter varies the target ship's desired avoidance radius.

8.1.1.3 Multi-Vessel Testing Space

The final testing space is designed to evaluate the ASV planner in multi-vessel situations where COLREGS become much more subjective and rule-based planners may not generalize. This 7D testing space, consisting of parameters $\mathcal{X}_{2v}^{7} = \mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{X}_3 \times \mathcal{X}_5 \times \mathcal{X}_6 \times \mathcal{X}_7 \times \mathcal{X}_8$, captures all encounter geometries for two vessels and also varies their desired avoidance radius.

8.1.1.4 Handcrafted Test Scenarios

As a baseline for comparison to the $\mathcal{X}_{1v}^{3}$ testing space, a test suite of standard COLREGS scenarios was generated by a group of experts and former ship captains with extensive knowledge of COLREGS. This test suite consisted of 64 scenarios, shown in Figure 8.2, that exercised the ASV in a select number of COLREGS geometries within $\mathcal{X}_{1v}^{3}$. The purpose of including a set such as this is to show that the conventional approach of handcrafting COLREGS test scenarios, as seen in other research [98, 7, 58], is incomplete for the purpose of fully evaluating ASV decision-making and performance.

8.1.2 ASV Performance Space Design

The second component to be defined is the performance space $\mathcal{Y}$, i.e., the scoring criteria for evaluating ASV performance on scenarios generated from each testing space. It is understood that the criteria used to evaluate appropriate

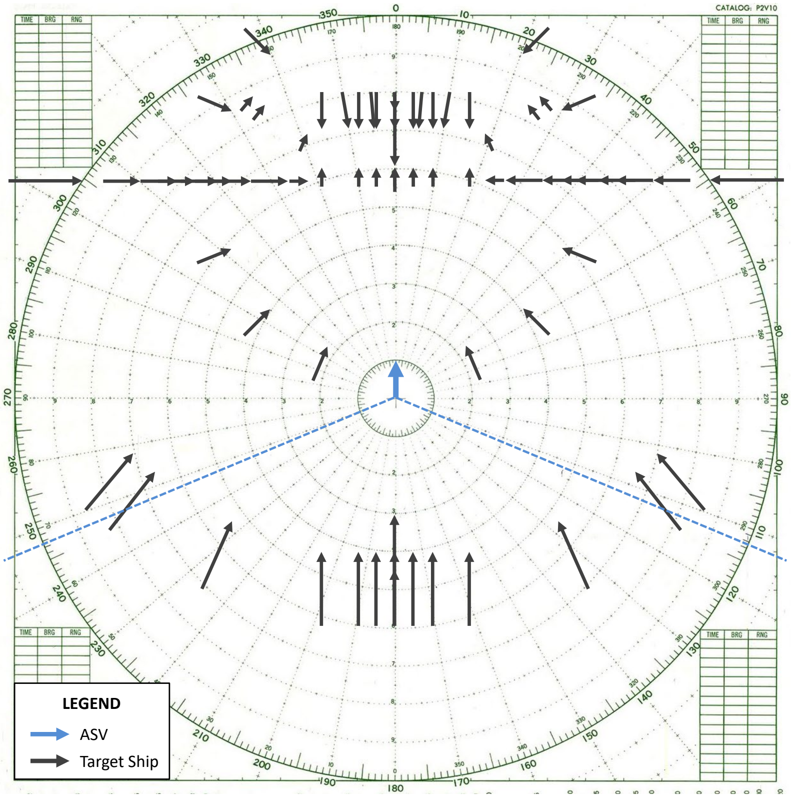


Figure 8.2: Depiction of the handcrafted COLREGS scenarios test suite. Each black arrow designates a separate test scenario, with the length of the arrow representing the relative velocity of the target ship with respect to the ASV.

behavior in an encounter may be subjective. For instance, one ship captain may feel comfortable with small avoidance distances, while another may deem those ranges to be unsafe. Therefore, the methodology below adopts a series of scoring functions that can be tuned to the preference of the user and the ASV under test.

While there have been a significant number of studies into ASV path planning approaches that consider COLREGS, there is much less literature on techniques to evaluate the COLREGS compliance of different planning algorithms. Evaluating whether fielded ASV software is compliant with COLREGS is a challenging problem, where it becomes necessary to perform an objective evaluation based on subjective COLREGS protocols. Tam and Bucknell [127] looked at assessing the collision risk based on the type of COLREGS encounter and the speed of the vessels, however, the analysis was geared more towards path planning as opposed to performance evaluation. COLREGS maneuvering compliance can be thought of as a subcategory to navigational safety assessment that evaluates not just the collision risk of an encounter, but whether the actions of each vessel were appropriate based on the COLREGS protocols. The most complete analysis of quantitative COLREGS evaluation was performed by Woerner et al. [137, 136]. These works define algorithms that evaluate the actions taken by a vessel for each COLREGS rule. This procedure provides a more principled analysis for comparing observed behavior against the expected behavior in different COLREGS scenarios. Woerner's analysis framework was used by Minne [88] to develop a test generation framework for comparing the performance of multiple ASV navigation strategies.

In this thesis, we offer improvements to state-of-the-art methods for evaluating ASV decision-making in the context of both single-vessel encounters, where COLREGS compliance dominates performance evaluation, and multi-vessel encounters that require a more generalized approach to performance evaluation. While the work of Woerner provides a path forward for single-vessel COLREGS encounters, the rule-based nature of the scoring criteria does not generalize well to multi-vessel scenarios. As such, we leverage the good seamanship principles developed in Chapter 6 to create evaluation criteria for multi-vessel encounters as a supplement to existing COLREGS evaluation, where specific COLREGS rule classification loses value due to potentially conflicting rules for each target ship.

A hierarchical approach to evaluation is taken such that a final score is calculated based on multiple layers of sub-scores. This approach is beneficial because the overall performance can be dissected to determine the root cause of a particular performance mode. For example, the safety of an ASV trajectory can be evaluated independently of its COLREGS compliance. In a similar form to the cost function of the ASV planner, the final score is calculated as a weighted combination of performance sub-scores in mission criteria $\mathcal{Y}_M$, safety criteria $\mathcal{Y}_S$, and COLREGS criteria $\mathcal{Y}_C$. While it may seem logical to apply the ASV planner cost function calculations as the performance criteria, there are two primary reasons why slightly different calculations are used: (i) the intent of the performance criteria is to remain independent of the ASV planning strategy such that it can be universally applied to any planner attempting COLREGS-compliant navigation, and (ii) the performance criteria below can be applied to the full, post-processed trajectories of all vessels in the encounter. The second condition means that the scoring criteria has the benefit of hindsight (and ground truth) in evaluating the full encounter and is not limited by a finite planning horizon or uncertainty of the future scenario evolution, as is the case when designing the cost function for the ASV planner. With this in mind, the overall performance space is calculated as

$$\mathcal{Y} = \frac{w_M \mathcal{Y}_M + w_S \mathcal{Y}_S + w_C \mathcal{Y}_C}{w_M + w_S + w_C}, \tag{8.1}$$

where w_M, w_S, and w_C are corresponding weights for the mission, safety, and

COLREGS criteria, respectively. The values for these are the same as those listed in Table 8.1.

8.1.2.1 Mission Score

Since the mission for all testing spaces is simply reaching a desired goal waypoint, the mission sub-score is calculated as the average of the ASV spatial and temporal efficiencies:

$$\mathcal{Y}_M = \frac{1}{2}\left(\frac{t_{nom}}{t_{actual}} + \frac{\rho_{nom}}{\rho_{actual}}\right). \tag{8.2}$$

Temporal efficiency measures a ratio of the nominal time t_{nom} required to reach the goal by following π_{nom} had there been no COLREGS encounter, against the actual time t_{actual} required to reach the goal. Similarly, spatial efficiency measures a ratio of the the length of π_{nom} (ρ_{nom}) against the actual distance traveled ρ_{actual} in the simulation.

8.1.2.2 Safety Score

The safety sub-score is calculated based on the actions taken by the ASV without consideration of specific COLREGS protocols. Thus, this sub-score primarily considers the overall risk $\Phi_S(t)$ associated with the final, post-processed trajectories. As highlighted in Section 6.3, the ship domain geometry used to calculate $\Phi_S(t)$ is better able to capture the risk of an encounter when compared to CPA-based methods, as CPA does not differentiate between geometries that are inherently more risky beyond the distance calculation.

Ideally, for an ASV exhibiting good seamanship, $\Phi_S(t)$ should be kept as close to zero as possible, indicating minimal future collision risk and/or the ASV taking early evasive action. We capture these characteristics by translating the overall risk index into a metric for evaluating seamanship performance over the full encounter. First, let $\Phi_{S,max}$ be the maximum overall risk of the ASV:

$$\Phi_{S,max} = \max_{t \in [t_s, t_f]} \Phi_S(t), \tag{8.3}$$

where t_s and t_f are the start and final times of the encounter, respectively. Also, let $\bar{t} = (t - t_s)/(t_f - t_s)$ be the normalized time vector such that $\bar{t} \in [0, 1]$. Two performance metrics can then be defined that measure the maximum risk (Eq. (8.4)) and the cumulative risk (Eq. (8.5)) over the encounter:

$$\Phi_M = 1 - \Phi_{S,max}, \tag{8.4}$$

$$\Phi_C = 1 - \int_0^1 \frac{\Phi_S(\bar{t})}{\Phi_{S,max}} \, d\bar{t}. \tag{8.5}$$

The calculation of Φ_C represents a score for the amount of risk acquired during the encounter, normalized by the maximum risk. Thus, a value of $\Phi_C = 0$ would mean that the ASV maintained a risk value of $\Phi_{S,max}$ for the entirety of the encounter. This metric is able to capture sustained periods of high risk as well as indecision that leads to multiple spikes in the risk.

Finally, the safety sub-score is calculated as a combination of Φ_M and Φ_C, both of which constitute good seamanship behaviors and risk reduction:

$$\mathcal{Y}_S = \Phi_M \big(1 + \alpha(2\Phi_C - 1)(1 - \Phi_M)\big). \tag{8.6}$$

The form of Eq. (8.6) rewards short periods of low risk and penalizes sustained

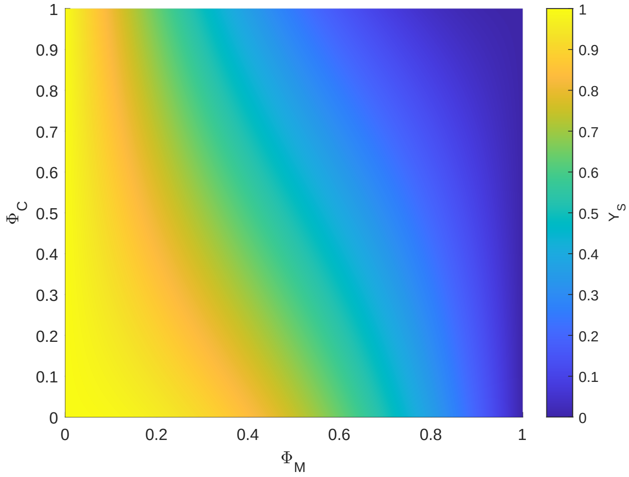


Figure 8.3: The $\mathcal{Y}_S$ performance surface resulting from Eq. (8.6).

periods of high risk, where α serves as a tuning parameter that controls the amount of penalty based on the value of Φ_C (this work uses a value of $\alpha = 0.75$ based on empirical testing). The performance surface of $\mathcal{Y}_S$ based on different values of Φ_M and Φ_C is shown in Fig. 8.3.

8.1.2.3 COLREGS Score

The final component $\mathcal{Y}_C$ of Eq. (8.1) is meant to specifically quantify COLREGS compliance as opposed to more general safety principles.

8.1.2.3.1 Single-Vessel COLREGS Scoring: While there are fairly obvious metrics that can be used with respect to mission and safety criteria, there is not currently an accepted framework for evaluating COLREGS compliance. Nevertheless, the most complete analysis of COLREGS scoring to date is presented by Woerner [137, 136] and we adopt this scoring convention here for single-vessel encounters, such as those generated in $\mathcal{X}_{1v}^3$ and $\mathcal{X}_{1v}^5$. The work by Woerner puts forth an algorithmic approach that quantifies the compliance of an ASV trajectory with respect to each applicable COLREGS rule. For single-vessel encounters, this is an appropriate approach, as the ASV should comply as closely as possible with the protocols as they were intended.

8.1.2.3.2 Multi-Vessel COLREGS Scoring: As discussed in Chapter 5, many of the COLREGS protocols were written with single-vessel encounters in mind, and their applicability to multi-vessel scenarios is debated. The Woerner scoring algorithm [137, 136] proposes to maintain the same scoring methodology used for single-vessel encounters for each target ship in a multi-vessel encounter. When COLREGS rules conflict with each other, Woerner proposes that user-supplied priorities should be assigned to each vessel and COLREGS rule. In this thesis, we adopt an alternative approach that does not apply single-vessel scoring to a multi-vessel encounter. Instead, the COLREGS score for multi-vessel encounters is determined using the same procedure as Alg. 5 such that $\mathcal{Y}_C = Q_C$. This approach relies on the risk-based evaluation of Eq. (8.6) to capture adherence to good seamanship principles, while additional considerations such as obvious maneuvering and minimizing action indecision are captured through Q_C. The primary difference here is that, as opposed to calculating Q_C over a receding planning horizon, the calculation is performed over the post-processed encounter defined over $\bar{t}$.

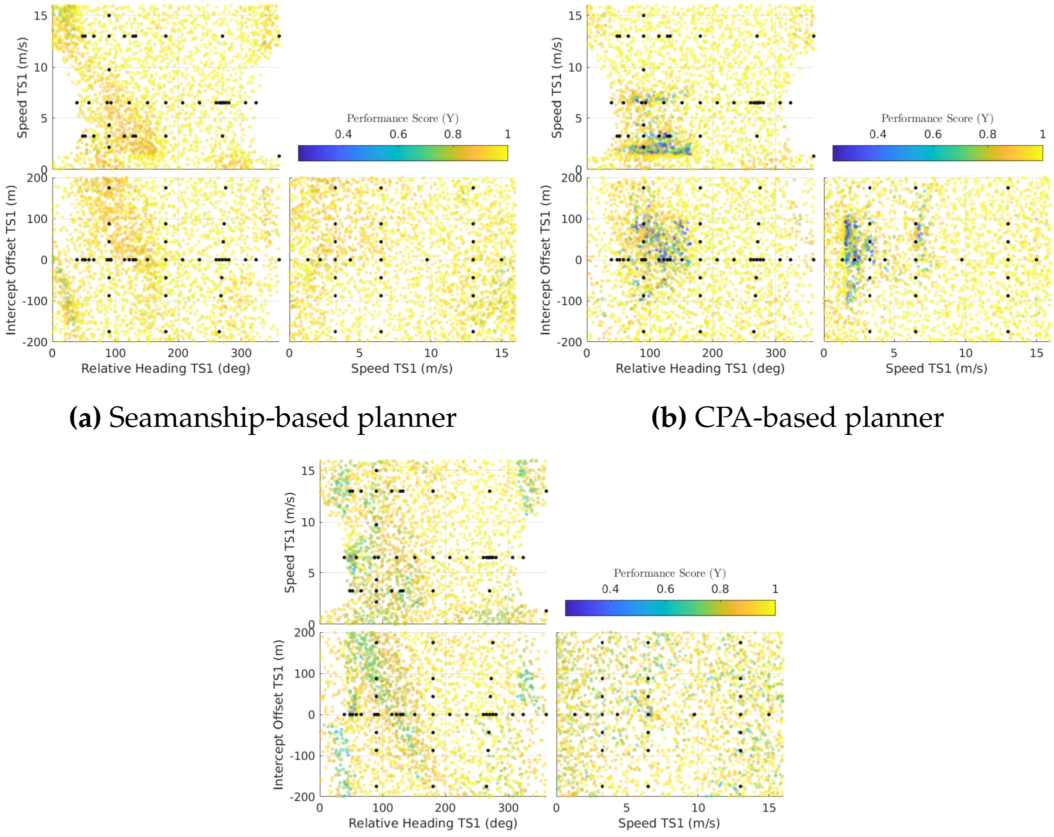
8.2 Performance Evaluation Results

We now transition to actually applying the performance evaluation described in Chapter 7 to the ASV planner of Chapter 6, using the 29-ft rigid-hull inflatable boat of Section 6.1 as the SUT. This section focuses on analyzing performance-based trends from simulation data with the intent of discovering the unique failure modes of the system. Section 8.3 then provides field test examples of the planner behavior in candidate scenarios.

For the analysis of this section, scenarios were adaptively-generated using the methodology of Section 7.2. A total of 3,000 scenarios were simulated on the $\mathcal{X}_{1v}^{3}$ testing space, 7,000 scenarios on the $\mathcal{X}_{1v}^{5}$ testing space, and 14,000 scenarios on the $\mathcal{X}_{2v}^{7}$ testing space. The threshold for performance scores of interest was set to $y^* = 0.9$, representing the approximate 20-th score percentile of the seamanship planner.

8.2.1 Comparison to Baseline Planning Strategies

We first evaluate the good seamanship ASV navigation strategy in simulation against two more traditional planners: (i) a variant of the good seamanship planner that utilizes CPA-based avoidance criteria / costs as opposed to ship domain-based criteria, and (ii) a COLREGS-supplemented velocity obstacles (VO) planner (similar to the approach by Kuwata et al. [58]). Figure 8.4 shows each planning strategy's performance landscape (projected onto pairs of 2D subspaces) for the studies performed on $\mathcal{X}_{1v}^{3}$, where each point represents a simulation and corresponding performance score. The black dots represent the handcrafted scenarios of Fig. 8.2 as a baseline for comparison.



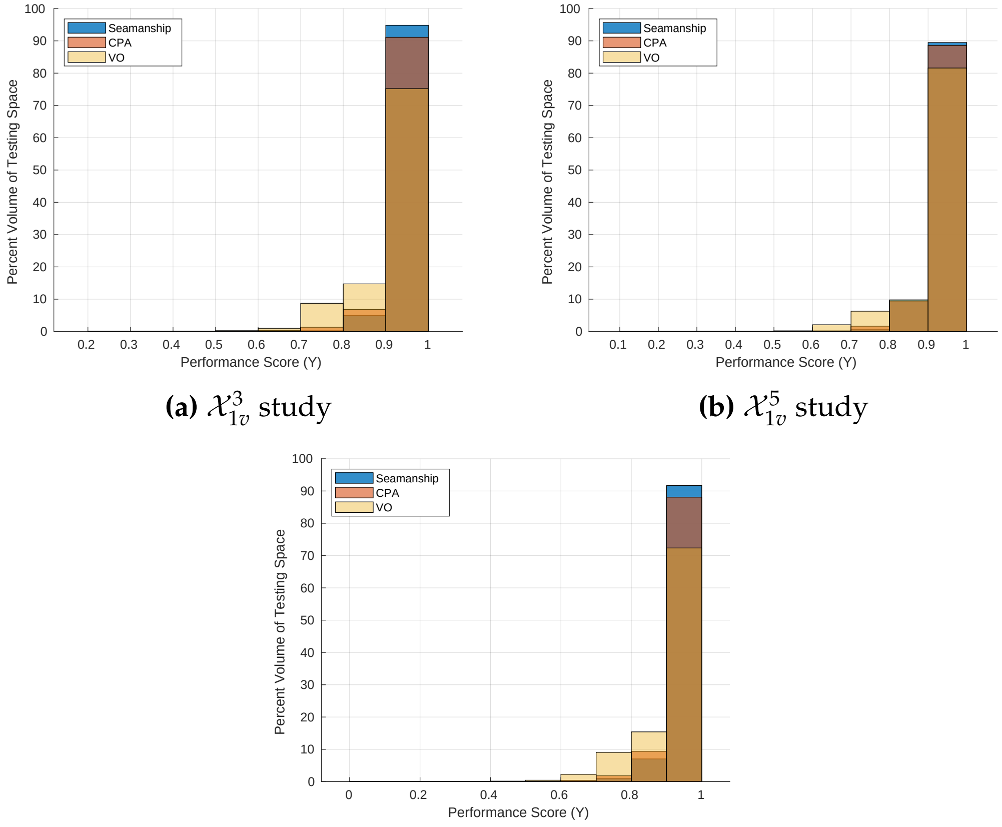
(a) Seamanship-based planner

(b) CPA-based planner

(c) VO-based planner

Figure 8.4: Performance landscapes of each planning strategy calculated from 3000 adaptively-generated scenarios on the $\mathcal{X}^3_{1v}$ testing space. The color of each scenario represents the performance scores $\mathcal{Y}$, while the black dots represent the set of handcrafted scenarios (introduced in Fig. 8.2) as a baseline for comparison.

Qualitatively, it is evident that the proposed seamanship-based planning strategy has significantly fewer and less severe failure modes when compared to the CPA-based and VO-based planners. Additionally, the resulting performance landscapes show that handcrafted test scenarios only cover a small portion of the testing space and reveal limited information about the overall system performance. Conversely, adaptive test generation is able to provide a more complete picture, producing parameter combinations that highlight



(a) $\mathcal{X}_{1v}^{3}$ study

(b) $\mathcal{X}_{1v}^{5}$ study

(c) $\mathcal{X}_{2v}^{7}$ study

Figure 8.5: Histograms for each planner representing the percent volume of the testing space occupied with respect to different scoring bins of $\mathcal{Y}$.

regions of deficient performance with much higher resolution.

Since it is difficult to visualize the performance landscapes of the higher dimensional testing spaces in scatter plot form, we examine the performance statistics of each study in Fig. 8.5. It is not appropriate to simply examine the raw distribution of scores collected from each study, as the adaptive nature of the sample collection produces a distribution that is biased towards the failure modes of the system. Thus, each bar in the histograms of Fig. 8.5 represents the volume of $\mathcal{X}$ characterized by a particular range of $\mathcal{Y}$ as a percentage of the total volume of $\mathcal{X}$. In other words, each bar is calculated as

$$b = 100 * \frac{Vol(\mathcal{X} \,|\, y^- < \mathcal{Y} \leq y^+)}{Vol(\mathcal{X})}, \tag{8.7}$$

where y^- and y^+ are the lower and upper scores of the bin edges, respectively. The values of $Vol(\mathcal{X} \,|\, y^- < \mathcal{Y} \leq y^+)$ are estimated by summing the Voronoi volumes of scenarios that have performance scores within the bin edges. Viewing the performance distribution as a percentage of volumes removes the bias introduced by the varying sample densities throughout the testing space.

These results show that the seamanship-based planner outperforms the other planners by achieving high performance scores in the vast majority of all testing spaces. The seamanship-based planner also minimizes lower performance scores when compared to the other planners. Table 8.3 summarizes some of these metrics for additional comparison. Overall, the seamanship-based planner has the smallest testing space volume below y^* for all studies, and also has higher minimum scores than the other planners for all studies. The metrics summarized in Table 8.3 also show that the seamanship-based planner had the lowest collision rate, and was able to avoid collisions in all single-vessel encounters, even when the target ship was noncompliant in obeying COLREGS.

8.2.2 Failure Mode Introspection

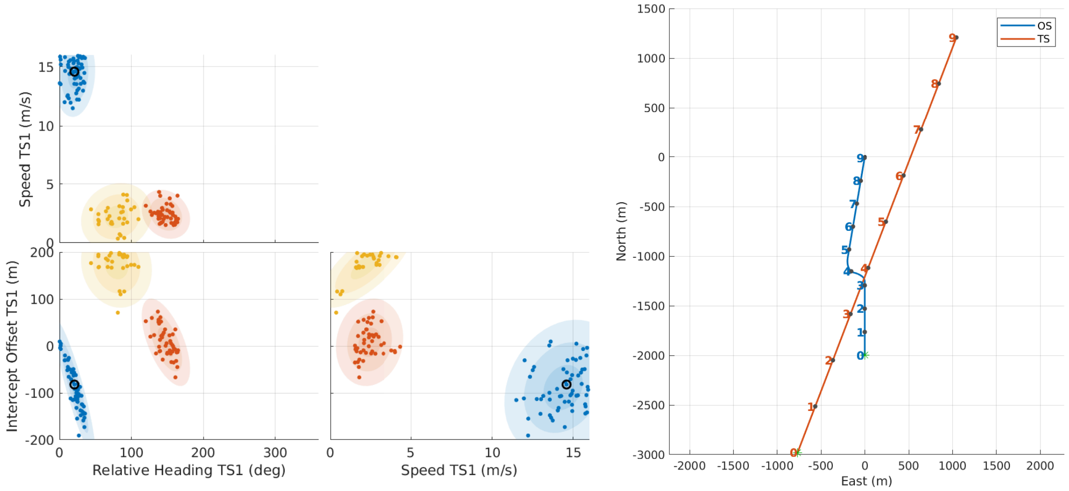
While the good seamanship planner offers improved performance when compared to other baseline planning strategies, there are still challenging test parameter combinations that stress the decision-making and lead to regions

Table 8.3: Performance metrics for each ASV planning method.

Planner		Seamanship	CPA	VO
$\mathcal{X}_{1v}^{3}$ **Study**	$Vol(\mathcal{X} \mid \mathcal{Y} < y^{*})$ %	5.0	8.7	24.6
	Minimum Score	0.71	0.23	0.32
	Collision %	0	0.075	0.030
$\mathcal{X}_{1v}^{5}$ **Study**	$Vol(\mathcal{X} \mid \mathcal{Y} < y^{*})$ %	10.5	11.2	18.4
	Minimum Score	0.59	0.23	0.19
	Collision %	0	0.023	0.005
$\mathcal{X}_{2v}^{7}$ **Study**	$Vol(\mathcal{X} \mid \mathcal{Y} < y^{*})$ %	8.2	11.8	27.6
	Minimum Score	0.13	0.11	0.05
	Collision %	0.003	0.016	0.095

of lower performance. We now wish to do a further introspection into these primary failure modes of the good seamanship planner.

For the $\mathcal{X}_{1v}^{3}$ testing space, Fig. 8.6 shows the "top 3" failure modes from the Q_{*} set (i.e, the three clusters that exhibited the lowest average scores) that were extracted using the methods of Section 7.2.3. The most significant failure mode, Q_1, is displayed in blue. This failure mode is characterized by a high-speed, noncompliant target ship that is overtaking ownship on a nearly parallel course. An example of the ASV behavior from a scenario near the centroid of this failure mode is shown in Fig. 8.6b, where the trajectories of ownship (OS) and the target ship (TS) are stamped with numbers between "0 – 9" representing equivalent time points. This scenario is particularly challenging because ownship is required to stand-on until the target ship is deemed noncompliant, occurring at approximately the "3" time stamp. At this point, the geometry of the scenario produces a decision-making boundary between performing an emergency avoidance maneuver to either port or



(a) "Top 3" failure clusters from Q_* **(b)** Failure scenario from Q_1.

Figure 8.6: (a) The "top 3" failure clusters from the Q_* set (Q_1 = blue, Q_2 = red, Q_3 = gold). The ellipses display each cluster's covariance structure and the the black circle highlights the scenario shown in (b).

starboard. Ownship ultimately decides that an avoidance maneuver to port minimizes the cumulative risk by getting out of the target ship's path more quickly than would be the case with a maneuver to starboard. The port maneuver, however, requires ownship to slightly cross the target ship's bow, which is less than ideal and lowers the performance score of the scenario. The end result is that ownship is able to avoid a collision with the unsafe target ship, but the minimum distance between the vessels is too close due to the high speed of the target ship.

The difficulty of the scenario geometries seen in Q_1 of $\mathcal{X}_{1v}^3$ is further exacerbated in multi-vessel scenarios. In examining the primary failure mode of the $\mathcal{X}_{2v}^7$ testing space (one that is characterized by a violation of the collision radius between ownship and one of the target ships), we see that once again this mode is dominated by high-speed, noncompliant target ships that are

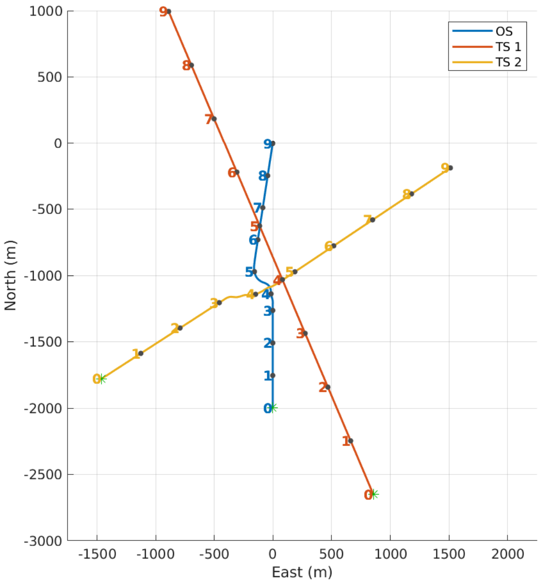


Figure 8.7: Example scenario from the primary failure mode of the $\mathcal{X}^{7}_{2v}$ testing space, characterized by a violation of ownship's collision radius by TS2.

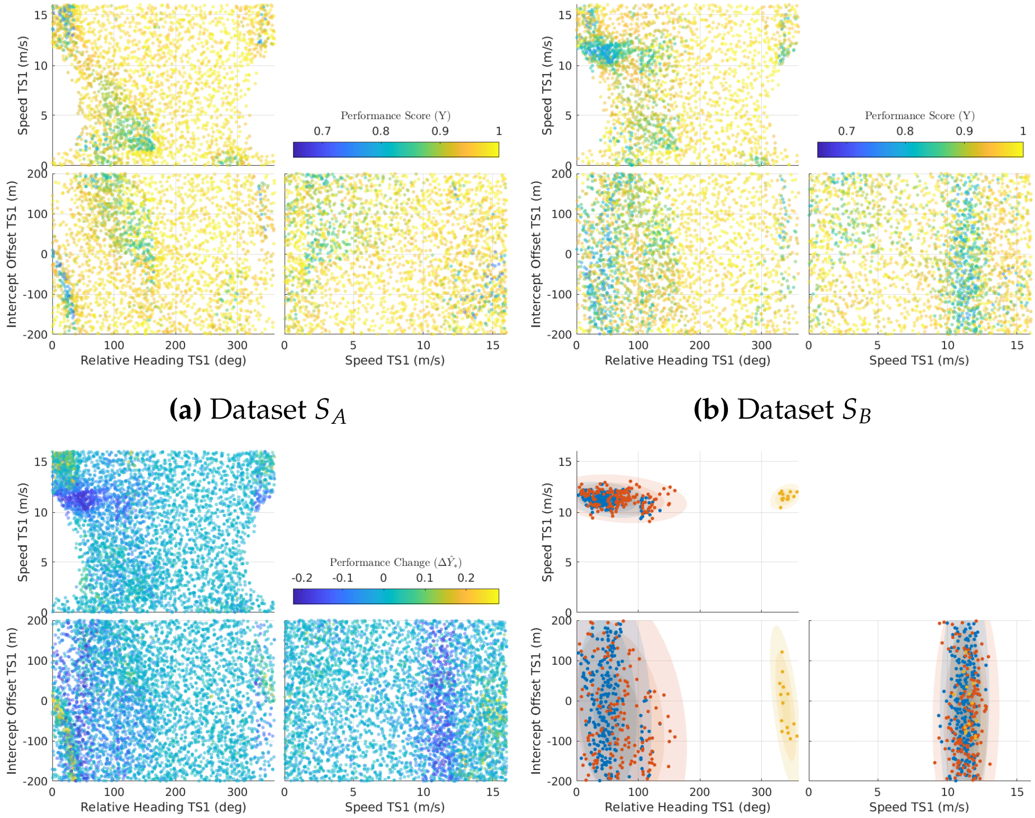
overtaking ownship. Figure 8.7 shows an example of how these conditions nearly lead to collision just after the "4" time stamp. Ownship is now being overtaken by two target ships from opposite sides, both of which have speeds of approximately 12 m/s and a desired avoidance radius of only 25 m. The short avoidance radius of the target ships results in a late maneuver by TS2 just after the "3" time stamp. This maneuver, however, does not open enough range between TS2 and ownship. At approximately the same time, ownship deems both target ships to be noncompliant. Because ownship is restricted on both sides, it reduces speed to avoid a collision (indicated by the shorter distance between the "3" and '4" time stamps of ownship's trajectory). The combination of the late avoidance maneuver by TS2 and the speed reduction of ownship puts the two vessels on a collision course at close range. As a final effort to avoid collision, ownship increases back to the nominal speed and tries to maneuver away from the paths of TS2 and TS1. While this successfully avoids a true physical collision, the minimum range between ownship and TS2 is only 35 meters, a range that is much too close and results in a very low performance score of $y = 0.14$. Although this scenario of two noncompliant target ships is unlikely to occur in a real environment, the knowledge of this failure mode is extremely valuable such that future iterations of the planner can improve performance under these conditions.

8.2.3 Performance Regression Analysis Case Study

Now that the primary failure modes of the good seamanship planner have been identified, this section presents a case study into applying the performance regression framework of Section 7.3; specifically, how this analysis can be leveraged to track any adverse changes in performance after introducing software updates. This case study focuses on the $\mathcal{X}_{1v}^{3}$ testing space due to its low-dimensional interpretability.

The good seamanship planner presented thus far serves as the baseline software version (deemed version A). This software version produces the performance dataset S_A shown in Fig. 8.8a. In order to fix the Q_1 failure mode discovered in Section 8.2.2, a software update was proposed that would allow the ASV to increase its speed above the nominal value during emergency situations. In other words, during emergencies, the speed primitives were updated to be chosen according to

$$\delta_u \in \begin{cases} \{-\frac{1}{2}u_{nom}, 0, 2u_{nom}\} & \text{if } n = 1, \\ \{0\} & \text{otherwise.} \end{cases} \tag{8.8}$$



(a) Dataset S_A

(b) Dataset S_B

(c) Regression dataset S_*

(d) “Top 3” regression clusters from R_*

Figure 8.8: Case study results of applying the performance regression analysis to the $\mathcal{X}^3_{1v}$ testing space of the good seamanship planner.

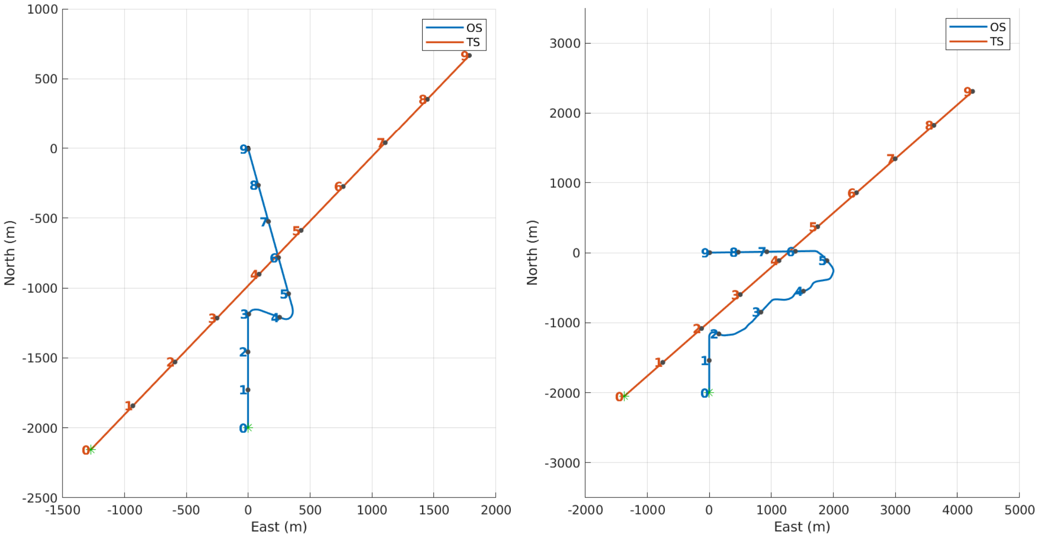
This proposed update would allow the ASV to decrease the relative velocity between itself and the target ship in the overtaking scenarios of Q_1 characterized by high-speed target ships. A smaller relative velocity would then give ownship additional time to reduce the collision risk of the encounter. This proposed update was implemented as version B and produced the performance landscape shown in Fig. 8.8b.

The regression analysis produces the regression dataset S_* in Fig. 8.8c, which predicts changes in performance $\Delta\hat{Y}_*$ over the $\mathcal{X}^3_{1v}$ testing space. The five-fold cross validation MAE of $\Delta\hat{Y}_*$ was 0.015 and the percentage of samples outside the prediction bounds was 2.9% (using 95% confidence intervals). Both of these measures indicate that the regression modeling performance on real autonomy datasets is consistent with the results seen on the test functions of Section 7.4.

Qualitatively, Fig. 8.8c shows that the software changes from version A to version B did in fact improve performance in the Q_1 scenarios where ownship is being overtaken by a high-speed target ship (displayed as the yellow region indicating an increase in performance). However, these software changes also produced unexpected performance regression in other areas of the testing space. The R_* set is shown in Fig. 8.8d, representing the "top 3" regression clusters in terms of severity and volume. While R_1 and R_2 (shown in blue and red, respectively) overlap in the testing space, they are unique clusters within the performance space, meaning that R_1 is characterized by a more severe performance regression than R_2.

Further introspection of the scenarios comprising the R_1 cluster reveals that performance has regressed in scenarios where the target ship is on the border between overtaking and crossing ownship at speeds between 11 – 12 m/s. Figure 8.9 shows a pair of representative scenarios from R_1, one from each S_A and S_B. In version A, the ASV maintains its nominal speed and makes an emergency maneuver to starboard. Once the target ship passes, the ASV resumes travel towards its goal waypoint.

After the software change, however, the ASV now has the ability to increase speed in this emergency situation. The evasive plan again maneuvers



(a) Version A

(b) Version B

Figure 8.9: Example scenarios of the primary R_1 regression cluster. After the software updates, scenarios within the R_1 cluster show that the ASV is susceptible to being herded when traveling on a similar velocity vector to the target ship.

to starboard, but also increases the speed of ownship as evidenced by the increased spacing between time stamps “2 – 6”. The resulting velocity vector of ownship becomes nearly identical to that of the target ship, putting them on parallel courses with similar speeds. This state leads to a decision-making edge case for the planner between attempting to pass ahead of the target ship, and performing a full circle in what is colloquially known as a “paperclip” maneuver. The choice between these decisions produces the oscillations seen between time stamps “2 – 5” as ownship is herded along the target ship trajectory. A herding condition such as this is not uncommon among receding horizon planners when the similar relative velocities of the vessels cause the scenario to evolve beyond the planning horizon.

The performance regression identified by R_1 encompasses a small region of the testing space, but it is extremely valuable to have knowledge of this new ASV behavior. This section acted as a case study, as it is currently not desired for the planner to travel above the nominal mission speed. However, if this requirement were to change in the future such that higher speeds were acceptable, then additional software updates would need to be made in subsequent releases to prevent this type of herding failure.

8.3 Field Experiments on Candidate Scenarios

After examining the performance of the good seamanship planner in simulation, field experiments with the 29-ft autonomous rigid-hull inflatable boat introduced in Section 6.1 (and again shown in Fig. 8.10) were performed to both demonstrate the proposed planning method on hardware and provide confidence to the simulation analysis. All experiments were performed in the Chesapeake Bay within the operating area shown in Fig. 8.10. A safety pilot was on board the vessel at all times to take control in the event of an emergency. A total of 14 scenarios were tested on the water to capture a diverse array of different COLREGS scenarios under both single-vessel and multi-vessel encounters. This section will examine a subset of the most representative scenarios from the field tests.

To replicate the simulations as closely as possible, the experiments utilized virtual target ships that were corrupted with state noise to mimic characteristics of the true perception system. Maritime perception for autonomous operations is its own challenge that is beyond the scope of this thesis. Not only must the perception system reliably detect other vessels on the water

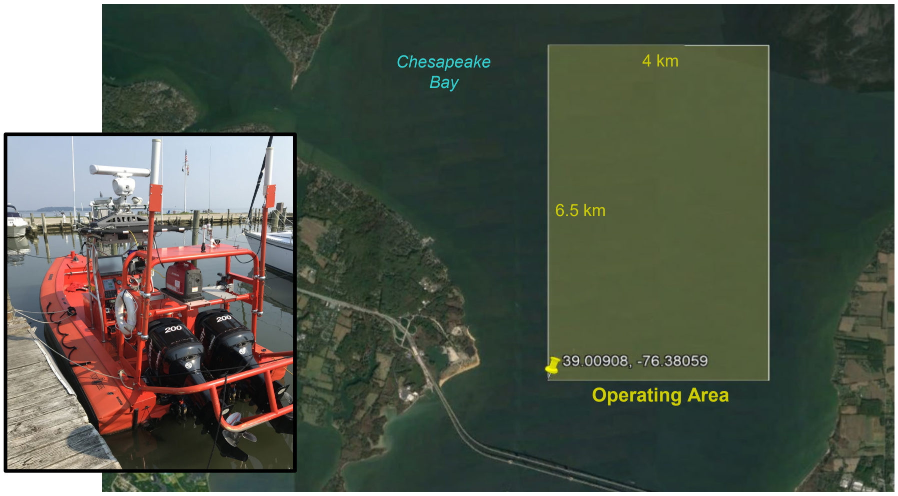


Figure 8.10: Field tests with a 29-ft autonomous RHIB were performed in the Chesapeake Bay to evaluate the good seamanship planner on an operational platform.

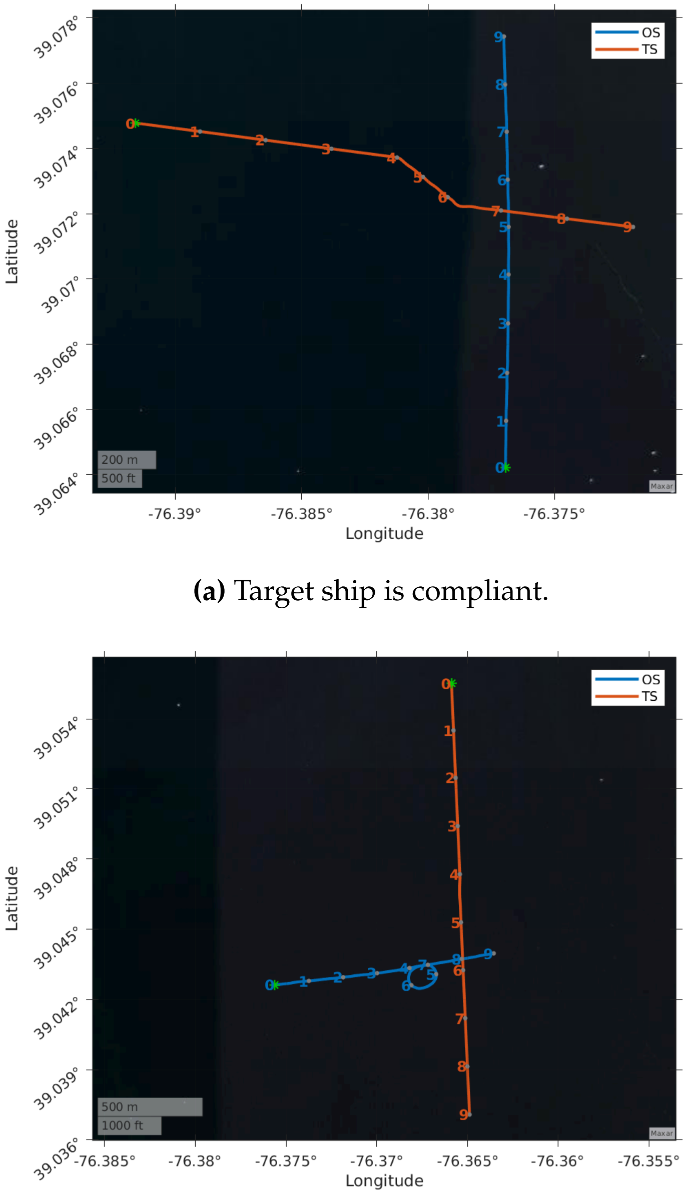
under different environmental conditions, but for proper COLREGS compliance, it must also be able to classify the type of vessel as well. Thus, virtual target ships offer an effective alternative that isolates the testing of the good seamanship planner, while also facilitating greater testing throughput.

8.3.1 Single-Vessel Scenarios

8.3.1.1 Crossing Stand-on – Fig. 8.11

The ASV is expected to stand on while the target ship gives way in this crossing scenario. In Fig. 8.11a, the target ship gives way and ownship properly maintains course and speed.

A second variation of this scenario, however, was repeated in Fig. 8.11b with a noncompliant target ship that did not attempt evasive action. Ownship correctly maintains course and speed until the target ship is deemed noncompliant after the "4" time stamp, which then requires the ASV to take its own



(a) Target ship is compliant.

(b) Target ship is noncompliant.

Figure 8.11: Crossing scenario where ownship is expected to stand on.

evasive action. The resulting avoidance maneuver resembles the standard "paperclip" maneuver taught to ship captains for precisely this situation. When *in extremis*, COLREGS prohibit turning to port because the vessels would then be on a collision course if the target ship gave way as it should. The appropriate maneuver then is to reduce collision risk by turning to starboard, continuing to monitor the situation, and performing a full roundabout to de-escalate the situation if necessary. This result falls naturally out of the seamanship-based planner without the need for a preplanned maneuver or rule-based heuristic to account for this situation.

8.3.1.2 Offset Head-on – Fig. 8.12

It is highly preferred in COLREGS for vessels to pass port-to-port when meeting on reciprocal courses (as would be captured by the decentralized ellipse ship domain). The scenario shown in Fig. 8.12 is difficult because the vessels approach from each other's starboard side, meaning that they must cross each other's bow in order to pass port-to-port. While "correct" maneuvering for these offset head-on edge cases is debated, for small offsets such as this scenario, it is generally accepted that the vessels should attempt to pass port-to-port if maneuvers are made early enough that crossing ahead of each other poses little risk. The ASV properly exhibits this behavior here, and even provides enough clearance to account for the noncompliance of the target ship.

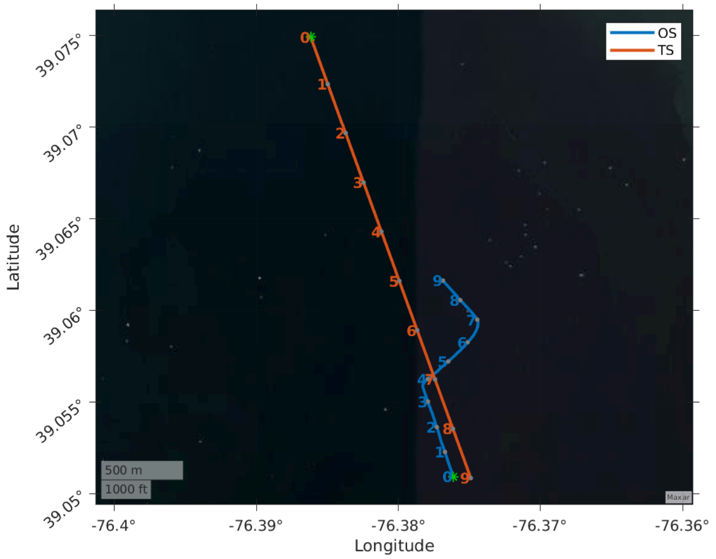


Figure 8.12: Offset head-on scenario where both vessels are expected to give way.

8.3.1.3 Overtaking – Fig. 8.13

The ASV is expected to give way in this scenario as it overtakes a low-speed target ship. As opposed to other COLREGS situations that dictate a strong preference for evasive maneuvers to starboard, it is generally acceptable to overtake a vessel on either side (whichever is safer). This scenario highlights how the ship domain lateral equalizer of Eq. (6.15) accounts for this nuance and allows the ASV to overtake with a maneuver to port. In this case, overtaking with a maneuver to port is highly preferred because it results in ownship passing astern of the target ship.

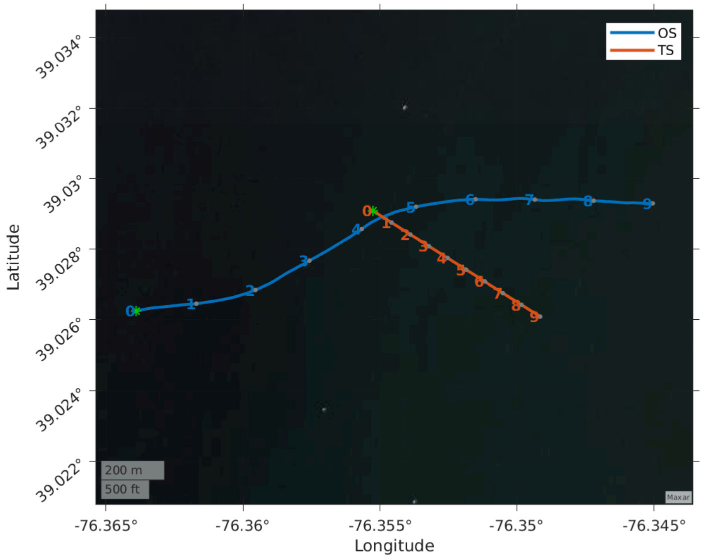


Figure 8.13: Overtaking scenario where ownship is expected to give way.

8.3.2 Multi-Vessel Scenarios

8.3.2.1 Conflicting Crossing Expectations – Fig. 8.14

This scenario features a roundabout geometry where, according to single-vessel COLREGS protocols, the ASV would be expected to give way to TS2 while somehow also standing on to TS1. With these conflicting expectations, the seamanship-based approach is still able to produce an avoidance maneuver that minimizes risk by prioritizing its responsibility to avoid TS2 off its starboard bow through the geometry of each ship domain and arena. The resulting maneuver correctly treats the situation as a roundabout where each vessel is expected to proceed counterclockwise around the encounter. Due to the shorter avoidance radius of the target ships in this scenario, TS1 does not give way because of the early action by ownship. TS2 then performs a small maneuver to starboard to fulfill its give way expectation with regards to TS1.

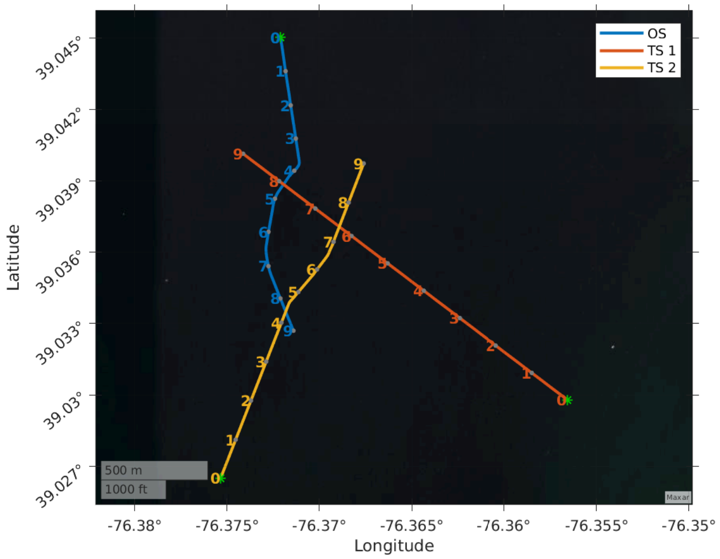


Figure 8.14: Scenario with two vessels crossing from opposite directions and conflicting COLREGS expectations.

8.3.2.2 Double Crossing Give-way – Fig. 8.15

The ASV is expected to give way to two different vehicles that are both crossing in the scenario shown in Fig. 8.15. The difficulty in this scenario lies in the high speed of TS1 coupled with its shallow crossing angle, one that lies near the threshold of TS1 actually overtaking the ASV. From a COLREGS and good seamanship perspective, course changes are preferred over speed changes when giving way because they are more readily perceivable by the other vessels in the situation. Given the approach angle of TS1, the planner deems it necessary to both reduce ownship's speed while also performing a COLREGS-compliant (i.e., readily apparent) heading change between time stamps "2 – 5". The additional speed reduction of the trajectory is necessary to appropriately decrease the collision risk of the encounter. The ASV then

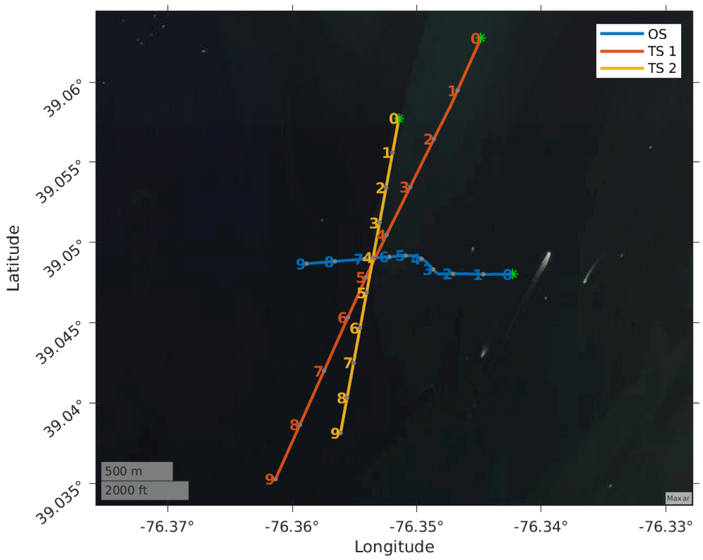


Figure 8.15: Scenario with two vessels crossing where ownship is expected to give way to both vessels.

resumes progress towards the goal after the situation is clear.

8.3.2.3 Conflicting Overtaking Expectation – Fig. 8.16

The scenario shown in Fig. 8.16 features another geometry with conflicting expectations when COLREGS are considered for each individual vessel. Ownship must give way as it overtakes TS1, but it is also expected to stand on as it is being overtaken by TS2. If this were a single-vessel scenario only involving TS1, then it is likely that the ASV would exhibit similar behavior to that of Fig. 8.13 by passing astern of TS1. With the addition of TS2, a maneuver to port would introduce additional risk to the scenario by encroaching on the path of TS2. Therefore, the planner overtakes TS1 on its starboard side, which then allows TS2 to overtake TS1 on its port side. The downside of this maneuver is that it requires ownship to then cross the bow of TS1; however, the trajectory

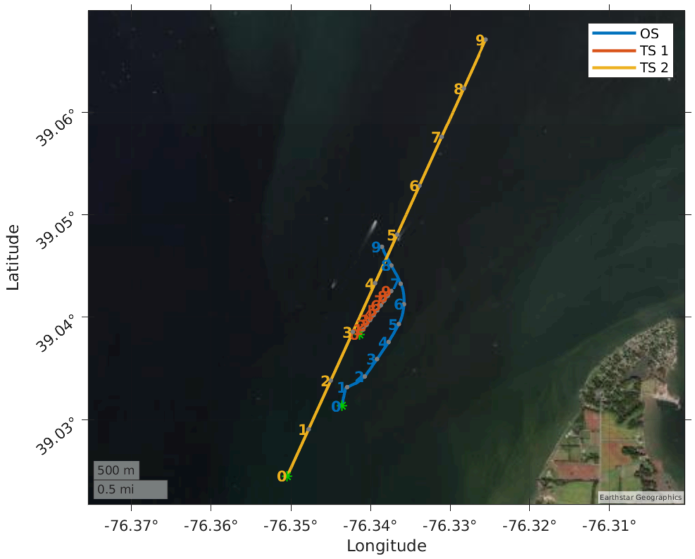


Figure 8.16: Scenario where ownship overtakes one vessel while being overtaken by a second vessel.

of ownship is safe by passing far enough ahead of TS1 so as not to violate the ship domain of either vessel.

Chapter 9

Conclusion

9.1 Part I Summary

The research in Part I of this thesis presented a Gaussian process-based adaptive sampling method with the goal of fully exploring an environment workspace, while also localizing and heavily sampling regions of interest that exhibit extreme sensory measurements and high spatially-correlated variability when compared to the surrounding area. The proposed GPAS algorithm automatically balances the trade-off between exploration and exploitation and dynamically adjusts the sampling density according to sensor measurements. Two informative path planning methods (one based on branch and bound techniques and one based on cross-entropy optimization) were presented to select future sampling locations in a 3D environment, while also considering the motion constraints of the system over the planning horizon. Both the adaptive sampling algorithm and the path planning subroutines were applied to an underactuated AUV, and their performance against area-coverage survey paths was studied in simulation. Further, two preliminary field experiments were performed in the Severn River, MD based on both a virtual environmental distribution and a distribution based on in situ dissolved oxygen measurements. Results from both the simulation studies and field experiments show that the proposed methodology is able to be fielded on operational platforms and preferentially collect measurements in regions of interest without sacrificing overall model fidelity of the full sampling area.

9.2 Part II Summary

The research in Part II then shifted to another domain within marine robotics: improved ASV navigation that complies with COLREGS protocols and good seamanship principles. The strategy to achieve this goal entailed a risk-based, multi-layer path planning approach that quantifies good seamanship through the use of ship domain and ship arena concepts. This approach naturally applies to both single-vessel scenarios, where adherence to COLREGS protocols is required, and multi-vessel scenarios where COLREGS are ill-defined.

Beyond just developing improved navigation for ASVs, Part II also introduced a simulation-based testing framework for improved performance evaluation of black-box autonomous systems. This framework utilizes statistical learning techniques such as adaptive scenario generation and unsupervised clustering to extract the system's failure modes. Additionally, this work introduced how these components can be used for regression testing purposes to track changes in performance between system updates. Ultimately, the performance analysis was applied to the good seamanship path planning approach to compare it with other baseline ASV planners, perform an analysis of its failure modes, and conduct a case study on the utility of tracking performance regression in a continuous integration fashion. This analysis culminated with several on-water tests with a 29-ft ASV that demonstrated how the good seamanship planner successfully handled various single-vessel and multi-vessel COLREGS scenarios.

9.3 Future Work

There are several areas of future work that are natural extensions to the research in this thesis. With regards to adaptive environmental sampling for AUV applications, the following items are a subset that offer immediate improvements to the goal of more informed data collection.

- Mission efficiency could be improved by investigating more intelligent surfacing strategies based on communication objectives or uncertainty thresholds. Surfacing only when necessary would limit the number of disruptions when collecting samples at depth. In applications where there is not a fixed mission duration, planning efficiency could also be improved by considering the additional desire to minimize travel distance or mission time while maximizing information gain.

- Another improvement to the AUV sampling work of Part I would be to incorporate knowledge of tidal currents (or other environmental effects) into the path planning process. Smaller platforms such as the AUV used in this research are particularly susceptible to even moderate tidal currents, which can lead to compromised maneuvering. Exploiting currents when performing vehicle planning could be beneficial when performing sampling missions over a larger area.

- A larger extension to the work in Part I would be to incorporate multiple sampling vehicles within a heterogeneous robot team. Robot teams are able to exploit the strengths of different platforms in order to collect the desired measurement profile over a larger sampling area. For instance, UAVs could collect sparse samples over very large areas that inform where more dense sampling should be executed using the AUV strategies of this thesis.

There are also many avenues to improve the ASV navigation work of Part II. The following research subjects are prime for continuing work in this area.

- An obvious direction that needs to be tested is the performance of the good seamanship planner when operating on data from a real perception system. While the field experiments tried to replicate some of the nuances of real data by adding noise effects, there is no substitute for testing a complete end-to-end system. Future work should plan to perform additional testing with real perception data that could identify new failure modes of the planner.

- In a similar vein, the uncertainty from real sensors affects the predicted state evolution of target ships in the environment. An interesting avenue to pursue would be improved prediction of how target ships might behave in the future when planning trajectories for ownship. This might entail fitting a distribution to the possible trajectories that the target ship might take, and then planning based on minimizing the overall risk that arises from the range of these possible behaviors. This is indeed an open and challenging research problem, as the behavior of the target ship would be dependent on things such as the COLREGS geometry, the type and size of both vessels, and even other vessels in the environment.

- Also related to the integration of improved ASV perception would be to extend the planning framework to accommodate COLREGS protocols that were not addressed in this research. The first area to extend would be to classify the target ship type (e.g., sailing vessel, fishing vessel, vessel constrained by draft, etc.), and then appropriately give way if the target ship has a higher priority per COLREGS. Further extensions could then be done to properly identify and react to lights, sound signals, and day shapes.

Both of the marine robotics applications discussed in this thesis (as well as autonomous systems from any domain) could also benefit from continued research into improved T&E of their black-box decision-making. The framework of Chapter 7 provides a starting point for this, but more work certainly needs to be done. In particular, one of the biggest challenges remaining is to appropriately handle the uncertainty associated with the performance landscape of the autonomy, especially given stochastic perception and the probabilistic decision-making of many modern autonomous systems. Methods to correlate the performance uncertainty across multiple samples of the testing space would be useful in determining the system's robustness under certain combinations of testing parameters. Detailed quantification of this uncertainty may also improve the probability estimates of rare-event failure modes.

As a final discussion point, another outstanding T&E challenge that is demanding extensive thought in the testing community is how to properly perform verification and validation (V&V) of autonomous systems, particularly when simulation-based testing must be relied on to augment performance data that field testing cannot provide. Traditional simulation V&V was able to use knowledge of physical processes (e.g., physics-based modeling) to gain trust in a simulation. The added component of autonomous decision-making exacerbates the challenge of simulation V&V, as it becomes much harder to predict how a scenario may translate from a simulation to the real world. The augmented continuous integration framework introduced in Chapter 7 is perhaps a step towards a common testing paradigm that would give testers the data and knowledge needed to gain trust in autonomous systems.

Bibliography

[1] R. Alexander, H. R. Hawkins, and A. J. Rae, "Situation coverage–a coverage criterion for testing autonomous robots," 2015.

[2] S. Aminikhanghahi and D. J. Cook, "A survey of methods for time series change point detection," *Knowledge and information systems*, vol. 51, no. 2, pp. 339–367, 2017.

[3] S. Arora and S. Scherer, "Randomized algorithm for informative path planning with budget constraints," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 4997–5004.

[4] F. Batsch, S. Kanarachos, M. Cheah, R. Ponticelli, and M. Blundell, "A taxonomy of validation strategies to ensure the safe operation of highly automated vehicles," *Journal of Intelligent Transportation Systems*, pp. 1–20, 2020.

[5] M. R. Benjamin, J. J. Leonard, J. A. Curcio, and P. M. Newman, "A method for protocol-based collision avoidance between autonomous marine surface craft," *Journal of Field Robotics*, vol. 23, no. 5, pp. 333–346, 2006.

[6] K. Bergman, O. Ljungqvist, J. Linder, and D. Axehill, “A colregs compliant motion planner for autonomous maneuvering of marine vessels in complex environments,” *arXiv preprint arXiv:2012.12145*, 2020.

[7] I. R. Bertaska, B. Shah, K. von Ellenrieder, P. Švec, W. Klinger, A. J. Sinisterra, M. Dhanak, and S. K. Gupta, “Experimental evaluation of automatically-generated behaviors for usv operations,” *Ocean Engineering*, vol. 106, pp. 496–514, 2015.

[8] E. Bertolazzi and M. Frego, “G1 fitting with clothoids,” *Mathematical Methods in the Applied Sciences*, vol. 38, no. 5, pp. 881–897, 2015.

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

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

[11] M. Breivik *et al.*, “Mpc-based mid-level collision avoidance for asvs using nonlinear programming,” in *2017 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE, 2017, pp. 766–772.

[12] A. C. Bukhari, I. Tusseyeva, Y.-G. Kim *et al.*, “An intelligent real-time multi-vessel collision risk assessment system from vts view point based on fuzzy inference system,” *Expert systems with applications*, vol. 40, no. 4, pp. 1220–1230, 2013.

[13] A. D. Bull, “Convergence rates of efficient global optimization algorithms,” *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2879–2904, 2011.

[14] W. Caharija, K. Y. Pettersen, M. Bibuli, P. Calado, E. Zereik, J. Braga, J. T. Gravdahl, A. J. Sørensen, M. Milovanović, and G. Bruzzone, “Integral line-of-sight guidance and control of underactuated marine vehicles: Theory, simulations, and experiments,” *IEEE Transactions on Control Systems Technology*, vol. 24, no. 5, pp. 1623–1642, 2016.

[15] N. Cao, K. H. Low, and J. M. Dolan, “Multi-robot informative path planning for active sensing of environmental phenomena: A tale of two algorithms,” in *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2013, pp. 7–14.

[16] H.-T. L. Chiang and L. Tapia, “Colreg-rrt: An rrt-based colregs-compliant motion planner for surface vehicle navigation,” *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2024–2031, 2018.

[17] A. L. Christensen, R. OâĂŹGrady, M. Birattari, and M. Dorigo, “Fault detection in autonomous robots based on fault injection and learning,” *Autonomous Robots*, vol. 24, no. 1, pp. 49–67, 2008.

[18] T. Coldwell, “Marine traffic behaviour in restricted waters,” *The Journal of Navigation*, vol. 36, no. 3, pp. 430–444, 1983.

[19] E. Contal, V. Perchet, and N. Vayatis, “Gaussian process optimization with mutual information,” in *International Conference on Machine Learning*, 2014, pp. 253–261.

[20] A. Corso, P. Du, K. Driggs-Campbell, and M. J. Kochenderfer, “Adaptive stress testing with reward augmentation for autonomous vehicle validatio,” in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 163–168.

[21] R. Cui, Y. Li, and W. Yan, “Mutual information-based multi-auv path planning for scalar field sampling using multidimensional rrt,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 7, pp. 993–1004, 2016.

[22] J. Das, J. Harvey, F. Py, H. Vathsangam, R. Graham, K. Rajan, and G. S. Sukhatme, “Hierarchical probabilistic regression for auv-based adaptive sampling of marine phenomena,” in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 5571–5578.

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

[24] J. Das, F. Py, T. Maughan, T. OâĂŹReilly, M. Messié, J. Ryan, G. S. Sukhatme, and K. Rajan, “Coordinated sampling of dynamic oceanographic features with underwater vehicles and drifters,” *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 626–646, 2012.

[25] P.-T. De Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, "A tutorial on the cross-entropy method," *Annals of operations research*, vol. 134, no. 1, pp. 19–67, 2005.

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

[27] B.-O. H. Eriksen, G. Bitar, M. Breivik, and A. M. Lekkas, "Hybrid collision avoidance for asvs compliant with colregs rules 8 and 13-17," *arXiv preprint arXiv:1907.00198*, 2019.

[28] B.-O. H. Eriksen, M. Breivik, E. F. Wilthil, A. L. Flten, and E. F. Brekke, "The branching-course model predictive control algorithm for maritime collision avoidance," *Journal of Field Robotics*, vol. 36, no. 7, pp. 1222–1249, 2019.

[29] B.-O. H. Eriksen, "Collision avoidance and motion control for autonomous surface vehicles," 2019.

[30] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[31] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.

[32] G. Flaspohler, V. Preston, A. P. Michel, Y. Girdhar, and N. Roy, "Information-guided robotic maximum seek-and-sample in partially observable continuous environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3782–3789, 2019.

[33] T. I. Fossen, *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, 2011.

[34] E. W. Frew, J. Elston, B. Argrow, A. Houston, and E. Rasmussen, "Sampling severe local storms and related phenomena: Using unmanned aircraft systems," *IEEE Robotics & Automation Magazine*, vol. 19, no. 1, pp. 85–95, 2012.

[35] Y. Fuji and K. Tanaka, "Traffic capacity," *The Journal of Navigation*, vol. 24, pp. 543–552, 1971.

[36] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous systems*, vol. 61, no. 12, pp. 1258–1276, 2013.

[37] M. Gertler and G. R. Hagen, "Standard equations of motion for submarine simulation," David w Taylor Naval Ship Research and Development Center Bethesda MD, Tech. Rep., 1967.

[38] E. M. Goodwin, "A statistical study of ship domains," *The Journal of navigation*, vol. 28, no. 3, pp. 328–344, 1975.

[39] A. Gotovos, N. Casati, G. Hitz, and A. Krause, “Active learning for level set estimation,” in *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.

[40] U. Grömping, “Variable importance assessment in regression: linear regression versus random forest,” *The American Statistician*, vol. 63, no. 4, pp. 308–319, 2009.

[41] U. S. C. Guard, *Navigation Rules*. United States Department of Transportation, 1999.

[42] I. B. Hagen, D. K. M. Kufoalor, E. F. Brekke, and T. A. Johansen, “Mpc-based collision avoidance strategy for existing marine vessel guidance systems,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 7618–7623.

[43] J. D. Hagy, W. R. Boynton, C. W. Keefe, and K. V. Wood, “Hypoxia in chesapeake bay, 1950–2001: long-term change in relation to nutrient loading and river flow,” *Estuaries*, vol. 27, no. 4, pp. 634–658, 2004.

[44] Z. J. Harris, “Model-based cooperative acoustic navigation and parameter identification for underactuated underwater vehicles,” Ph.D. dissertation, Johns Hopkins University, 2019.

[45] Y. He, Y. Jin, L. Huang, Y. Xiong, P. Chen, and J. Mou, “Quantitative analysis of colreg rules and seamanship for autonomous collision avoidance at open sea,” *Ocean Engineering*, vol. 140, pp. 281–291, 2017.

[46] P. Henkart, "Severn riverkeeper water quality monitoring program 2012 report," Severn Riverkeeper, Tech. Rep., 2012.

[47] G. Hitz, E. Galceran, M.-È. 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.

[48] G. A. Hollinger and G. S. Sukhatme, "Sampling-based robotic information gathering algorithms," *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1271–1287, 2014.

[49] T. A. Johansen, T. Perez, and A. Cristofaro, "Ship collision avoidance and colregs compliance using simulation-based control behavior selection with predictive hazard assessment," *IEEE transactions on intelligent transportation systems*, vol. 17, no. 12, pp. 3407–3422, 2016.

[50] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.

[51] K. Kawaguchi, L. P. Kaelbling, and T. Lozano-Pérez, "Bayesian optimization with exponential convergence," in *Advances in neural information processing systems*, 2015, pp. 2809–2817.

[52] S. Kemna and G. S. Sukhatme, "Surfacing strategies for multi-robot adaptive informative sampling with a surface-based data hub," in *OCEANS 2018 MTS/IEEE Charleston*. IEEE, 2018, pp. 1–10.

[53] W. M. Kemp, W. R. Boynton, J. E. Adolf, D. F. Boesch, W. C. Boicourt, G. Brush, J. C. Cornwell, T. R. Fisher, P. M. Glibert, J. D. Hagy *et al.*, "Eutrophication of chesapeake bay: historical trends and ecological interactions," *Marine Ecology Progress Series*, vol. 303, pp. 1–29, 2005.

[54] K. Kijima and Y. Furukawa, "Automatic collision avoidance system using the concept of blocking area," *IFAC Proceedings Volumes*, vol. 36, no. 21, pp. 223–228, 2003.

[55] M. Kobilarov, "Cross-entropy motion planning," *The International Journal of Robotics Research*, vol. 31, no. 7, pp. 855–871, 2012.

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

[57] D. K. M. Kufoalor, E. F. Brekke, and T. A. Johansen, "Proactive collision avoidance for asvs using a dynamic reciprocal velocity obstacles method," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 2402–2409.

[58] Y. Kuwata, M. T. Wolf, D. Zarzhitsky, and T. L. Huntsberger, "Safe maritime autonomous navigation with colregs, using velocity obstacles," *IEEE Journal of Oceanic Engineering*, vol. 39, no. 1, pp. 110–119, 2013.

[59] N. R. Lawrance, J. J. Chung, and G. A. Hollinger, "Fast marching adaptive sampling," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 696–703, 2017.

[60] A. M. Lekkas and T. I. Fossen, “Line-of-sight guidance for path following of marine vehicles,” *Advanced in marine robotics*, pp. 63–92, 2013.

[61] N. E. Leonard, D. A. Paley, R. E. Davis, D. M. Fratantoni, F. Lekien, and F. Zhang, “Coordinated control of an underwater glider fleet in an adaptive ocean sampling field experiment in monterey bay,” *Journal of Field Robotics*, vol. 27, no. 6, pp. 718–740, 2010.

[62] N. E. Leonard, D. A. Paley, F. Lekien, R. Sepulchre, D. M. Fratantoni, and R. E. Davis, “Collective motion, sensor networks, and ocean sampling,” *Proceedings of the IEEE*, vol. 95, no. 1, pp. 48–74, 2007.

[63] A. Q. Li, I. Rekleitis, S. Manjanna, N. Kakodkar, J. Hansen, G. Dudek, L. Bobadilla, J. Anderson, and R. N. Smith, “Data correlation and comparison from multiple sensors over a coral reef with a team of heterogeneous aquatic robots,” in *International Symposium on Experimental Robotics*. Springer, 2016, pp. 717–728.

[64] B. Li and F.-W. Pang, “An approach of vessel collision risk assessment based on the d–s evidence theory,” *Ocean Engineering*, vol. 74, pp. 16–21, 2013.

[65] L. Li, W.-L. Huang, Y. Liu, N.-N. Zheng, and F.-Y. Wang, “Intelligence testing for autonomous vehicles: a new approach,” *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 2, pp. 158–166, 2016.

[66] S. Li, Q. Meng, and X. Qu, “An overview of maritime waterway quantitative risk assessment models,” *Risk Analysis: An International Journal*, vol. 32, no. 3, pp. 496–512, 2012.

[67] Z. W. Lim, D. Hsu, and W. S. Lee, "Adaptive informative path planning in metric spaces," *The International Journal of Robotics Research*, vol. 35, no. 5, pp. 585–598, 2016.

[68] J. Liu, F. Zhou, Z. Li, M. Wang, and R. W. Liu, "Dynamic ship domain models for capacity analysis of restricted water channels," *The Journal of Navigation*, vol. 69, no. 3, pp. 481–503, 2016.

[69] S. Liu, Y. Yue, and R. Krishnan, "Non-myopic adaptive route planning in uncertain congestion environments," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 9, pp. 2438–2451, 2015.

[70] K. H. Low, J. M. Dolan, and P. K. Khosla, "Information-theoretic approach to efficient adaptive path planning for mobile robotic environmental sensing." in *ICAPS*, 2009, pp. 233–240.

[71] M. Luckcuck, M. Farrell, L. A. Dennis, C. Dixon, and M. Fisher, "Formal specification and verification of autonomous robotic systems: A survey," *ACM Computing Surveys (CSUR)*, vol. 52, no. 5, pp. 1–41, 2019.

[72] K.-C. Ma, L. Liu, H. K. Heidarsson, and G. S. Sukhatme, "Data-driven learning and planning for environmental sampling," *Journal of Field Robotics*, vol. 35, no. 5, pp. 643–661, 2018.

[73] K.-C. Ma, L. Liu, and G. S. Sukhatme, "An information-driven and disturbance-aware planning method for long-term ocean monitoring," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 2102–2108.

[74] S. Manjanna and G. Dudek, "Data-driven selective sampling for marine vehicles using multi-scale paths," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 6111–6117.

[75] S. Manjanna, J. Hansen, A. Q. Li, I. Rekleitis, and G. Dudek, "Collaborative sampling using heterogeneous marine robots driven by visual cues," in *2017 14th Conference on Computer and Robot Vision (CRV)*. IEEE, 2017, pp. 87–94.

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

[77] S. Manjanna, H. Van Hoof, and G. Dudek, "Policy search on aggregated state space for active sampling," in *International Symposium on Experimental Robotics*. Springer, 2018, pp. 211–221.

[78] S. Mannor, R. Y. Rubinstein, and Y. Gat, "The cross entropy method for fast policy search," in *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, 2003, pp. 512–519.

[79] 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.

[80] J. A. Marvel and R. Bostelman, "A cross-domain survey of metrics for modelling and evaluating collisions," *International Journal of Advanced Robotic Systems*, vol. 11, no. 9, p. 142, 2014.

[81] MATLAB, *version 9.8.0 (R2020a)*. Natick, Massachusetts: The MathWorks Inc., 2020.

[82] G. J. McLachlan and D. Peel, *Finite mixture models*. John Wiley & Sons, 2004.

[83] J. McMahon, H. Yetkin, A. Wolek, Z. J. Waters, and D. J. Stilwell, "Towards real-time search planning in subsea environments," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 87–94.

[84] K. Meinke and P. Nycander, "Learning-based testing of distributed microservice architectures: Correctness and fault injection," in *SEFM 2015 Collocated Workshops*. Springer, 2015, pp. 3–10.

[85] N. Meinshausen, "Quantile regression forests," *Journal of Machine Learning Research*, vol. 7, no. Jun, pp. 983–999, 2006.

[86] F. Melgani, G. Moser, and S. B. Serpico, "Unsupervised change detection methods for remote sensing images," in *Image and Signal Processing for Remote Sensing VII*, vol. 4541. International Society for Optics and Photonics, 2002, pp. 211–222.

[87] M. Miller, I. Ketcham, and T. Guay, "A mid-summer 2019 report: Tracking the severn river's dead zone," Severn River Association, Tech. Rep., 2019.

[88] P. K. E. Minne, "Automatic testing of maritime collision avoidance algorithms," Master's thesis, NTNU, 2017.

[89] S. Mukhopadhyay, C. Wang, M. Patterson, M. Malisoff, and F. Zhang, "Collaborative autonomous surveys in marine environments affected by oil spills," in *Cooperative Robots and Sensor Networks 2014*. Springer, 2014, pp. 87–113.

[90] A. C. Muller, D. L. Muller, and A. Muller, "Resolving spatiotemporal characteristics of the seasonal hypoxia cycle in shallow estuarine environments of the severn river and south river, md, chesapeake bay, usa," *Heliyon*, vol. 2, no. 9, p. e00157, 2016.

[91] G. E. Mullins, P. G. Stankiewicz, and S. K. Gupta, "Automated generation of diverse and challenging scenarios for test and evaluation of autonomous vehicles," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1443–1450.

[92] G. E. Mullins, P. G. Stankiewicz, R. C. Hawthorne, J. D. Appler, M. H. Biggins, K. Chiou, M. A. Huntley, J. D. Stewart, and A. S. Watkins, "Delivering test and evaluation tools for autonomous unmanned vehicles to the fleet," *JOHNS HOPKINS APL TECHNICAL DIGEST*, vol. 33, no. 4, pp. 279–288, 2017.

[93] G. E. Mullins, P. G. Stankiewicz, R. C. Hawthorne, and S. K. Gupta, "Adaptive generation of challenging scenarios for testing and evaluation of autonomous vehicles," *Journal of Systems and Software*, vol. 137, pp. 197–215, 2018.

[94] A. Munafò, E. Simetti, A. Turetta, A. Caiti, and G. Casalino, "Autonomous underwater vehicle teams for adaptive ocean sampling: a data-driven approach," *Ocean Dynamics*, vol. 61, no. 11, pp. 1981–1994, 2011.

[95] K. P. Murphy, *Machine learning: a probabilistic perspective.* MIT press, 2012.

[96] G. Muscato, F. Bonaccorso, L. Cantelli, D. Longo, and C. D. Melita, "Volcanic environments: robots for exploration and measurement," *IEEE Robotics & Automation Magazine*, vol. 19, no. 1, pp. 40–49, 2012.

[97] M. F. Mysorewala, D. O. Popa, and F. L. Lewis, "Multi-scale adaptive sampling with mobile agents for mapping of forest fires," *Journal of Intelligent and Robotic Systems*, vol. 54, no. 4, p. 535, 2009.

[98] W. Naeem, G. W. Irwin, and A. Yang, "Colregs-based collision avoidance strategies for unmanned surface vehicles," *Mechatronics*, vol. 22, no. 6, pp. 669–678, 2012.

[99] P. K. Novak, N. Lavrač, and G. I. Webb, "Supervised descriptive rule discovery: A unifying survey of contrast set, emerging pattern and subgroup mining." *Journal of Machine Learning Research*, vol. 10, no. 2, 2009.

[100] M. O'Kelly, A. Sinha, H. Namkoong, R. Tedrake, and J. C. Duchi, "Scalable end-to-end autonomous vehicle testing via rare-event simulation," *Advances in Neural Information Processing Systems*, vol. 31, pp. 9827–9838, 2018.

[101] R. Ouyang, K. H. Low, J. Chen, and P. Jaillet, "Multi-robot active sensing of non-stationary gaussian process-based environmental phenomena," in *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2014, pp. 573–580.

[102] L. Perera, J. Carvalho, and C. G. Soares, "Fuzzy logic based decision making system for collision avoidance of ocean navigation under critical collision conditions," *Journal of marine science and technology*, vol. 16, no. 1, pp. 84–99, 2011.

[103] Z. Pietrzykowski, "Ship's fuzzy domain–a criterion for navigational safety in narrow fairways," *The Journal of Navigation*, vol. 61, no. 3, pp. 499–514, 2008.

[104] Z. Pietrzykowski and J. Uriasz, "The ship domain–a criterion of navigational safety assessment in an open sea area," *The Journal of Navigation*, vol. 62, no. 1, pp. 93–108, 2009.

[105] B. Portlock, "State of the bay 2020," Chesapeake Bay Foundation, Tech. Rep., 2020.

[106] W. B. Powell, *Approximate Dynamic Programming: Solving the curses of dimensionality*. John Wiley & Sons, 2007, vol. 703.

[107] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “Ros: an open-source robot operating system,” in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

[108] C. E. Rasmussen and C. K. Williams, *Gaussian process for machine learning*. MIT press, 2006.

[109] W. Reid, R. Fitch, A. H. Göktoğan, and S. Sukkarieh, “Sampling-based hierarchical motion planning for a reconfigurable wheel-on-leg planetary analogue exploration rover,” *Journal of Field Robotics*, 2019.

[110] S. Riedmaier, T. Ponn, D. Ludwig, B. Schick, and F. Diermeyer, “Survey on scenario-based safety assessment of automated vehicles,” *IEEE Access*, vol. 8, pp. 87 456–87 477, 2020.

[111] E. Rocklage, H. Kraft, A. Karatas, and J. Seewig, “Automated scenario generation for regression testing of autonomous vehicles,” in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017, pp. 476–483.

[112] P. J. Rousseeuw, “Silhouettes: a graphical aid to the interpretation and validation of cluster analysis,” *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[113] B. C. Shah, P. Švec, I. R. Bertaska, A. J. Sinisterra, W. Klinger, K. von Ellenrieder, M. Dhanak, and S. K. Gupta, “Resolution-adaptive risk-aware trajectory planning for surface vehicles operating in congested civilian traffic,” *Autonomous Robots*, vol. 40, no. 7, pp. 1139–1163, 2016.

[114] R. N. Smith, M. Schwager, S. L. Smith, B. H. Jones, D. Rus, and G. S. Sukhatme, "Persistent ocean monitoring with underwater gliders: Adapting sampling resolution," *Journal of Field Robotics*, vol. 28, no. 5, pp. 714–741, 2011.

[115] M. Song, Y. Zhong, and A. Ma, "Change detection based on multi-feature clustering using differential evolution for landsat imagery," *Remote Sensing*, vol. 10, no. 10, p. 1664, 2018.

[116] X. Song, M. Wu, C. Jermaine, and S. Ranka, "Statistical change detection for multi-dimensional data," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2007, pp. 667–676.

[117] N. Srinivas, A. Krause, S. M. Kakade, and M. Seeger, "Gaussian process optimization in the bandit setting: No regret and experimental design," *arXiv preprint arXiv:0912.3995*, 2009.

[118] P. Stankiewicz, M. Heistand, and M. Kobilarov, "Quantifying good seamanship for autonomous surface vessel performance evaluation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 8309–8315.

[119] P. Stankiewicz and M. Kobilarov, "Identifying performance regression conditions for testing and evaluation of autonomous systems," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

[120] P. Stankiewicz and M. Kobilarov, "A primitive-based approach to good seamanship path planning for autonomous surface vessels," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[121] P. Stankiewicz and G. Mullins, "Improving evaluation methodology for autonomous surface vessel colregs compliance," in *OCEANS 2019 - Marseille*. IEEE, 2019, pp. 1–7.

[122] P. Stankiewicz, Y. T. Tan, and M. Kobilarov, "Adaptive sampling with an autonomous underwater vehicle in static marine environments," *Journal of Field Robotics*, vol. 38, no. 4, pp. 572–597, 2021.

[123] P. Švec, B. C. Shah, I. R. Bertaska, W. Klinger, A. J. Sinisterra, K. Von Ellenrieder, M. Dhanak, and S. K. Gupta, "Adaptive sampling based colregs-compliant obstacle avoidance for autonomous surface vehicles," in *Proceedings of ICRA 2014 workshop on persistent autonomy for marine robotics*, 2014.

[124] R. Szlapczynski, "A unified measure of collision risk derived from the concept of a ship domain," *The Journal of navigation*, vol. 59, no. 3, pp. 477–490, 2006.

[125] R. Szlapczynski and J. Szlapczynska, "An analysis of domain-based ship collision risk parameters," *Ocean Engineering*, vol. 126, pp. 47–56, 2016.

[126] R. Szlapczynski and J. Szlapczynska, "Review of ship safety domains: Models and applications," *Ocean Engineering*, vol. 145, pp. 277–289, 2017.

[127] C. Tam and R. Bucknall, “Collision risk assessment for ships,” *Journal of marine science and technology*, vol. 15, no. 3, pp. 257–270, 2010.

[128] Y. T. Tan, A. Kunapareddy, and M. Kobilarov, “Gaussian process adaptive sampling using the cross-entropy method for environmental sensing and monitoring,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 6220–6227.

[129] T. Tengesdal, T. A. Johansen, and E. Brekke, “Risk-based autonomous maritime collision avoidance considering obstacle intentions,” in *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*. IEEE, 2020, pp. 1–8.

[130] C. E. Tuncali, G. Fainekos, H. Ito, and J. Kapinski, “Simulation-based adversarial test generation for autonomous vehicles with machine learning components,” in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 1555–1562.

[131] C. E. Tuncali, T. P. Pavlic, and G. Fainekos, “Utilizing s-taliro as an automatic test generation framework for autonomous vehicles,” in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2016, pp. 1470–1475.

[132] J. H. Uphoff Jr, M. McGinty, R. Lukacovic, J. Mowrer, and B. Pyle, “Impervious surface, summer dissolved oxygen, and fish distribution in chesapeake bay subestuaries: linking watershed development, habitat conditions, and fisheries management,” *North American Journal of Fisheries Management*, vol. 31, no. 3, pp. 554–566, 2011.

[133] D. Wang, P. F. Lermusiaux, P. J. Haley, D. Eickstedt, W. G. Leslie, and H. Schmidt, "Acoustically focused adaptive sampling and on-board routing for marine rapid environmental assessment," *Journal of Marine Systems*, vol. 78, pp. S393–S407, 2009.

[134] N. Wang, "An intelligent spatial collision risk based on the quaternion ship domain," *The Journal of Navigation*, vol. 63, no. 4, pp. 733–749, 2010.

[135] Z. Wang, B. Shakibi, L. Jin, and N. de Freitas, "Bayesian multi- scale optimistic optimization," 2014.

[136] K. Woerner, "Multi-contact protocol-constrained collision avoidance for autonomous marine vehicles," Ph.D. dissertation, Massachusetts Institute of Technology, 2016.

[137] K. Woerner, M. R. Benjamin, M. Novitzky, and J. J. Leonard, "Quantifying protocol evaluation for autonomous collision avoidance," *Autonomous Robots*, pp. 1–25, 2018.

[138] C.-K. Wong and M. C. Easton, "An efficient method for weighted sampling without replacement," *SIAM Journal on Computing*, vol. 9, no. 1, pp. 111–113, 1980.

[139] S. Yoo and M. Harman, "Regression testing minimization, selection and prioritization: a survey," *Software testing, verification and reliability*, vol. 22, no. 2, pp. 67–120, 2012.

[140] X. Zhu, H. Xu, and J. Lin, "Domain and its model based on neural networks," *The Journal of Navigation*, vol. 54, no. 1, pp. 97–103, 2001.

[141] X. Zou, R. Alexander, and J. McDermid, “Testing method for multi-uav conflict resolution using agent-based simulation and multi-objective search,” *Journal of Aerospace Information Systems*, pp. 191–203, 2016.