UNIVERSITÉ DE LORRAINE | BIBLIOTHÈQUES UNIVERSITAIRES

AVERTISSEMENT
DROIT D'AUTEUR

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document. Toute contrefaçon, plagiat ou reproduction illicite encourt une poursuite pénale.

Contact bibliothèque : ddoc-theses-contact@univ-lorraine.fr
(Cette adresse ne permet pas de contacter les auteurs)

Code de la Propriété Intellectuelle. articles L 122. 4
Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

UNIVERSITÉ DE LORRAINE
34 COURS LÉOPOLD — CS 25233
54052 NANCY CEDEX
TÉL : 03 54 50 54 00
CONTACT@UNIV-LORRAINE.FR
WWW.UNIV-LORRAINE.FR

UNIVERSITÉ DE LORRAINE

École doctorale IAEM Lorraine

Autonomous System For Identifying and Capturing Floating Waste

THÈSE

présentée et soutenue publiquement le 19 decembre 2025

pour l'obtention du

Doctorat de l'Université de Lorraine
en cotutelle avec Georgia Institute of Technology
(mention automatique, traitement du signal et des images, génie informatique)

par

Luis Felipe Wolf Batista

Composition du jury

Président :	Roland Lenain
Rapporteurs :	Alberto Ortiz Roland Lenain
Examinateurs :	Assia Benbihi Claire Dune Jeremy Fix Sehoon Ha
Encadrants :	Cédric Pradalier Seth Hutchinson

Laboratoire Georgia Tech-CNRS IRL 2958

Mis en page avec la classe thesul.

Acknowledgments

First, I'd like to thank both of my advisors. I am deeply grateful for the privilege of discovering robotics research under your mentorship. Cédric, thank you for your incredible availability and constant support throughout this PhD, and for creating the opportunity that made it possible in the first place. Your ability to bring fresh ideas at every stage, to ask the right questions, and to stay fully engaged from high-level research direction to the practical implementation details is truly remarkable. I am profoundly grateful for your continued support and trust. Seth, thank you for providing invaluable guidance and inspiration. Your perspective helped me learn different aspects of research, and your encouragement regarding the value of engineering contributions in academia gave me confidence in the direction I was pursuing. I am grateful for your generosity with your time and the example you set as a researcher and mentor.

I would also like to acknowledge all members of the jury for their insightful feedback and support at different stages of my research. Your comments and questions helped improve this thesis and sharpen the way I think about the broader impact of the work.

Special thanks are due to all the friends and colleagues from the DREAM Lab, GT, and UL whom I had the privilege to work with during these years. If I tried to list everyone here, this section might end up longer than the thesis itself. Thank you for making this such a stimulating and welcoming environment through lively discussions, collaboration, and everyday support. I am especially grateful for all the help with field tests in the lake, including the patience, troubleshooting, and logistics that made the experiments possible.

Agradecimento especial à minha família. À minha esposa, Carolina, pelo apoio desde a oportunidade de voltar a ser estudante e, no dia a dia, por tolerar com paciência a minha ausência. Sem você, nada disso seria possível. À minha irmã, Luciana, por ser exemplo e inspiração. Aos meus pais, Edilson e Lucia, por todo suporte, exemplo e amor incondicional. Sou eternamente grato a vocês.

"You cannot swim for new horizons until you have courage to lose sight of the shore."
— William Faulkner

Contents

Glossary **xi**

Résumé étendu en français **xiii**

Chapter 1 Introduction **1**

1.1 About the PhD dissertation topic 1
1.2 Research Questions 1
1.3 Contributions 2
1.4 Outline 3
1.5 List of Publications 4

I Perception **7**

Chapter 2 Polarization-Based Detection **9**

2.1 Introduction 10
2.2 Related Work 11
2.3 PoTATO Dataset 12
2.4 Overview of Light Polarization Theory 14
2.4.1 Sources of Polarization 14
2.4.2 Extracting Polarimetric Data 14
2.4.3 Implemented Pipeline 16
2.5 Experiments and Analysis 17
2.5.1 Object Detection Baseline 17
2.5.2 Advantages of Polarimetric Information 18
2.5.3 Qualitative Analysis 19
2.5.4 Challenges 20
2.6 Conclusion 21

Chapter 3 Polarimetric and Chromatic Fusion for Segmentation **23**
3.1 Introduction . . . 24
3.2 Related Work . . . 24
3.3 Methodology . . . 25
3.3.1 Dataset Preparation . . . 25
3.3.2 Segmentation Architectures . . . 26
3.3.3 Experimental Details . . . 26
3.4 Results . . . 27
3.5 Discussion . . . 29
3.6 Conclusion . . . 30

II Reinforcement Learning-based Navigation **31**

Chapter 4 Simulation and Reinforcement Learning Framework **33**
4.1 Introduction . . . 34
4.2 Reinforcement Learning Framework . . . 34
4.3 Framework evolution beyond this thesis . . . 35
4.4 Research Collaboration . . . 37
4.5 Conclusion . . . 37

Chapter 5 Deep Reinforcement Learning Framework for ASV Navigation **39**
5.1 Introduction . . . 40
5.2 Related Work . . . 41
5.3 Problem Formulation . . . 42
5.3.1 Capture task . . . 42
5.3.2 Dynamic Model . . . 42
5.4 Methodology . . . 42
5.4.1 System Identification . . . 42
5.4.2 RL Framework for ASV . . . 44
5.4.3 Training process . . . 45
5.4.4 Simulation Environment . . . 47
5.4.5 Autonomous Surface Vehicle Platform Integration . . . 47
5.5 Experiments . . . 48
5.5.1 Simulation Evaluation . . . 48
5.5.2 Real-world Evaluation . . . 48
5.5.3 Evaluation Metrics . . . 48
5.5.4 Results . . . 49

5.6 Discussion . . . 50
5.7 Conclusion . . . 51

Chapter 6 Robust ASV Navigation **53**
6.1 Introduction . . . 54
6.2 Methodology . . . 55
6.2.1 Problem Formulation . . . 55
6.2.2 Platform . . . 55
6.2.3 Control approach . . . 56
6.2.4 Evaluation Methodology . . . 57
6.3 Experiments . . . 58
6.3.1 Simulation . . . 58
6.3.2 Field tests . . . 60
6.4 Discussion . . . 61
6.5 Challenges and Lessons Learned . . . 61
6.6 Conclusion . . . 63

III Integration **65**

Chapter 7 Integrated Autonomous Floating Waste Collection **67**
7.1 Introduction . . . 68
7.2 Related Work . . . 68
7.2.1 Autonomous Surface Vehicles and Challenges . . . 69
7.2.2 Environmental Application . . . 70
7.2.3 Perception . . . 70
7.2.4 Reinforcement Learning-based Control . . . 70
7.2.5 Robustness and Policy Transfer . . . 71
7.3 Experimental Setup . . . 71
7.3.1 Robot Platform . . . 71
7.3.2 Integration and Evaluation Testbed . . . 72
7.3.3 Policy Training Simulator . . . 74
7.4 Polarimetric Perception . . . 75
7.4.1 Polarimetric Object Detection . . . 75
7.4.2 Calibration and Projection . . . 75
7.4.3 Perception Pipeline Characterization . . . 77
7.5 Reinforcement Learning-based Control . . . 79
7.5.1 Problem Formulation . . . 79

7.5.2 Methodology . . . 79
7.5.3 Experiments . . . 82
7.6 Integration . . . 88
7.6.1 Problem Formulation . . . 88
7.6.2 Methodology . . . 88
7.6.3 Experiments . . . 90
7.7 Discussion . . . 93
7.7.1 Limitations and Future Work . . . 93
7.7.2 Lessons Learned . . . 94
7.8 Conclusion . . . 97

Chapter 8 Conclusion **99**
8.1 Contributions . . . 99
8.1.1 Perception . . . 99
8.1.2 Reinforcement learning-based control . . . 99
8.1.3 Integration and field testing . . . 100
8.2 Future Work . . . 100
8.3 Concluding Remarks . . . 101

Bibliography **103**

Résumé

La pollution plastique dans les milieux aquatiques constitue un défi critique, causant des dommages significatifs aux écosystèmes et à la santé humaine. Les méthodes de nettoyage traditionnelles requièrent beaucoup de main-d'œuvre et sont limitées en termes de scalabilité et de couverture, suscitant un intérêt pour des solutions robotiques autonomes. Cette thèse étudie le développement d'un système autonome de détection et de collecte des déchets flottants, combinant une perception visuelle avancée, un contrôle fondé sur l'apprentissage par renforcement, et une intégration système adaptée aux environnements aquatiques. L'application envisagée répond non seulement à un problème écologique majeur, mais présente également des défis techniques substantiels en matière de perception, de navigation et de déploiement sur le terrain.

La première contribution de ce travail est l'amélioration de la perception visuelle par l'imagerie polarimétrique. Les réflexions spéculaires à la surface de l'eau compliquent les tâches de détection avec des caméras standard. Pour y remédier, le jeu de données PoTATO a été créé, contenant des images de bouteilles en plastique flottantes acquises avec une caméra polarimétrique. Il inclut plusieurs modalités polarimétriques, et les expériences montrent que les entrées polarimétriques, en particulier lorsqu'elles sont fusionnées avec l'information chromatique, améliorent la précision de détection sous des conditions d'éclairage variées.

La deuxième contribution est le développement d'un cadre d'apprentissage par renforcement (RL) pour la navigation d'un véhicule de surface autonome (ASV), intégrant une modélisation hydrodynamique et une randomisation de domaine afin de réduire le écart « sim-to-real » (simulation-vers-réalité). Le contrôleur, fondé sur des méthodes de RL, est évalué en termes d'efficacité énergétique et de temps d'accomplissement de la tâche, et il est testé en présence de perturbations telles que des charges utiles décalées et une traînée asymétrique. Des essais de terrain approfondis démontrent la robustesse des politiques entraînées et valident l'approche d'entrainement.

Enfin, la thèse intègre la perception et le contrôle au sein d'une chaîne de traitement de bout en bout pour la collecte autonome des déchets. Les objets détectés sont cartographiés en cibles de navigation à l'aide de transformations calibrées, et l'ASV est guidé par l'agent de RL. Le système intégré est évalué en simulation comme en conditions réelles, mettant en évidence son applicabilité pour des applications de dépollution environnementale.

En résumé, ce travail propose une approche globale de la collecte autonome des déchets en faisant progresser les méthodes de perception, de contrôle et d'intégration. Les essais sur le terrain constituent un élément central de l'évaluation, démontrant des performances fiables en conditions réelles. Ces résultats soulignent le potentiel des systèmes autonomes à contribuer à la surveillance de l'environnement et à l'atténuation de la pollution par les déchets.

Mots-clés: Véhicules de Surface Autonomes, Vision par Ordinateur, Apprentissage par Renforcement, Robotique de Terrain.

Abstract

Plastic pollution in aquatic environments poses a critical challenge, causing significant harm to ecosystems and human health. Traditional cleanup methods are often labor-intensive and limited in scalability and coverage, prompting interest in autonomous robotic solutions. This dissertation investigates the development of an autonomous system for detecting and collecting floating waste, combining advanced visual perception and control based on reinforcement learning (RL) into an integrated system tailored for aquatic environments. The application not only addresses an important ecological problem but also presents substantial technical challenges in perception, navigation, and field deployment.

The first contribution of this work is the enhancement of visual perception through polarimetric imaging. Specular reflections from water surfaces complicate detection tasks using standard cameras. To overcome this, the PoTATO dataset was created, containing images of floating plastic bottles captured with a polarimetric camera. The dataset includes multiple polarimetric modalities, and experiments demonstrate that polarimetric inputs, especially when fused with color information, improve detection accuracy under diverse lighting conditions.

The second contribution is the development of a reinforcement learning framework for autonomous surface vehicle (ASV) navigation, incorporating hydrodynamic modeling and domain randomization to reduce the sim-to-real (simulation-to-reality) gap. The RL-based controller is evaluated with respect to energy efficiency and task completion time, and is tested under disturbances such as shifting payloads and asymmetric drag. Comprehensive field experiments demonstrate the robustness of the trained policies and validate the training approach.

Finally, the dissertation integrates perception and control into an end-to-end pipeline for autonomous waste collection. Detected objects are mapped to navigation targets using calibrated transformations, and the ASV is guided by the RL agent. The integrated system is evaluated in both simulation and real-world settings, highlighting the feasibility for environmental cleanup applications.

In summary, this work demonstrates a comprehensive approach to autonomous waste collection by advancing perception, control, and integration methods. Field trials are central to the evaluation, ensuring that the system performs reliably under real-world conditions. These findings highlight the potential of autonomous systems to contribute to environmental monitoring and waste mitigation.

Keywords: Autonomous Surface Vehicles, Computer Vision, Reinforcement Learning, Field Robotics.

Glossary

ACADO : Automatic Control and Dynamic Optimization (toolkit)
AI : Artificial Intelligence
AoLP : Angle of Linear Polarization
AP : Average Precision
AR : Average Recall
ASD : Average Surface Distance
ASV : Autonomous Surface Vehicle

COCO : Common Objects in Context (dataset)
CoM : Center of Mass
CPU : Central Processing Unit
CUDA : Compute Unified Device Architecture

DIF : Diffuse-only reflection image (glare-suppressed)
DOLP : Degree of Linear Polarization (visualization)
DoLP : Degree of Linear Polarization
DR : Domain Randomization
DRL : Deep Reinforcement Learning

GNSS : Global Navigation Satellite System
GPS : Global Positioning System
GPU : Graphics Processing Unit

HDR : High Dynamic Range
HSV : Hue–Saturation–Value

IADYS : Interactive Autonomous Dynamic Systems (company)
IMU : Inertial Measurement Unit
IoU : Intersection over Union
IQR : Interquartile range

LiPo : Lithium-Polymer (battery)

MDP : Markov Decision Process
MPC : Model Predictive Control

NMPC : Nonlinear Model Predictive Control

OCP : Optimal Control Problem
OIGE : OmniIsaacGymEnvs

PAULI : Pauli-inspired polarimetric composite image
POL : Polarimetric HSV composite (DoLP as value, AoLP as hue)
PPM : Pulse Position Modulation
PPO : Proximal Policy Optimization

RGB : Red–Green–Blue
RGGB : Red–Green–Green–Blue (Bayer pattern)
RL : Reinforcement Learning
ROS : Robot Operating System
RTK-GPS : Real-Time Kinematic Global Positioning System

SAM : Segment Anything Model

USV : Uncrewed Surface Vessel
UUV : Underwater Unmanned Vehicle (Gazebo plugin)

YOLOv5 : You Only Look Once, version 5

Résumé étendu en français

Chapitre 1 : Introduction

Cette thèse s'inscrit dans le développement de solutions de perception et de commande pour des systèmes autonomes opérant en conditions réelles, avec pour application la collecte de déchets flottants à la surface de l'eau. L'enjeu est à la fois environnemental (détection et récupération plus fiables) et robotique (transfert de contributions méthodologiques vers des systèmes déployables). Ces objectifs structurent le projet R3AMA (Robust Reinforcement Learning for Real-World Autonomous Multitask Agents), mené conjointement par le SpaceR Lab (Université du Luxembourg), IADYS (Jellyfishbot) et le DREAM Lab (CNRS IRL 2958, Georgia Tech Europe).

Le problème scientifique est la conception d'une chaîne autonome complète, de la perception à la navigation, capable de détecter des objets flottants puis de guider un véhicule de surface autonome (ASV) jusqu'à leur capture malgré les contraintes du milieu aquatique. Trois questions guident l'approche. (1) La perception visuelle sur l'eau est fortement dégradée par l'éblouissement, les variations d'illumination et les réflexions spéculaires, ce qui limite la fiabilité des systèmes fondés uniquement sur des caméras RGB. La thèse étudie l'apport de la polarimétrie, naturellement présente en extérieur, pour renforcer la robustesse des algorithmes de détection. (2) Une fois la cible localisée, l'ASV doit converger vers celle-ci sous incertitudes dynamiques et perturbations (courants, vent) difficiles à modéliser et à observer. Dans ce cadre, l'apprentissage par renforcement profond (Deep Reinforcement Learning, DRL) est considéré comme une alternative robuste aux erreurs de modélisation, au bruit de capteurs et aux perturbations, avec un accent mis sur la validation en conditions réelles. (3) L'intégration perception–commande introduit des erreurs couplées (incertitude de détection, calibration, latence), susceptibles de dégrader la performance sur le terrain. La thèse privilégie donc une architecture modulaire et une démarche de transfert *zero-shot* simulation-vers-réel, associée à une évaluation end-to-end fondée sur des métriques explicites.

Les contributions se regroupent en trois axes. (i) **Perception polarimétrique** : exploitation de caméras polarimétriques pour atténuer les effets de réflectivité et améliorer la détectabilité d'objets flottants. (ii) **DRL pour la navigation d'un ASV** : cadre open-source étendant des méthodes avancées afin d'obtenir des politiques robustes aux variations dynamiques, au bruit et aux perturbations, avec validation sur robot réel. (iii) **Intégration perception–commande et évaluation sim-to-real** : chaîne complète couplant détection polarimétrique et navigation par DRL, assortie d'un protocole de stress-test et de métriques end-to-end pour quantifier la robustesse et guider les améliorations nécessaires au transfert *zero-shot*.

Le manuscrit est organisé en trois parties (Perception, Navigation par apprentissage, Intégration), dans un format de thèse par publications. La validation terrain constitue le fil conducteur : l'objectif est de concevoir des solutions techniquement fondées et surtout éprouvées en conditions réelles, afin d'en caractériser les limites et d'identifier des pistes d'amélioration vers des systèmes

opérationnels.

Chapitre 2 : Détection fondée sur la polarisation

Ce chapitre traite de la détection de déchets flottants en environnement aquatique extérieur. Il part du constat que les détecteurs fondés sur l'imagerie RGB se dégradent fortement sur l'eau (illumination variable, éblouissement, réflexions). La polarisation fournit une information complémentaire mesurable par des caméras polarimétriques, et l'objectif est de préciser dans quelles conditions cette modalité améliore la détection, tout en proposant une base de données ouverte pour la robotique de terrain.

La contribution principale est PoTATO, un jeu de données pour la détection de la classe *bouteille* à partir d'images polarimétriques brutes. PoTATO contient 12 380 instances annotées et se distingue par (i) la mise à disposition de données brutes issues d'un capteur à micro-réseau de polariseurs couplé à une matrice de Bayer, assurant une synchronisation intrinsèque, et (ii) la possibilité d'extraire plusieurs modalités cohérentes spatialement à partir d'une acquisition unique. Les données sont acquises depuis un ASV avec une perspective compatible avec la collecte (caméra frontale inclinée), sur plusieurs journées et conditions, et les ensembles d'entraînement/validation/test sont séparés par sessions afin de limiter les fuites temporelles.

Le chapitre décrit un pipeline d'extraction vers des modalités opérationnelles utilisées en apprentissage : **RGB**, **MONO**, **DOLP**, **POL** (pseudo-couleur DoLP/AoLP), **DIF** (intensité diffuse visant à réduire les reflets), et **PAULI**. L'accès aux données brutes est mis en avant comme un levier pour étudier l'impact des choix d'extraction (dématriçage, interpolation, fusion) sur les artefacts et sur la performance.

L'évaluation compare trois détecteurs après *fine-tuning* sur PoTATO (Faster R-CNN, YOLOv5, RetinaNet), entraînés séparément sur chaque modalité. Les résultats montrent une tendance stable : les entrées polarimétriques et hybrides apportent un gain lorsque les objets sont *moyens à grands* (typiquement proches du véhicule), tandis que **RGB** conserve un avantage relatif pour les *petits* objets lointains. La modalité **DIF** apparaît la plus régulière sur l'ensemble des tailles, ce qui en fait un choix pertinent pour une perception embarquée. Les courbes précision–rappel indiquent aussi que l'écart entre modalités augmente lorsque l'on exige une localisation plus précise, suggérant un bénéfice de la polarisation pour délimiter la cible.

Le chapitre discute enfin des limites observées : dominance d'objets très petits, artefacts liés au capteur à micro-réseau et aux vagues, et saturation sous fortes réflexions affectant l'estimation DoLP/AoLP. En conclusion, la polarimétrie est particulièrement adaptée aux scènes aquatiques où elle améliore la robustesse et la localisation près du robot. PoTATO fournit une fondation expérimentale réutilisée dans la suite, notamment pour générer des objectifs dans la chaîne intégrée, et prépare la transition vers la fusion pour la segmentation sémantique.

Chapitre 3 : Fusion polarimétrique et chromatique pour la segmentation

Ce chapitre passe de la détection à la segmentation sémantique (classification pixel-à-pixel). Pour la collecte de déchets, la segmentation peut améliorer la délimitation des objets et séparer cible et fond, mais les scènes aquatiques restent difficiles (reflets, éblouissement, illumination variable). Le chapitre étudie si des indices polarimétriques peuvent renforcer la segmentation en fournissant des informations complémentaires à la couleur (RGB).

La contribution est un benchmark reproductible sur PoTATO, via **PoTATO-Seg**. À partir des images brutes, trois modalités alignées sont considérées : **RGB**, **DIF** et **POL**. Des masques sont générés avec Segment Anything Model (SAM) [1] guidé par les boîtes, puis contrôlés et corrigés manuellement pour exclure les reflets lorsque c'est possible. Les images sont recadrées autour de la zone d'intérêt et normalisées, aboutissant à un corpus de 1 500 images partitionné en entraînement/validation/test.

Deux familles de modèles sont évaluées : (i) des baselines mono-modales (U-Net, DeepLabV3) pour isoler l'effet de la représentation d'entrée, et (ii) des architectures de fusion récentes (CMNeXt, MMSFormer, StitchFusion) combinant RGB avec des canaux *DoLP* et/ou *AoLP* (configurations **RGBD**, **RGBA**, **RGBAD**). L'augmentation de données est volontairement limitée afin de préserver la cohérence physique des signaux polarimétriques. Les métriques couvrent la qualité des régions (mIoU, F1) et la qualité de contour via $\mathrm{ASD}_{\mathrm{over}}$.

Les résultats montrent que, pour les baselines, **DIF** fournit le meilleur compromis global, confirmant l'intérêt de la suppression de l'éclat de surface. **POL** peut aussi être compétitive, notamment à faible contraste. Les réseaux de fusion modernes surpassent globalement les modèles classiques, et l'ajout de canaux polarimétriques réduit typiquement l'erreur de frontière, produisant des masques plus nets. Toutefois, les gains dépendent fortement de l'architecture : StitchFusion exploite le plus la complémentarité, tandis que d'autres schémas montrent des gains plus modestes.

Le chapitre souligne également des limites : lorsque le signal de polarisation est faible (géométrie défavorable, objets lointains), l'apport devient marginal; certains fonds peuvent induire de nouveaux faux positifs en fusion; et l'absence de pré-entraînement standard pour ces modalités rend l'apprentissage plus sensible sur un corpus limité. Enfin, le surcoût computationnel des variantes multi-canaux impose un arbitrage précision/latence pour l'embarqué. En conclusion, la polarimétrie est complémentaire à RGB pour la segmentation en environnement aquatique, mais, dans l'optique de génération de buts pour la collecte, le manuscrit privilégie ensuite la détection comme compromis performance/complexité, tout en ouvrant des perspectives (données plus vastes, pré-entraînement adapté, fusion plus légère).

Chapitre 4 : Cadre de simulation et d'apprentissage par renforcement

Ce chapitre pose les prérequis techniques pour l'apprentissage par renforcement (RL) appliqué à la navigation d'un ASV. Il souligne un verrou pratique : les simulateurs maritimes classiques sont utiles pour le prototypage mais limités pour l'entraînement DRL à grande échelle, faute de parallélisation efficace et d'accélération GPU. Ce constat motive un environnement combinant simulation hautement parallélisée et modélisation maritime pertinente afin de soutenir des politiques entraînées en simulation puis déployées sur robot réel.

Le chapitre justifie le choix de l'écosystème NVIDIA Isaac comme socle, notamment pour l'exécution d'un grand nombre d'environnements en parallèle avec une physique accélérée sur GPU et une intégration avec des bibliothèques RL modernes. Il explicite aussi le coût d'ingénierie associé à une plateforme en évolution rapide (changements d'API, migrations, maintenance), et l'importance de documenter précisément les versions utilisées pour garantir la traçabilité des résultats.

Une contribution centrale est l'extension du *Isaac Lab* par des modèles maritimes (flottabilité, hydrostatique/hydrodynamique, propulsion), implémentés de manière vectorisée pour préserver le débit de simulation nécessaire au DRL. Le chapitre met également en perspective l'usage

du framework au-delà des expériences principales via RoboRAN, un banc d'essai unifié orienté navigation basée RL (séparation robot–tâche, tâches et métriques uniformes, déploiement open-source via ROS 2). Enfin, il illustre un usage en sécurité et vérification (p. ex. NeuralPARC [2]) où la simulation sert à générer des trajectoires réalistes et riches en perturbations.

En conclusion, ce chapitre établit le socle de simulation et d'entraînement sur lequel reposent les politiques RL évaluées ensuite en conditions réelles, en articulant exigences scientifiques (robustesse, sim-to-real, évaluation systématique) et contraintes logicielles (modèles physiques, architecture, déploiement).

Chapitre 5 : Apprentissage par renforcement pour la navigation d'un ASV

Ce chapitre présente la méthodologie DRL pour piloter un ASV vers une cible flottante à capturer. Au-delà de l'atteinte de l'objectif, l'accent est mis sur la robustesse (écarts de modélisation, perturbations, bruit) et l'efficacité énergétique, contraintes déterminantes en mission réelle. Le cadre Isaac est utilisé pour entraîner efficacement des politiques continues (PPO) grâce à la parallélisation GPU.

Une contribution majeure est la combinaison d'identification système (SID) et de domain randomization (DR) pour réduire l'écart sim-to-real. Les paramètres hydrodynamiques et les courbes de poussée sont calibrés à partir d'essais réels, puis randomisés durant l'entraînement (paramètres de traînée, efficacité propulsive, bruit d'observation et d'action, forces/couples exogènes) afin d'obtenir des politiques plus généralisables. La tâche est formulée comme une capture en 2D dans un environnement sans obstacles, avec une portée et un champ de vision compatibles avec l'intégration perception–commande. La dynamique s'appuie sur un modèle inspiré de Fossen [3] en mouvement plan.

Le design RL est volontairement compact : observation dans le repère du robot ($\mathbb{R}^6$) et action différentielle sur les propulseurs. La récompense combine progression, alignement, pénalisation énergétique et critère terminal, afin de produire un comportement rapide et économe, compatible avec la capture par filet. La validation suit une chaîne à quatre étapes : identification réelle, entraînement sous Isaac, test dans un simulateur indépendant (Gazebo) pour limiter la dépendance au simulateur d'entraînement et faciliter l'intégration ROS, puis essais terrain (Kingfisher, inférence embarquée).

Quatre agents sont comparés (NV, NV-DR, SID, SID-DR). Les résultats en simulation montrent que les variantes avec SID et/ou DR convergent plus directement, et que la combinaison SID-DR fournit le meilleur compromis. Les essais terrain confirment qualitativement ces tendances malgré un volume d'essais contraint. L'analyse, basée sur le temps d'achèvement T_c et une énergie accumulée E_{acc} (proportionnelle à l'amplitude des commandes), indique que SID-DR améliore simultanément temps et énergie, tout en mettant en évidence l'écart attendu entre simulation et terrain sous perturbations naturelles.

En conclusion, ce chapitre montre qu'un cadre DRL accéléré par GPU, enrichi de modèles maritimes et couplé à SID + DR, permet un transfert sim-to-real efficace pour la navigation vers des cibles, et constitue la base des chapitres consacrés à la robustesse et à l'intégration perception–commande.

Chapitre 6 : Navigation robuste d'un ASV

Ce chapitre étend l'étude en se concentrant sur la robustesse des politiques de navigation, condition nécessaire au déploiement. Le système est non stationnaire : la charge collectée modifie masse, inertie et traînée, tandis que des perturbations externes imposent des efforts difficiles à mesurer. L'objectif est de caractériser, de manière systématique, comment ces changements affectent une politique DRL et de comparer la dégradation à celle d'un contrôleur de référence fondé sur modèle, un MPC. Plutôt qu'une comparaison directe des objectifs (coût MPC vs récompense RL), l'évaluation mesure la sensibilité aux perturbations.

Le chapitre introduit une procédure de *stress-test* combinant simulations à grande échelle, test dans Gazebo et essais terrain incluant perturbations naturelles et perturbations contrôlées. Le DRL repose sur une randomisation de domaine et sur un entraînement stabilisé (curriculum). Le MPC est implémenté sur un modèle 2D simplifié et résolu en ligne (horizon glissant) à une fréquence temps réel, fournissant une baseline interprétable.

Trois métriques structurent l'analyse : temps normalisé $T_{norm} = T/d$, énergie accumulée normalisée $E_{\text{acc}}^{\text{norm}}$, et excès de longueur de trajectoire Δ_d. En simulation, l'analyse se concentre sur les perturbations les plus impactantes, notamment le décalage du centre de masse (CoM) et la traînée rotationnelle (N_r). Les résultats montrent que la politique DRL maintient un taux de succès proche de 100% et des trajectoires stables sur une large plage de variations, tandis que le MPC est particulièrement sensible aux variations de CoM, avec chute de réussite et dégradation marquée des métriques.

Les essais terrain confirment la tendance : le DRL conserve globalement des trajectoires directes, alors que le MPC dérive davantage sous perturbations. Un compromis est mis en évidence : sous perturbation, le DRL tend à augmenter l'énergie pour compenser activement couple et traînée, tandis que le MPC limite davantage l'énergie mais accepte une déviation géométrique plus forte. Le chapitre documente enfin une limite pratique critique : des rotations proches et à grand angle sont plus fragiles en réel qu'en simulation, et l'analyse instrumentée révèle une limitation de taux dans la chaîne d'actionnement, induisant un délai cumulatif d'environ deux secondes sur des changements importants.

En conclusion, le chapitre propose un protocole de stress-test reproductible et montre que la politique DRL est plus robuste que le MPC face à certaines variations paramétriques majeures, tout en identifiant l'actionnement comme un facteur dominant d'écart sim-to-real, directement traité au chapitre suivant.

Chapitre 7 : Intégration perception–contrôle pour la collecte autonome

Ce chapitre présente une architecture intégrée validée sur le terrain pour la collecte autonome de déchets flottants par un ASV. Le système couple (i) une perception polarimétrique réduisant l'impact des reflets et (ii) un contrôleur RL léger, entraîné *uniquement en simulation* puis transféré *zero-shot*. L'objectif est d'identifier quantitativement les facteurs qui rendent le transfert robuste ou fragile, afin de prioriser les améliorations de fidélité là où elles sont déterminantes.

Le chapitre documente la plateforme (Kingfisher rétrofitée, RTK/IMU, calcul embarqué, caméra polarimétrique) et met l'accent sur la chaîne d'actionnement : les commandes transitent par un microcontrôleur imposant une limitation de pente (*rate limit*) pouvant induire un retard $\sim 2\,\text{s}$ lors de variations pleine échelle. Mesurée et reproduite en simulation, cette dynamique est intégrée à l'entraînement afin d'améliorer la transférabilité.

La perception repose sur YOLOv5 sur l'image DIF (*glare-suppressed*), puis sur la conversion des détections en un but plan $\mathbf{g}_r = [x_r, y_r]^\top$ dans le repère du robot. Une contribution méthodologique est l'usage d'une perception simulée déterministe réutilisant la calibration et un modèle projectif, permettant des expériences reproductibles et une séparation claire entre erreurs de perception et comportement du contrôleur. Une caractérisation embarquée situe la latence de perception autour de $\sim 250\,\text{ms}$ et motive une gestion explicite des timestamps.

Le module de commande est une politique PPO entraînée dans Isaac Lab avec randomisation (bruit, perturbations dynamiques, variations d'efficacité propulsive). Deux politiques sont comparées : une avec actionnement raffiné (rate limit inclus) et une naïve (sans rate limit), afin d'isoler l'impact de la fidélité d'actionnement. L'évaluation suit une campagne appariée simulation/terrain sur 14 régimes de perturbations. Les résultats indiquent une robustesse élevée dans la majorité des conditions, avec une vulnérabilité dominante : l'actionnement. En particulier, l'omission du rate limit à l'entraînement induit oscillations, hausse d'énergie et baisse de réussite, tandis que des perturbations perceptives importantes affectent faiblement la réussite et dégradent surtout l'efficacité.

Au-delà des objectifs programmés, le chapitre valide une boucle complète de recherche et capture avec perception réelle, via un arbitrage minimal entre motif de couverture et cibles détectées. Trois expériences structurent la progression et montrent la viabilité de bout en bout en conditions naturelles. Les limites observées proviennent principalement de la couche décisionnelle (gestion multi-cibles, sélection sans mémoire) et d'erreurs de projection à longue distance, tandis que la commande bas niveau demeure robuste.

En conclusion, le chapitre établit des principes pratiques pour le sim-to-real : améliorer en priorité les composantes auxquelles la politique est la plus sensible (ici, l'actionnement), utiliser une randomisation ciblée, conserver un simulateur indépendant pour révéler tôt les fragilités, et traiter explicitement la synchronisation temporelle. Il ouvre des extensions vers la planification (multi-cibles, mémoire, évitement) et la réduction des biais de projection à longue portée.

Chapitre 8 : Conclusion

Cette thèse a étudié le couplage d'une perception polarimétrique et d'un contrôle par apprentissage par renforcement pour une autonomie robuste d'un ASV dédié à la collecte de déchets flottants. L'originalité réside dans une approche end-to-end, depuis l'acquisition et la fusion de modalités visuelles jusqu'au pilotage différentiel et à l'intégration système, avec validation en simulation et en conditions réelles selon un protocole permettant de quantifier la robustesse et d'identifier les sources dominantes d'écart sim-to-real.

Sur le plan perceptif, la polarimétrie apporte une information complémentaire à RGB (DoLP, AoLP) qui améliore la détection et la segmentation dans des scènes fortement réfléchissantes, en réduisant l'éblouissement et en renforçant le contraste cible/fond. Un jeu de données dédié, du code et des baselines ont été publiés pour favoriser la reproductibilité. Les limites identifiées concernent la variabilité du signal polarimétrique, les artefacts des capteurs à division de plan focal et le risque de faux positifs en fusion lorsque certains fonds présentent des signatures proches des déchets.

Concernant le contrôle, la thèse montre qu'un transfert *zero-shot* vers le réel est atteignable en combinant identification système (paramètres dominants) et domain randomization (incertitudes résiduelles), avec des politiques légères et efficaces énergétiquement. Un point clé est l'importance d'une modélisation réaliste de l'actionnement : une non-idéalité telle qu'une limitation de taux peut dominer l'échec du transfert si elle n'est pas représentée à l'entraînement.

Au niveau système, une chaîne modulaire perception–commande a été validée sur le terrain, et un protocole d'évaluation multi-niveaux (simulation, simulateur indépendant, essais réels) a permis de relier métriques et causes d'échec. Les limites observées proviennent davantage des mécanismes de décision (sélection de buts sans mémoire, contraintes de champ de vision) que du contrôleur bas niveau.

Les perspectives portent sur (i) des données polarimétriques plus vastes et des méthodes d'extraction plus adaptées pour limiter artefacts et saturation, (ii) des architectures de fusion plus légères et mieux contraintes pour l'embarqué, et (iii) des extensions de l'autonomie vers des tâches plus réalistes (multi-cibles, évitement, planification), tout en conservant une démarche d'évaluation guidée par le terrain. En conclusion, les résultats montrent que l'autonomie embarquée dépend autant de l'innovation algorithmique que d'une intégration disciplinée et d'une validation expérimentale rigoureuse, indispensable pour révéler des modes de défaillance invisibles en simulation et prioriser les améliorations réellement impactantes.

1

Introduction

1.1 About the PhD dissertation topic

This dissertation aims to enhance perception and control solutions for real-world autonomous systems, focusing on the collection of floating waste from water surfaces. This application is driven by two main factors: First, floating debris poses a serious threat to marine ecosystems, so better detection and retrieval methods can yield a positive environmental impact. Second, our laboratory's established expertise in mobile robots offers a solid foundation for bridging theoretical advances to practical applications.

Together, these motivations led to the establishment of the R3AMA project, which includes three partners: (1) the SpaceR Lab at the University of Luxembourg, which has extensive experience in reinforcement learning for mobile robots; (2) IADYS (Interactive Autonomous DYnamic Systems), the industrial partner specialized in developing robotics solutions, including the Jellyfishbot—a multi-purpose autonomous surface vehicle (ASV) for collecting waste from water surfaces; and (3) the DREAM (Data-driven Robotics for Environment Assessment and Monitoring) Lab, housed within the CNRS International Research Lab 2958 at the Georgia Tech Europe campus with expertise in autonomy and data processing for robotic applications in natural environments.

R3AMA, which stands for Robust Reinforcement Learning for Real-World Autonomous Multitask Agents, seeks to improve how machine learning is applied to mobile robotics. This thesis focuses on the development of autonomous systems for collecting floating waste, a topic with broad opportunities for research in perception, control, and planning. Additionally, we emphasize extensive field testing, ensuring that these solutions are not only theoretically feasible but also robust and viable in real-world applications.

1.2 Research Questions

This section summarizes some of the challenges in developing an autonomous system for collecting floating waste and introduces the corresponding research questions. Here, we provide a comprehensive overview. A detailed background and related work are presented in later chapters.

To enable waste collection on water surfaces, it is first necessary to detect floating objects. Although deep learning has made significant progress in object detection [4], outdoor water environments pose unique challenges due to glare, variable lighting, and reflections. In particular, reflective water surfaces can produce false positives or occlude real objects, making detection with standard RGB cameras unreliable [5]. Because outdoor scenes exhibit strong light polarization,

this thesis proposes leveraging polarimetric cameras to supplement traditional vision and improve detection algorithms.

Question 1

How can we leverage polarimetric information to enhance the detection of floating waste in outdoor water environments?

Once the target object is located, the ASV must navigate to its position. Traditional approaches often depend on precise system dynamics models, which can be both complex to characterize and subject to change—for example, drag may vary as the robot captures floating waste. In addition, external disturbances such as water currents and wind can be difficult to measure without supplementary sensors. Deep Reinforcement Learning (DRL) has shown promise in addressing model inaccuracies and disturbances within robotic applications [6]. However, many DRL-based navigation algorithms for ASVs remain heavily simulation-focused, with limited real-world validation [7–9]. To address such challenges, this thesis aims to develop and evaluate RL-based control policies that can successfully capture floating waste in diverse real-world conditions.

Question 2

Is RL-based control capable of performing real-world ASV navigation despite dynamic model inaccuracies, sensor noise, and external disturbances?

Finally, for a fully autonomous solution, perception and navigation must be seamlessly integrated. In many ASV applications, traditional control strategies remain popular, with their evaluation often tied to specific task requirements [10]. By contrast, RL-based navigation approaches are frequently tested only in simulation or in isolation from a perception framework [11]. However, placing image-based perception in the control loop introduces coupled error sources such as detection uncertainty, calibration errors, time synchronization errors, and latency, which can degrade policy performance in the field. This thesis integrates perception and RL-based control in a modular stack and focuses on zero-shot sim-to-real transfer from policies trained only in simulation to field deployments. To reach this goal, we introduce a comprehensive evaluation protocol with clear end-to-end metrics and field validation.

Question 3

How well can a fully integrated pipeline, combining perception and an RL-based navigation policy, achieve zero-shot sim-to-real transfer, and what evaluation methodology reliably characterizes its field performance?

1.3 Contributions

The main contributions of this thesis include improved perception for water environments, integrated with an RL-based navigation policy, and validation through extensive field tests to assess its robustness and limitations. These core contributions are detailed below.

Polarimetric perception. A unique approach that leverages polarimetric cameras to address reflectivity and visibility challenges in water environments, improving the detection of floating waste.

Deep reinforcement learning for ASV navigation. An open-source RL framework extending state-of-the-art methods to autonomous surface vehicles, emphasizing adaptive policies that remain functional in dynamic, real-world conditions.

Integrated perception and control with zero-shot capability and sim-to-real evaluation. An integrated pipeline that couples polarimetric waste detection with an RL navigation policy in a modular stack, validated in the field. We introduce a comprehensive evaluation protocol defining metrics to assess robustness and prioritize improvements for reaching zero-shot sim-to-real policy transfer.

1.4 Outline

The chapters of this dissertation[1] are grouped into three thematic parts (Perception, RL-based Navigation, and Integration) and are illustrated in Figure 1.1. Because the thesis follows a publication-based format, most chapters are derived from published or submitted articles and are written to be self-contained, including their own introductions and related work. To limit redundancy when the thesis is read sequentially, overlapping background and methodology sections were reduced or explicitly signposted. However, some repetition remains so that readers can consult individual chapters independently. In particular, common related work for RL-based navigation was reduced across Chapters 4 to 7: Chapter 6 adopts the problem formulation and methodology introduced in Chapter 5 by cross-reference, while Chapter 7 provides additional detail and formalism. To reduce fragmentation, Chapter 7 retains a complete description of its methodology even where it overlaps with earlier chapters.

- **Chapter 1** introduces the motivation, research questions, and contributions of this thesis.
- **Chapter 2** presents the design and analysis of a novel dataset comprising polarimetric and chromatic images for object detection. It empirically demonstrates that polarimetric information provides strong discriminative features and, under certain conditions, can outperform traditional chromatic images for object detection.
- **Chapter 3** complements the perception analysis by evaluating polarimetric and chromatic fusion algorithms for semantic segmentation in complex outdoor aquatic environments.
- **Chapter 4** provides the context and rationale behind the development of a highly parallelized simulation and reinforcement learning framework tailored for maritime robotics applications, including examples of research articles where it was used.
- **Chapter 5** proposes a deep reinforcement learning methodology for ASV navigation, addressing the sim-to-real gap through system identification and domain randomization. The proposed method is validated through real-world field experiments.
- **Chapter 6** introduces a pipeline to evaluate the robustness of ASV navigation under controlled perturbations and real-world conditions. Additionally, it presents a Model Predictive Control (MPC) baseline.

[1] This dissertation is submitted under the Georgia Institute of Technology and Université de Lorraine doctoral programs; aside from institution-specific formatting requirements, the manuscript content is identical.

- **Chapter 7** presents the integration of the perception system and control policies into a fully autonomous floating waste collection pipeline, including the development process, the evaluation protocol for zero-shot sim-to-real transfer, and robustness analysis.
- **Chapter 8** provides concluding remarks and perspectives on future work.

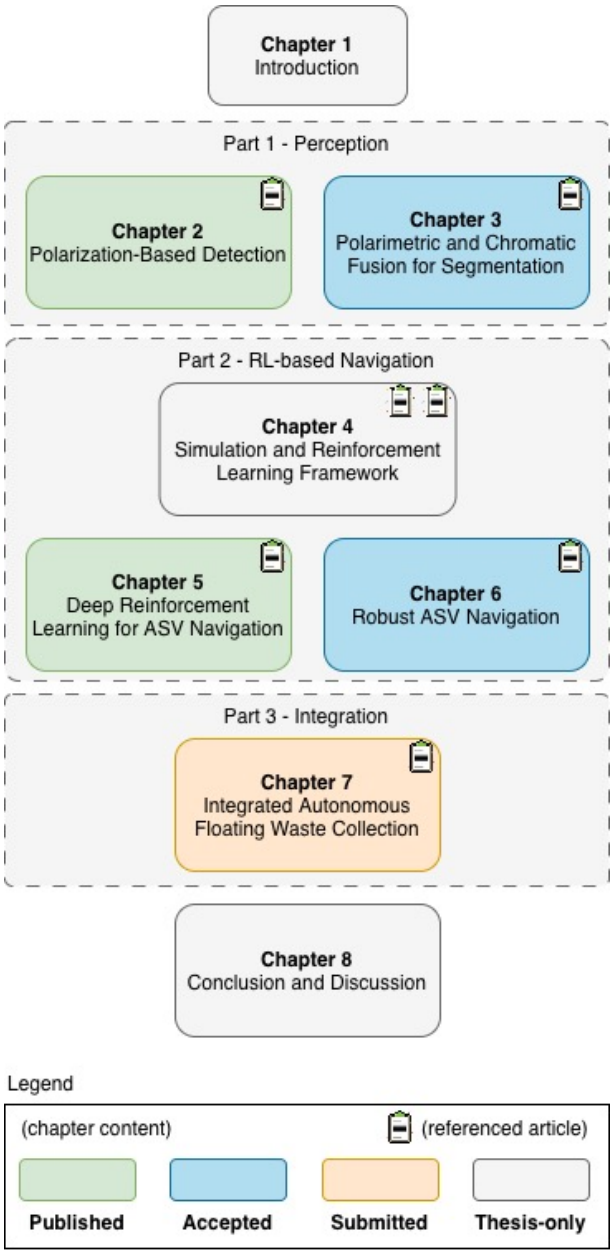


Figure 1.1: Structure of the dissertation. The colors indicate whether the content of each chapter has been used in a research article or has been written specifically for the thesis.

1.5 List of Publications

Published

- **Batista, L.F.W.***, Ro, J.*, Richard, A., Schroepfer, P., Hutchinson, S. and Pradalier, C. "A Deep Reinforcement Learning Framework and Methodology for Reducing the Sim-to-Real Gap in ASV Navigation". In: 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2024, pp. 1258–1264.
- **Batista, L.F.W.**, Khazem, S., Adibi, M., Hutchinson, S. and Pradalier, C. "PoTATO: A Dataset for Analyzing Polarimetric Traces of Afloat Trash Objects". In: Computer Vision – ECCV 2024 Workshops. Ed. by A. Del Bue et al. Cham: Springer Nature Switzerland, 2025, pp. 190–205.
- Chung, L.K., Jung, W., Pullabhotla, S., Shinde, P., Sunil, Y., Kota, S., **Batista, L.F.W.**, Pradalier, C. and Kousik, S. "Guaranteed Reach Avoid for Black-Box Systems through Narrow Gaps via Neural Network Reachability". In: 2025 IEEE International Conference on Robotics and Automation (ICRA). 2025.
- El-Hariry, M.*, Richard A.*, Castan, R.M.*, **Batista, L.F.W.***, Geist, M., Pradalier C., Olivares-Mendez M. "RoboRAN: A Unified Robotics Framework for Reinforcement Learning-Based Autonomous Navigation". Transactions on Machine Learning Research, 2025.

Accepted

- **Batista, L.F.W**., Aravecchia, S., Hutchinson, S. and Pradalier, C. "Evaluating Robustness of Deep Reinforcement Learning for Autonomous Surface Vehicle Control in Field Tests". In: 2025 Workshop on Field Robotics (ICRA). 2025
- **Batista, L.F.W.***, Bourbon, T.* and Pradalier, C. "Evaluation of Polarimetric Fusion for Semantic Segmentation in Aquatic Environments". In: 2025 IEEE International Conference on Visual Communications and Image Processing (VCIP). 2025.

In preparation

- **Batista, L.F.W**, Aravecchia, S, and Pradalier, C. "Sim-to-Real Transfer and Robustness Evaluation of Reinforcement Learning Control with Integrated Perception on an ASV for Floating Waste Capture". 2025.

Part I

Perception

2

Polarization-Based Detection

Autonomous robots can be deployed to collect floating waste, but they require accurate object identification. While deep learning has been widely used for this task, its performance is often constrained by outdoor illumination and water-surface reflections. Light polarization, abundant in such environments yet invisible to the human eye, can be captured by modern polarimetric cameras to improve litter detection accuracy on water surfaces. With this goal in mind, this chapter introduces PoTATO, a dataset of raw, pixel-aligned polarimetric images with 12,380 labeled bottle instances. We characterize the conditions under which polarization enhances object detection and, by releasing raw images, enable exploration of novel detection approaches. This chapter provides the basis for the polarimetric perception used in the remainder of this thesis, including for goal generation in the integrated system. Code and data are publicly available at `https://github.com/luisfelipewb/PoTATO/tree/eccv2024`

This chapter closely follows the following peer-reviewed workshop paper [12]:

- **Batista, L.F.W.**, Khazem, S., Adibi, M., Hutchinson, S. and Pradalier, C. "PoTATO: A Dataset for Analyzing Polarimetric Traces of Afloat Trash Objects". In: Computer Vision – ECCV 2024 Workshops. Ed. by A. Del Bue et al. Cham: Springer Nature Switzerland, 2025, pp. 190–205. [2]

Compared with the original article, this chapter contains minor modifications to improve its integration with the thesis.

[2] *Statement of contributions:* Luis contributed to developing the idea of using polarimetric cameras, creating and labeling the dataset, and conducting both theoretical investigations and experiments. Salim offered continuous input during the research and manuscript editing. Mehran provided valuable discussions and feedback on the ideas. Seth and Cedric provided guidance and feedback throughout the research process.

2.1 Introduction

Plastic pollution is a global threat known for its capability to damage aquatic life, ecosystems, and even human health [13]. Plastic is inherently designed to last a long time, but the weathering processes cause fragmentation into smaller particles that can reach remote locations on the planet [14]. The plastic accumulation in such areas is poorly reversible because the cleanup actions are infeasible, and the natural removal process is slow. According to Chamas et al. [15], the only known solution to decrease plastic accumulation is to reduce its emissions.

To prioritize actions that can prevent plastic from reaching the natural environment, it is necessary to identify the transport mechanisms and how it reaches the ocean. van Emmerik and Schwarz [13] have identified a lack of thorough understanding of how plastic is transported from land to aquatic systems, especially going through rivers. Most observation-based studies rely on existing infrastructure, such as bridges, and have limited coverage for a detailed understanding of how trash is reaching the ocean. Such systems can aid in monitoring the trash flow, but are rarely capable of collecting it.

Machine learning, specifically through object detection, has been widely utilized to automate the quantification of plastic in bodies of water [16]. The field of object detection, considered one of the most fundamental tasks in computer vision, has made remarkable progress over the past decade, with neural networks emerging as a powerful tool for identifying and classifying objects in various contexts [4]. Despite the advancements in object detection techniques, challenges remain, especially in outdoor environments where lighting conditions can vary significantly, creating bright sunlight, shadows, glare, and reflections on bodies of water. To address these challenges, the fusion of sensors of different modalities has been explored in different studies [5, 17].

Modern camera sensors can have the Bayer array integrated with microgrid polarizers, allowing capture of polarized and color images simultaneously [18]. This technology has led to increased interest in the computer vision and robotics research community to leverage polarimetric information for enhanced perception systems. According to Andreou et al. [19], utilizing polarization sensitivity can aid object detection by increasing the visual contrast of concealed objects. This is particularly relevant in outdoor environments and bodies of water, where two natural sources of polarization are abundant: scattered light and surface reflections [20]. Integrating polarimetric data into object detection has the potential to improve system performance and reliability.

The main contributions of our work are:

1. The first labeled dataset with raw, pixel-wise-aligned polarimetric and chromatic data for object detection.

2. An analysis of polarized light's key physical properties in outdoor, water-rich environments, highlighting its potential for object detection.

3. A baseline comparison of three well-known object detection algorithms across six image modalities, showcasing scenarios where polarized images surpass color images in detection efficacy.

The PoTATO dataset offers value beyond the object detection problem due to its versatility. Providing the raw data allows for research across various domains, including microgrid polarized image demosaicking and multimodal fusion.

2.2 Related Work

Deep learning-based object detection on water surfaces. Over the past three years, there has been a significant increase in the number of publications utilizing AI techniques on marine macrolitter datasets [16]. Despite the recent advancements, a lack of deep learning models with satisfactory generalization capability to detect trash on bodies of water has been identified, and further research is necessary to develop robust models that can overcome the challenges posed by the diverse geographical and environmental conditions [21]. Additionally, the vast majority of existing approaches rely on satellite images, drone images, or cameras fixed on bridges. While these are valuable for monitoring and quantifying litter, they cannot be easily integrated into robotic systems that are capable of autonomously collecting the identified waste.

The detection of floating objects is often hindered by challenges such as the varying outdoor illumination conditions and sun glare, and the reflections on the water's surface, as highlighted in [22]. To address these challenges, recent research has focused on the fusion of sensors employing different modalities. For example, in [23], a combination of a long-wave infrared polarimeter camera and a visible wavelength optical camera was utilized, while in [5], a millimeter wave radar was integrated with a camera. However, synchronizing and aligning measurements from different sensors introduces additional complexity and can potentially impact the performance of the model.

Object Detection using polarized images. The development of microgrid polarization sensors has leveraged polarized images for various applications, including 3D reconstruction [24], pose prediction [25], and reflection separation [26]. More specifically for object detection, a deep learning approach utilizing polarization images was proposed for car detection, and the results showed promise in addressing illumination and reflection issues [27]. In a subsequent study, Blin et al. compared models trained independently on a range of polarimetric encoded inputs and RGB images, demonstrating improved results compared to RGB-based detection [28]. Next, various multimodal fusion schemes and combinations of chromatic and polarimetric features were investigated [29]. While the results are encouraging, fusion of two independent sensing modalities is more challenging in the absence of pixel-wise correspondence, since no dense spatial alignment can be assumed between the images.

Recently, polarization information has been successfully combined with grayscale intensity to improve the results of semantic segmentation of transparent objects [30]. In their work, an attention-based early-fusion approach is proposed, but it does not use color information, and the indoor setting lacks the complexity of the outdoor illumination conditions.

Current research shows the strong potential of using polarized images for object detection in adverse illumination conditions, but lacks a public dataset including raw data and pixel-wise alignment.

Related datasets. Environmental protection has been receiving increased attention, resulting in the proposal of several datasets targeting litter detection. The analysis performed by Politikos et al. [16] provides a database for multiple datasets. Nonetheless, very few images are captured from the point of view of the vessel, making them hard to use in robotic systems capable of collecting waste autonomously. Vessel-mounted cameras have already been employed to quantify the presence of plastic in the ocean [31], and a large dataset comprising colored images was utilized. Nonetheless, public access to it was not identified.

The Flow-Img dataset [5] presents 2000 real-world labeled images captured from an Autonomous Surface Vehicle (ASV) and shares a similar application. Their study demonstrates that existing alternatives, such as TrashNet [32] and TACO [33], do not effectively generalize to complex environments and diverse perspectives and emphasize the need for expanding datasets

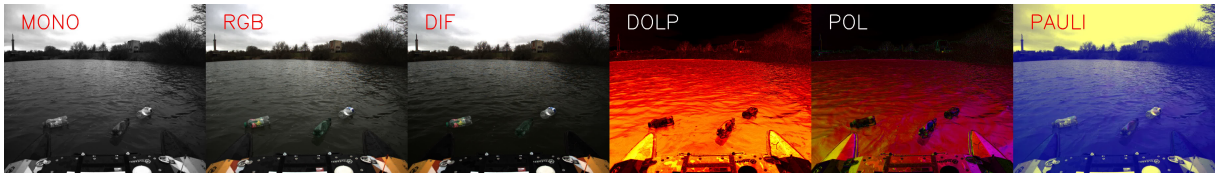


Figure 2.1: Different modalities that can be extracted from the dataset: Grayscale (MONO), Color (RGB), Color image with diffuse-only reflections (DIF), Degree of Linear Polarization (DOLP), pseudo-color image combining degree and angle of polarization (POL), and pseudo-color Pauli-inspired image (PAULI)

containing images captured from the point of view of the vessel.

The PolarLITIS dataset [17] was used specifically to compare the performance of color and polarization images. However, it is intended for road scenes without abundant water surface reflection, and the data was acquired using different sensors, leading to pixel-wise misalignment.

To the best of our knowledge, PoTATO is the first dataset containing raw polarized and colored images with pixel-wise alignment for object detection.

2.3 PoTATO Dataset

To acquire the images, the camera Triton $5.0MP$ TRI050S1-QC was used. This camera utilizes the SONY sensor IMX264MYR, capable of capturing colored images and polarization with a resolution of 2448×2048 pixels. It was paired with a lens with a focal length of $6mm$, providing an angle of view of $80.8°(H) \times 61.6°(V)$. The camera was mounted on a Kingfisher ASV from Clearpath Robotics. It was positioned facing forward with a downward angle to achieve the best field of view for observing the floating objects in the water up to the vessel's approach, as shown in Figure 2.2.



Figure 2.2: Polarized camera mounted on the Kingfisher ASV

Data collection took place on Lake Symphonie across seven distinct days, each characterized by varying weather as detailed in Table 2.1. Each recording session comprised several short clips to capture a broad spectrum of conditions and ensure variability of the polarimetric information. By approaching bottles from various directions and relocating them across the lake, we ensured a diversity of backgrounds, lighting, and relative sun positions. The boat was operated manually, and the images were recorded at two frames per second. Due to the variable light conditions in the outside environment, the auto-brightness function of the camera was enabled. All images containing bottles also include varied background features. Since background-only images

Table 2.1: Dataset Statistics and Weather Conditions

Day	Images	Labels	Weather
01	27	81	Sunny
02	114	414	Sunny
03	462	1450	Sunny
04	1658	4392	Sunny
05	902	2096	Partially Cloudy
06	459	787	Cloudy
07	978	3160	Cloudy
	4600	12380	

minimally enhance detection, the dataset exclusively comprises images with at least one bottle. To allow flexibility and enable the exploration of different approaches, we recorded our dataset in the raw image format with a resolution of 2448×2048 pixels, and six different modalities depicted in Figure 2.1 were extracted using the pipeline explained in Section 2.4.2.

The dataset has a single class *bottle* and annotation was performed using the Label Studio tool [34] using machine-learning-aided pre-annotations. The bounding boxes were drawn tightly, encapsulating only the bottle and not including its reflection on the water surface. During labeling, the RGB images were used due to its easy interpretability to the annotator. Nonetheless, the raw image format potentially contained more features that could allow identifying objects that were not visible on the colored image only. To avoid introducing this bias in the dataset, the annotation process was carried out carefully only by individuals who participated in the image recording. This allowed using the previous knowledge of the locations of the bottles to minimize the chance of False Negatives and False Positives in the ground truth. In addition to that, the images were labeled sequentially, enabling the annotator to leverage temporal information to create precise annotations.

The dataset was split into train, validation, and test sets with 2000 (43.5%), 600 (13.0%), and 2000 (43.5%) images, respectively. In our experiments, we kept a high number of images in the test set to increase the statistical significance of our evaluations. This was important due to the high variety of illumination conditions present in the dataset. During the split of the dataset, we considered that consecutive images can have higher similarity, and to prevent any leak from training to the test set, the images from each recording session were split sequentially into train, validation, and test sets. Each image contains between 1 and 8 labels. In total, the dataset has $12,380$ annotated plastic bottles, and the bounding box size distribution is shown Figure 2.3.

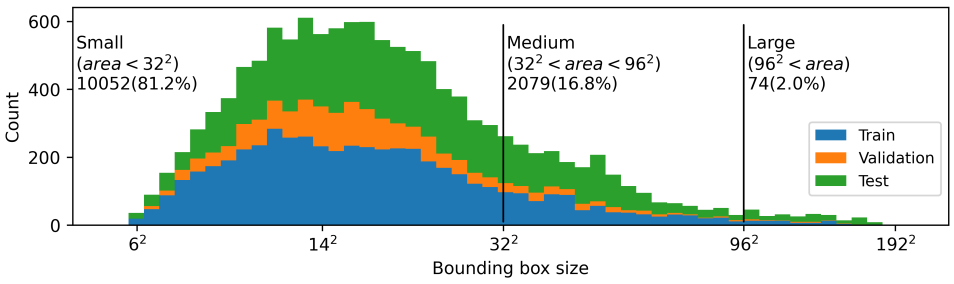


Figure 2.3: Size distribution of bounding boxes.

2.4 Overview of Light Polarization Theory

This section provides a theoretical background on light polarization and its application to the PoTATO dataset. We first discuss natural sources of polarized light in aquatic environments and their impact on image formation. Next, we explore the mathematical framework for extracting polarimetric data from raw sensor measurements. Finally, we present an overview of the pipeline used to obtain the desired polarimetric visualizations.

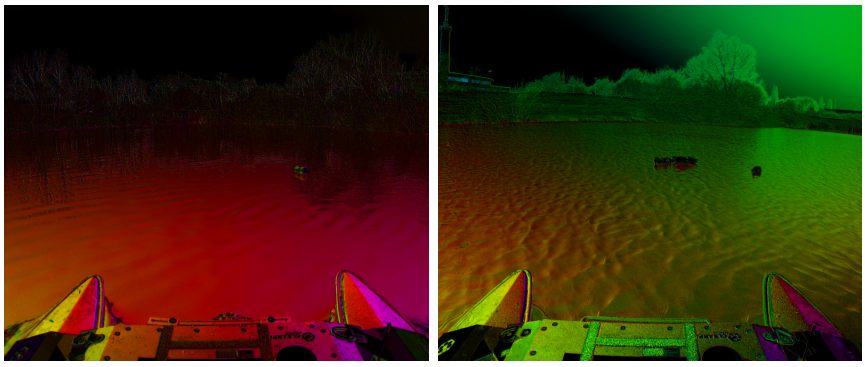
2.4.1 Sources of Polarization

Water bodies in outdoor environments have an abundance of two natural sources of polarized light: reflection and skylight scattering [20]. The first occurs when the light reflected on a dielectric material becomes linearly horizontally polarized, and its degree of polarization depends on the angle of incidence relative to the surface normal. The angle where the maximum polarization occurs is known as Brewster's angle (θ_{i_B}) and, from Snell's Law, it can be related to the refractive index of the mediums with the simple expression shown in Equation (2.1). [35]

$$\theta_{i_B} = \arctan(\frac{\eta_{water}}{\eta_{air}}) \tag{2.1}$$

Given the refractive index of water and air $\eta_{air} = 1$ and $\eta_{water} \approx 1.33$, the Brewster's angle θ_{i_B} can be calculated and has a value of approximately 53°. The camera is installed in the vessel at a height of $75cm$, therefore, the region with the strongest polarization by reflection is located around $1m$ distance. Polarization by reflection is predominant in cloudy weather and shown in Figure 2.4a.

The second occurs when light enters the atmosphere and scatters toward an observer through interactions with atoms and molecules. The degree of polarization increases with angular deviation, reaching its peak at 90°. This phenomenon is known as Rayleigh scattering and produces diverse polarization patterns depending on the sun's position relative to the observer [20]. Skylight polarization is abundant under clear skies and is shown in Figure 2.4b.



(a) Polarization by reflection on the water surface during a cloudy day

(b) Rayleigh scattering of skylight during a day with clear sky

Figure 2.4: Main natural sources of light polarization

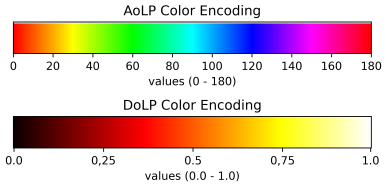
2.4.2 Extracting Polarimetric Data

The PoTATO dataset provides the raw data obtained from the sensor in a single-channel image. The sensor features a 2×2 microgrid polarizer capable of measuring the intensity of light

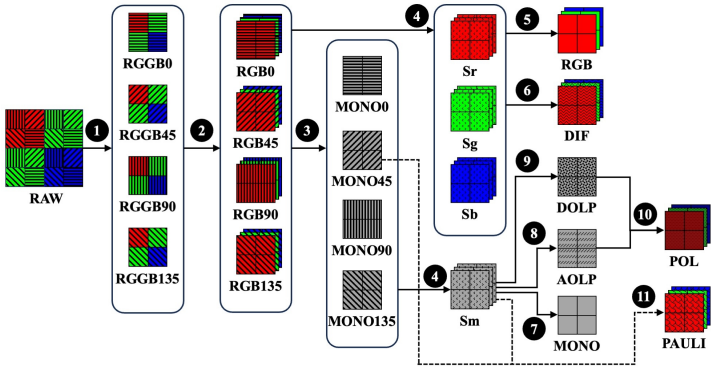
after it has passed through a linear polarizer oriented at four distinct angles $(0°, 45°, 90°, 135°)$. These intensity measurements are denoted as I_0, I_{45}, I_{90}, and I_{135}, respectively, and provide the necessary data to calculate the Stokes parameters. By superimposing the polarization filter with the Bayer array, a superpixel structure is created, allowing the measurement of polarization for multiple colors (Figure 2.5a). This integration with varying spatial positions of pixels introduces the possibility of edge artifacts [36] but ensures completely synchronized measurements, with high stability.



(a) Super Pixel



(b) Pseudo color encoding



(c) Extraction Pipeline

Figure 2.5: Super Pixel, Extraction Pipeline and Color Encoding

The Stokes parameters, which are calculated using the intensity measurements mentioned above, provide a complete description of the polarization state of the electromagnetic radiation [35]. These parameters are derived from specific combinations of the measured intensities (Equation (2.2)) and are critical for understanding the polarization properties of the light in the image. As the circular polarization is rare in nature and the parameters I_L and I_R (left and right circular polarization intensity) cannot be measured by this sensor, the Stokes parameter S_3 is considered to be zero and omitted in subsequent equations.

$$\begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \end{bmatrix} = \begin{bmatrix} I_0 + I_{90} \\ I_0 - I_{90} \\ I_{45} - I_{135} \\ I_R - I_L \end{bmatrix} \tag{2.2}$$

Using the Stokes parameters, it is possible to derive additional parameters. In our study, we utilize the Degree of Linear Polarization ($DoLP$), Angle of Linear Polarization ($AoLP$), and the Intensity of Diffuse Reflection (I_{dif}), with their respective equations extracted from [35]. The $DoLP$ indicates the proportion of linearly polarized light with values ranging from 0 (completely unpolarized) to 1 (completely polarized) Equation (2.3). The $AoLP$ represents the orientation of the electric field vector with values ranging from 0° to 180° (Equation (2.4)). Finally, I_{dif} can be estimated through Equation (2.5). It filters out specular reflection by considering that polarization is created by reflection. This technique allows for reducing the intensity of reflections on the water surface.

$$DoLP = \frac{\sqrt{S_1^2 + S_2^2}}{S_0} \tag{2.3}$$

$$AoLP = \frac{1}{2} arctan(\frac{S_2}{S_1}) \tag{2.4}$$

$$I_{dif} = \frac{S_0 - \sqrt{S_1^2 + S_2^2}}{2} \tag{2.5}$$

2.4.3 Implemented Pipeline

In our work, we implemented the pipeline shown in Figure 2.5c for extracting six distinct visualizations for color and polarimetric data. Nonetheless, the raw data available in the PoTATO dataset enables the extraction of metrics through various approaches. In the subsequent sections, we elaborate on the main steps we implemented and the corresponding extracted visualizations.

STEP 1: The RAW image is split into four $RGGB$ images, each corresponding to a specific polarization angle ($RGGB_0$, $RGGB_{45}$, $RGGB_{90}$, $RGGB_{135}$). This operation reduces the resolution by half to 1224×1024.

STEP 2: The four $RGGB$ channels are debayered to generate the corresponding RGB images (RGB_0, RGB_{45}, RGB_{90}, RGB_{135}).

STEP 3: A color conversion is performed to extract grayscale images for each of the four polarization angles, providing channels (M_0, M_{45}, M_{90}, M_{135}).

STEP 4: Extracting the Stokes parameters is accomplished by applying Equation (2.2) to each of the four channels (red, blue, green, monochrome), yielding four sets of Stokes vectors (S_R, S_G, S_B, S_M).

STEP 5: The intensity parameters from the Stokes vectors ($S_{R_0}, S_{G_0}, S_{B_0}$) are used to generate the **RGB** visualization, providing a color representation of visible light that can be captured by regular cameras.

STEP 6: By applying Equation (2.5) to the three Stokes vectors (S_R, S_G, S_B), the **DIF** representation is generated. This enables filtering out of specular reflections and focusing on diffuse light. It is useful to reduce the bright reflections on the water's surface regardless of its angle of polarization, achieving greater flexibility compared to linear polarization filters fixed in front of the camera lens.

STEP 7: The first Stokes parameter (S_{M_0}) is selected from the monochrome Stokes (S_M) to generate a visualization referred to as **MONO**. This representation shows the intensity of visible light in a grayscale image.

STEP 8: The Angle of Linear Polarization (AoLP) is extracted from the monochrome Stokes vector S_M using Equation (2.4). This step assumes that the angle of polarization does not vary significantly based on color.

STEP 9: The Degree of Linear Polarization (DoLP) is obtained by utilizing Equation (2.3) on the monochrome Stokes vector (S_M). This step also assumes negligible variation of DoLP across different wavelengths. The DoLP values are then normalized between 0 and 255, and the color encoding presented in Figure 2.5b is applied to generate the **DOLP** visualization.

STEP 10: The DoLP and AoLP information are combined to generate a visualization referred to as **POL**. This image utilizes HSV encoding, with DoLP as the value, AoLP as the hue, as shown in Figure 2.5b, and saturation equal to 1. Such representation aids in visualizing the polarimetric data captured by the sensor, enabling rapid identification of the Angle of Linear Polarization, while simultaneously suppressing the meaningless and noisy AoLP information from regions where the light is not polarized.

STEP 11: The **PAULI** visualization combines the S_{m_1}, I_{45}, and S_{m_0} channels as RGB images, mixing light intensity and polarimetric information. This approach draws inspiration from the Pauli decomposition method applied to polarimetric information from polarization-encoded SAR images as presented by Blin et al., [37].

2.5 Experiments and Analysis

This section presents the experimental analysis using different object detection models. We show that off-the-shelf pre-trained object detection models don't perform well on this dataset and require fine-tuning. Experimental results indicate the superiority of polarimetric-based inputs for medium and large objects, while RGB performs better for small objects. Finally, we also explore these observations through qualitative analysis and discuss the main challenges.

2.5.1 Object Detection Baseline

To establish a baseline for the PoTATO dataset, we conducted two sets of experiments: one using an off-the-shelf pre-trained model, and another using models fine-tuned specifically on our dataset. The evaluation employed commonly used metrics from the COCO object detection challenge [38], and all metrics were derived from the test set with 2000 images.

First, we experimented with YOLOv5 [39] using pre-trained weights on the COCO dataset [38], which already includes the *bottle* class. The results in Table 2.2 indicate that the model trained on general object detection datasets does not generalize well for detecting floating bottles on the water surface, particularly when the objects appear small in the image. This underscores the unique challenges presented in our dataset and the need for more effective detection approaches. Notably, even without fine-tuning, the DIF channel shows significantly better performance when compared to other channels, suggesting that reducing reflections significantly enhances detection performance, particularly for large objects.

Table 2.2: COCO Metrics (AP at IoU=.50:.05:.95) with pre-trained weights without fine-tuning. While the DIF channel yields the highest AP, particularly for large objects, overall detection performance remains limited.

	Channel	AP	APs	APm	APl	ARs	ARm	ARl
Yolo-v5 m	MONO	0.009	0.000	0.008	0.150	0.000	0.001	0.155
	RGB	0.019	0.000	0.025	0.350	0.000	0.022	0.383
	DIF	**0.025**	0.000	**0.055**	**0.412**	0.000	**0.052**	**0.440**
	DOLP	0.009	0.000	0.000	0.025	0.000	0.000	0.021
	POL	0.005	0.000	0.004	0.017	0.000	0.001	0.017
	PAULI	0.010	0.000	0.008	0.126	0.000	0.007	0.125

For the second set of experiments, we fine-tuned three models to each one of the six different channels of our dataset using the training set with 2000 images. They were evaluated independently, and the results are presented in Table 2.3.

In addition to YOLOv5, we also included two other widely-used object detection models: Faster R-CNN and RetinaNet from Detectron2 [40]. These models were chosen for their diverse architectures: Faster R-CNN is a two-stage detector with an efficient region proposal network, YOLOv5 represents a one-stage detection paradigm, and RetinaNet, another one-stage detector with a modified focal loss function. This variety enabled us to assess the consistency of performance differences across different input images. We used ResNet-50 [41] with pre-trained weights as a backbone, and fine-tuned on our dataset. For consistency, we fine-tuned the three models with similar parameters, including training for 100 epochs and a batch size of 8. It is important to emphasize that augmentation was not used for preserving the correctness of the

Table 2.3: COCO Metrics (AP at IoU=.50:.05:.95) after fine tuning on PoTATO dataset. DIF and POL images have better results for medium and large bounding boxes. RGB image, performs better on small bounding boxes.

	Channel	AP	APs	APm	APl	ARs	ARm	ARl
Faster R-CNN	MONO	0.448	0.416	0.555	0.465	0.476	0.617	0.493
	RGB	**0.482**	**0.446**	0.599	0.483	**0.504**	0.651	0.531
	DIF	0.477	0.434	0.609	**0.550**	0.492	0.658	**0.609**
	DOLP	0.418	0.357	0.592	0.486	0.424	0.646	0.539
	POL	0.428	0.361	**0.627**	0.506	0.426	**0.678**	0.579
	PAULI	0.446	0.411	0.557	0.497	0.468	0.622	0.529
RetinaNet	MONO	0.362	0.309	0.512	0.481	0.424	0.593	0.520
	RGB	**0.394**	**0.323**	0.564	0.523	**0.449**	0.633	0.562
	DIF	0.388	0.303	0.593	**0.537**	0.423	0.653	0.575
	DOLP	0.321	0.229	0.560	0.503	0.352	0.631	0.558
	POL	0.345	0.244	**0.597**	0.506	0.367	**0.655**	**0.584**
	PAULI	0.368	0.304	0.537	0.468	0.419	0.610	0.496
Yolo-v5 m	MONO	0.402	0.381	0.500	0.376	0.466	0.592	0.455
	RGB	**0.461**	**0.431**	0.578	0.457	**0.517**	0.662	0.528
	DIF	0.459	0.425	0.589	**0.574**	0.509	0.662	**0.632**
	DOLP	0.400	0.357	0.560	0.464	0.442	0.633	0.566
	POL	0.429	0.386	**0.597**	0.391	0.478	**0.674**	0.527
	PAULI	0.450	0.418	0.572	0.379	0.504	0.669	0.453

physical properties of light in the polarized channels. For single-channel images (MONO and DOLP), we used channel replication to match model input.

After fine-tuning on our dataset, substantial improvements were observed across all channels and models. The results highlight the advantages of polarimetric-based inputs, particularly POL and DIF images, which perform better in scenarios with medium and large bounding boxes. Conversely, RGB images excel when detecting small bounding boxes. Notably, the POL input, derived purely from polarimetric data, often surpasses RGB input, emphasizing the effectiveness of polarimetric modalities in object detection. It is worth noting that the AP metric, computed by averaging ten different IoU thresholds, is heavily influenced by the imbalanced distribution of bounding box sizes shown in Figure 2.3.

Overall, our results indicate that the DIF channel, which simultaneously leverages chromatic and polarimetric information, is the most effective choice for detection applications. It demonstrates high performance across all bounding box sizes, making it particularly suitable for detecting bottles on the water surface.

2.5.2 Advantages of Polarimetric Information

Our experiments indicate that POL and DIF images show consistently increased performance for the images in which the bottles are closer to the camera, while the RGB images provide better results when the bottles are farther away. Moreover, the accuracy gap between polarimetric-based and chromatic-based inputs increases with higher IoU thresholds, revealing that polarimetric data is more robust in precisely detecting the position of the bottles.

Such observation derives from Figure 2.6, where we selected the three better performing channels: RGB, POL, and DIF for the Faster R-CNN model and plotted the Precision-Recall curves for three different IoU thresholds. Faster R-CNN was selected due to its higher scores, and similar results are also observed on the other two models. The outcome is expected given that the region closer to the vessel has stronger polarimetric signals due to the height of the camera position and Brewster's angle for the air/water interface, explained in Section 2.4. In longer distances, the angle of incidence gets larger, and the intensity of the polarimetric information decreases. In this scenario, the RGB tends to perform better for the detection of the bottles.

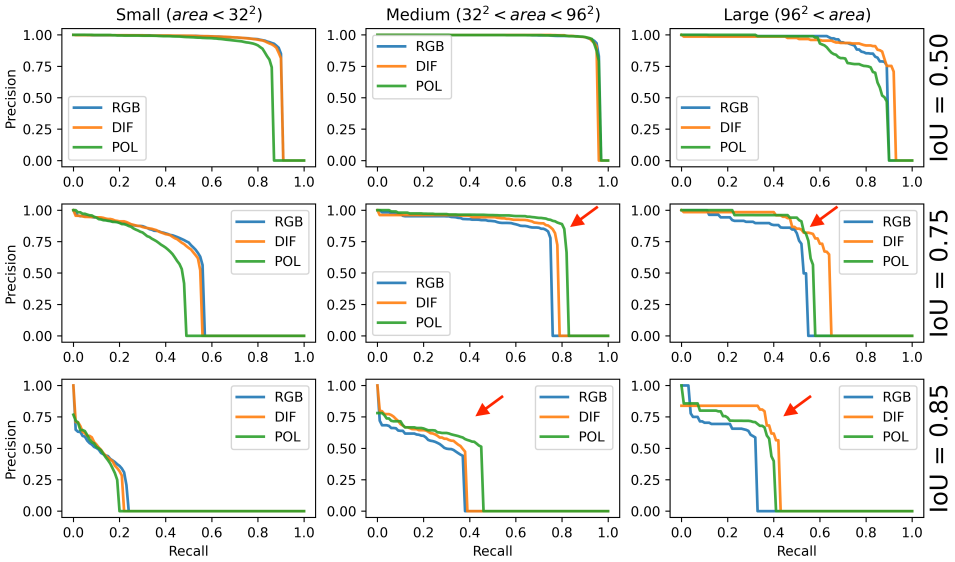
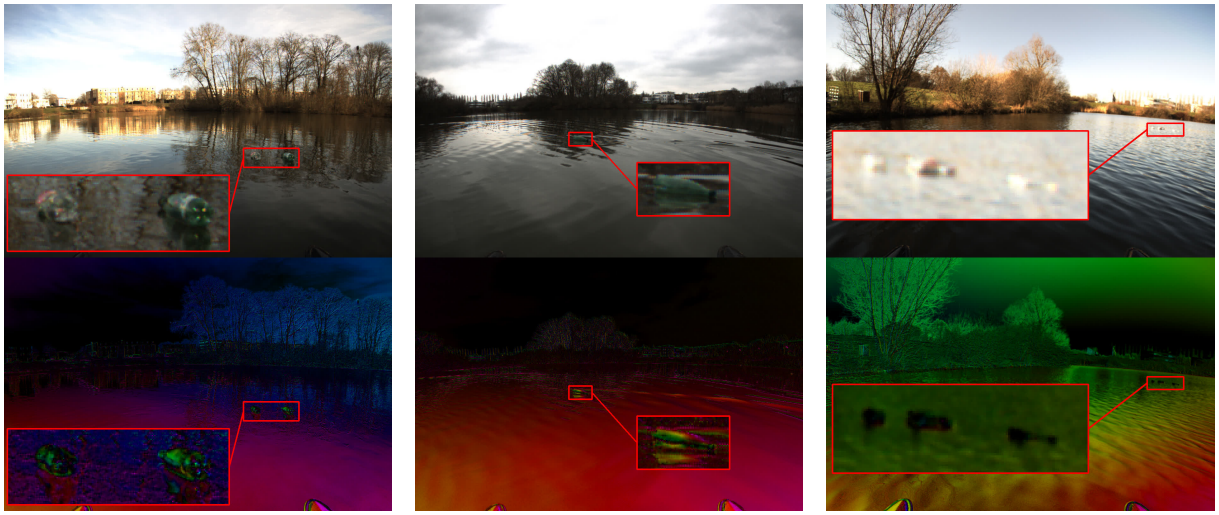


Figure 2.6: Precision-Recall curves for Faster R-CNN show that polarimetric-based information (POL and DIF) presents better results in medium and large objects as the IoU threshold increases.

The relationship between bounding box size and distance is attributed to the position of the forward-facing camera fixed on the vessel. As a result, the bounding box size variation primarily derives from perspective geometry, with closer objects appearing larger on the image plane.

2.5.3 Qualitative Analysis

To illustrate the difference in the features conveyed by the polarimetric and chromatic images, three images were selected. First, Figure 2.7a showcases bottles floating amidst overlapping tree reflections. In the RGB image, it is challenging to distinguish them from the background. However, in the corresponding POL image, the distinct Angle of Linear Polarization (AoLP) is emphasized, exhibiting stronger contrast in a different color. Second, Figure 2.7b presents a green bottle nearly concealed by the water with a similar color, while the POL image effectively accentuates the contrasting AoLP. This image was captured on a cloudy day, where the skylight polarization is blocked by the clouds, and the predominant source of polarization is the reflection on the water's surface. Third, Figure 2.7c illustrates a scenario where plastic bottles are situated in a well-lit area of the lake. The RGB image captures intense light reflection, indicating the presence of an object, but fails to clearly delineate its boundaries. The corresponding POL image shows sharply delineated edges of the bottles, highlighting the contrast with the water surface.



(a) Object Concealed in Background Reflection.

(b) Color-based Concealing During a Cloudy Day.

(c) High Brightness and skylight reflection on a sunny day.

Figure 2.7: Qualitative Analysis

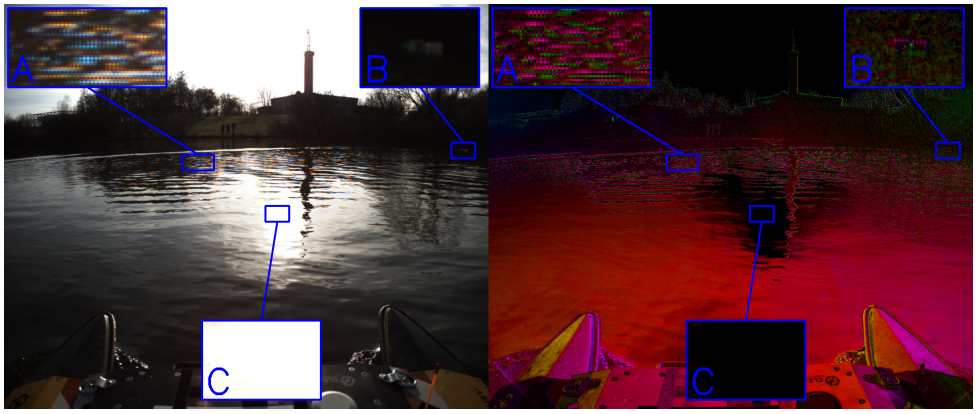


Figure 2.8: Main Challenges: A: Artifacts; B: Small Object; C: Saturation

2.5.4 Challenges

Three main challenges were identified in the PoTATO dataset and illustrated in Figure 2.8. First, the large number of images where the far-away bottles appear small in the image. The dataset contains a considerable amount of bounding boxes that have dimensions lower than 14^2 pixels, posing a challenge for object detection. Second, there is a correlated problem deriving from the fact that the superpixel shown in Figure 2.5a is quite large on the image plane, creating artifacts, especially on edges and image areas with high frequency, which is particularly common due to patterns created by ripples in the water surface. Both these challenges can be mitigated with different strategies for extracting the channels from the raw data [42] and can become the scope of further research. The third challenge was the saturation of the sensor under strong light reflection. In such cases, all polarization channels were measuring maximum intensity without any relative difference between them, making it impossible to estimate the degree and angle of polarization. Beyond using the auto-brightness function, this could be mitigated by using High

Dynamic Range (HDR) to capture two consecutive images with shorter and longer exposure to capture data on bright and dark regions. This approach was avoided to prevent temporal misalignment between image pairs.

2.6 Conclusion

In this chapter, we demonstrate the potential of polarimetric images to enhance vision system performance, particularly within the scope of autonomous water-cleaning robots operating in challenging outdoor environments. Our quantitative analysis shows that polarimetric images are capable of outperforming chromatic images for the object detection task in regions with heightened polarization properties. Through our initial experiments, we offer theoretical and practical insights into the advantages and challenges of using microgrid-polarization sensors for object detection in environments where light reflections on water surfaces are abundant. By providing the first dataset with raw polarimetric images and accompanying code, we aim to inspire further research and enable the development of novel approaches for fusing color and polarization modalities, with a strong potential for advancing the state-of-the-art perception algorithms. In Chapter 3, we shift from object detection to semantic segmentation and evaluate the performance of fusion algorithms on the PoTATO dataset.

3

Polarimetric and Chromatic Fusion for Segmentation

Chapter 2 established that polarization can improve detection in outdoor environments with strong polarimetric signals. This chapter advances the perception pipeline from detection to semantic segmentation, benchmarks state-of-the-art fusion networks on the PoTATO dataset, and then compares their performance with single-image baselines using traditional models. The results indicate that polarimetric cues help recover low-contrast objects and suppress reflection-induced false positives, raising mean IoU and lowering contour error relative to RGB inputs. These sharper masks come at a cost: the additional channels enlarge the models, increasing the computational load and introducing the risk of new false positives. The segmentation models explored here indicate potential for identifying the water surface and for shoreline and obstacle segmentation, which can support future mapping and safe navigation. Nonetheless, subsequent chapters utilize object detection instead of semantic segmentation for perception.

This chapter substantially reuses content from the following research article [43].

- **Batista, L.F.W.***, Bourbon, T.* and Pradalier, C. "Evaluation of Polarimetric Fusion for Semantic Segmentation in Aquatic Environments". In: 2025 IEEE International Conference on Visual Communications and Image Processing (VCIP). 2025. [3]

Compared with the original article, this chapter contains minor modifications to improve its integration with the thesis.

[3] *Statement of contributions:* Luis contributed to the literature review, research conception, dataset preparation and annotation, implementation of segmentation baselines, and manuscript writing, and supported the fusion evaluation. Tom, as an equal contribution author, focused on the evaluation of fusion techniques, providing a significant contribution to the execution of experiments, and editing the manuscript. Cedric provided guidance and feedback throughout the research process.

3.1 Introduction

Perception is fundamental to autonomous systems, enabling effective interpretation of and interaction with their environments. Semantic segmentation, in particular, plays a critical role by providing pixel-level classification, essential for precise localization and object identification in complex scenes [44]. However, achieving robust semantic segmentation in outdoor, water-rich environments remains a significant challenge due to water surface reflections and dynamic lighting conditions. These conditions obscure object boundaries and degrade the performance of standard vision algorithms.

Individual sensors often exhibit limitations under specific environmental conditions, and therefore, multimodal approaches have emerged to address these weaknesses by integrating data from different sensor types, thereby enhancing robustness and reliability [45]. Various strategies for combining different modalities involve trade-offs in terms of computational complexity, synchronization requirements, and the ability to capture cross-modal correlations.

As introduced in Chapter 2, there has been growing interest in polarimetric imaging. Microgrid-based polarimetric sensors capture polarization and color data synchronously, avoiding the alignment issues common to multi-sensor systems [18]. This makes them particularly suitable for outdoor applications, where abundant natural polarization sources, such as skylight scattering and water surface reflections, produce rich polarization patterns [20].

In this chapter, we investigate the potential of polarimetric imaging for semantic segmentation of floating objects in water-rich environments. Specifically, we focus on evaluating fusion algorithms that integrate chromatic and polarimetric information to enhance perception. Our main contributions are summarized as follows:

- Establish baseline for well-known semantic segmentation architectures using polarimetric-aware images.
- Conduct a comprehensive evaluation of state-of-the-art fusion algorithms for combining color and polarimetric modalities, specifically Angle of Linear Polarization ($AoLP$) and Degree of Linear Polarization ($DoLP$).
- Analyze the practical potential and limitations of applying these methods in autonomous systems, with particular attention to performance gains and computational overhead.

3.2 Related Work

According to [22], detecting floating objects is often hindered by changing outdoor lighting, sun glare, and reflections on the water surface. In a comprehensive survey of maritime computer vision datasets and deep learning techniques, [46], the authors emphasize that unpredictable reflections on the water surface remain a key bottleneck, arguing for further research to improve the stability of deep learning models in response to such fluctuating aquatic conditions.

To address these perception challenges, multi-sensor configurations have gained increasing attention in recent years. While the number of annotated datasets for semantic segmentation and object detection has expanded, smaller yet increasing collections of multi-modal sensor data have also begun to attract attention [46]. Notable efforts include pairing mmWave radar with RGB cameras for floating-bottle detection [5], and fusing RGB, grayscale, and thermal cameras with both LiDAR and radar for maritime perception [47]. Focusing on inland waterways, Cheng *et al.* [48] introduce a benchmark comprising multiple radars, a stereo camera, and a LiDAR.

However, the temporal and spatial misalignment typical of heterogeneous sensors increases fusion complexity and can degrade overall performance.

As an alternative to multi-sensor setups, polarization imaging has gained growing attention due to its ability to capture complementary information using a single, synchronous sensor. A recent survey highlights the rapid expansion of polarimetric imaging research within deep learning [49]. Polarization signals have also improved the segmentation of transparent objects [30], yet these experiments were performed on indoor images that do not capture challenging outdoor lighting, and their implementation is not publicly released.

In contrast, recent architectures offer more flexible multi-modal fusion frameworks and are often accompanied by publicly available code and datasets. MCubeS [50] provides outdoor RGB and polarimetric imagery but focuses on material recognition, with its water class limited to small urban puddles. CMX [51] augments an RGB backbone by concatenating a single additional modality channel. Extending this idea, CMNeXt [52] reports substantial gains by flexibly fusing RGB with arbitrary modalities, thus accommodating more than two inputs. MMSFormer [53] and StitchFusion [54] can also integrate an arbitrary number of modalities and achieve improved results when evaluated on the MCubeS dataset. Although encouraging, these studies investigate polarization fusion mainly for autonomous-driving or outdoor scenes in limited datasets that contain little or no water surfaces.

Despite the versatility of recent fusion architectures, polarization has rarely been explored in water-rich environments. Chapter 2, evaluating polarimetric object detection based on the PoTATO dataset, concluded that leveraging *DoLP* (D) and *AoLP* (A) information improves robustness to specular reflections. While the results indicate the value of polarimetric signals in aquatic scenes, the PoTATO benchmark currently evaluates each modality independently, leaving the potential benefits of fusion unexplored.

Our work addresses the lack of polarimetric fusion studies in aquatic environments. Using the publicly available PoTATO dataset, we establish single-modality segmentation baselines and benchmark state-of-the-art fusion algorithms.

3.3 Methodology

3.3.1 Dataset Preparation

We base our experiments on the PoTATO dataset introduced in Chapter 2, which contains raw polarimetric images of plastic bottles floating on water. This section details the derivation of polarimetric modalities from the raw data, the conversion of bounding-box annotations to pixel-accurate segmentation masks, and the construction of the final training, validation, and test partitions.

As introduced in Chapter 2, the raw images in the PoTATO dataset contain the intensity of light at four polarization orientations, I_0, I_{45}, I_{90}, and I_{135}, for each pixel of the RGGB Bayer pattern. These measurements allow the computation of the Stokes vector (Equation (2.2)).

From the Stokes components we derive three scalar quantities that leverage polarimetric information: the Degree of Linear Polarization, $\text{DoLP} = \sqrt{S_1^2 + S_2^2}/S_0$; the Angle of Linear Polarization, $\text{AoLP} = \frac{1}{2}\arctan(S_2/S_1)$; and an approximation of the diffuse-only intensity, $I_{\text{dif}} = (S_0 - \sqrt{S_1^2 + S_2^2})/2$. Following the extraction pipeline presented in Chapter 2, these signals are rendered as three images: RGB (standard color), DIF (reflection-suppressed using I_{dif}), and POL (pseudo-color composite with *AoLP* as hue and *DoLP* as value). Example images are shown in Figure 3.1



(a) *RGB.* (b) *DIF.* (c) *POL.*

Figure 3.1: Samples from the PoTATO dataset. Figure 3.1a Standard color image. (Figure 3.1b) Diffuse-only image with surface glare suppressed. (Figure 3.1c) Pseudo-color image encoding (*AoLP*) and (*DoLP) information*

Because the PoTATO dataset provides bounding-box annotations only, we produced pixel-level masks with the Segment Anything Model (SAM) [1], prompted with the original boxes. All masks were inspected and, where required, adjusted manually. Reflections on the water surface were excluded whenever distinguishable from the bottle.

Distant bottles appear as small objects, occupying only a few pixels in the image. To mitigate this scale imbalance, we cropped each image around the region of interest, removing irrelevant sky, vegetation, and boat hull regions. The resulting images have a resolution of 1224×512 pixels. We refer to this final version of the dataset as PoTATO-Seg, which comprises 1,500 images divided into 1,000 for training, 200 for validation, and 300 for testing.

3.3.2 Segmentation Architectures

Fusion models. We evaluate CMNeXt [51], MMSFormer [53] and StitchFusion [54]. Each network is trained with an RGB image, with different polarization modalities resulting in four input sets: RGB, RGBD, RGBA, and RGBAD. **Single-image baselines**. To benchmark fusion, we fine-tuned U-Net [55] and DeepLabV3 [56] on each image type (RGB, DIF, and POL). Supplying a single image to any fusion backbone simplifies the network to a SegFormer-style encoder-decoder [57]; therefore, a dedicated SegFormer baseline was not included. Together, these three models cover convolutional, atrous-convolutional, and transformer paradigms, providing a comprehensive reference for the fusion experiments.

3.3.3 Experimental Details

Training was carried out on two NVIDIA RTX 3090 GPUs with 24 GB of memory each. To preserve the physical consistency of the polarimetric information, we applied only horizontal flipping as data augmentation. Because many ground-truth masks are small, no additional down-sampling was performed; instead, each cropped region was resized to 512×512 before being passed to the network. Each model was optimised for 50 epochs with a batch size of 4. The learning rate was linearly warmed up during the first 10 epochs and then scaled by a factor of 0.01. Full configuration files and scripts are available in the public code repository.

3.4 Results

In this section, we present our quantitative and qualitative analysis. All metrics and examples are from the test set.

Single-image baseline metrics in Table 3.1 show that input modality influences segmentation quality for every architecture. DIF images yield the highest mIoU and F1 scores for all models except DeepLab, where the polarimetric input (POL) is slightly superior. Additionally, the modern fusion-style networks StitchFusion, MMSFormer, and CMNeXt surpass the classical U-Net and DeepLab. These findings confirm that suppressing water-surface glare through diffuse-only preprocessing provides consistent benefit, and polarimetric cues are a valuable complementary source of information for segmentation tasks. Our baseline metrics are aligned with the results presented in Chapter 2, and support the validity of the experimental setup.

Table 3.1: Single-image baselines on the PoTATO-Seg test set.

Model	Modality	mIoU(%)	Precision	Recall	F1
U-Net	RGB	73.034	0.841	0.839	0.833
	POL	71.518	0.887	0.770	0.812
	DIF	**75.786**	0.875	0.848	0.854
DeepLab	RGB	70.946	0.838	0.818	0.817
	POL	**71.752**	0.874	0.783	0.819
	DIF	70.922	0.851	0.796	0.814
CMNeXt	RGB	76.081	0.868	0.861	0.857
	POL	73.394	0.832	0.848	0.832
	DIF	**76.646**	0.884	0.850	0.859
MMSFormer	RGB	75.679	0.867	0.857	0.853
	POL	72.361	0.819	0.849	0.825
	DIF	**76.729**	0.882	0.853	0.860
StitchFusion	RGB	77.043	0.871	0.866	0.862
	POL	71.944	0.833	0.830	0.821
	DIF	**77.197**	0.875	0.865	0.863

The metrics from fusion models shown in Table 3.2 indicate that adding *DoLP* and *AoLP* channels to RGB images generally enhances segmentation, although the magnitude of the gain depends on the fusion approach. StitchFusion benefits most: each auxiliary channel increases mIoU, and the RGBDA variant achieves the highest overall scores. MMSFormer sees moderate improvements that plateau when both cues are present, whereas CMNeXt shows only minor, non-significant gains. These findings suggest that architectures incorporating explicit cross-modal fusion modules exploit multi-channel inputs more effectively. Moreover, polarimetric cues provide complementary information, although the benefits of their integration vary across models.

To quantify boundary accuracy across modalities, we computed the Average Surface Distance (ASD_{over}) defined as the mean of the shortest distances from each point on the prediction contour to the ground-truth contour. Smaller values denote better alignment. Figure 3.2 shows these values after discarding statistical outliers located more than 1.5 times the interquartile range (the distance between the first and third quartiles) beyond the first or third quartile to eliminate a few rare false detections that would otherwise distort the distributions. The plots reveal a consistent trend across all three models: adding polarimetric channels lowers the ASD_{over} metric, demonstrating reduced average contour error compared with RGB alone.

Table 3.2: Fusion models including polarimetric cues.

Model	Modality	mIoU(%)	Precision	Recall	F1
StitchFusion	RGB	77.04	0.871	0.866	0.862
	RGBD	78.71	0.894	0.862	0.874
	RGBA	78.86	0.900	0.860	0.875
	RGBDA	**79.03**	0.902	0.862	0.876
MMSFormer	RGB	75.68	0.867	0.857	0.853
	RGBD	77.11	0.857	0.876	0.862
	RGBA	**78.20**	0.896	0.856	0.871
	RGBDA	77.42	0.883	0.855	0.864
CMNeXt	RGB	76.08	0.868	0.861	0.857
	RGBD	76.47	0.870	0.860	0.859
	RGBA	76.43	0.888	0.843	0.857
	RGBDA	**76.74**	0.883	0.849	0.860

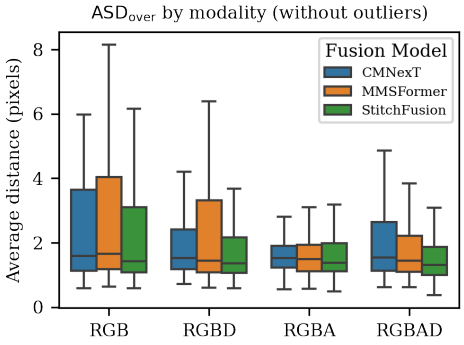


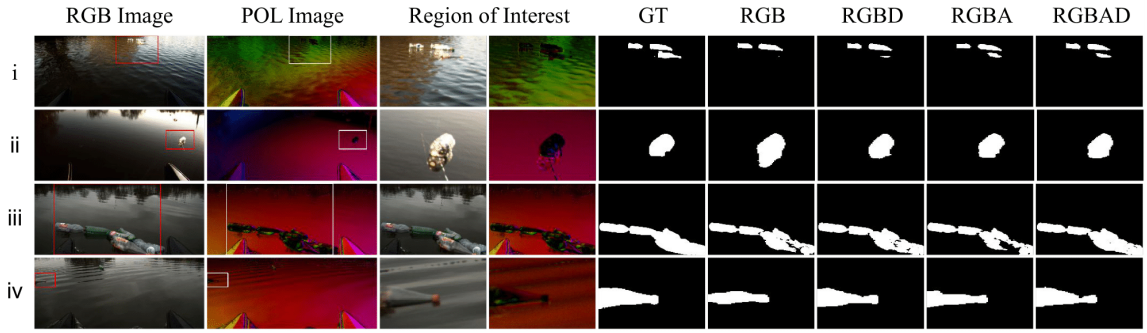
Figure 3.2: Adding polarimetric channels lowers boundary error.

The mean inference time across the test set is presented in Table 3.3. Adding extra modalities consistently slowed inference. While DeepLab is marginally faster than U-Net, both are significantly faster when compared to any fusion model.

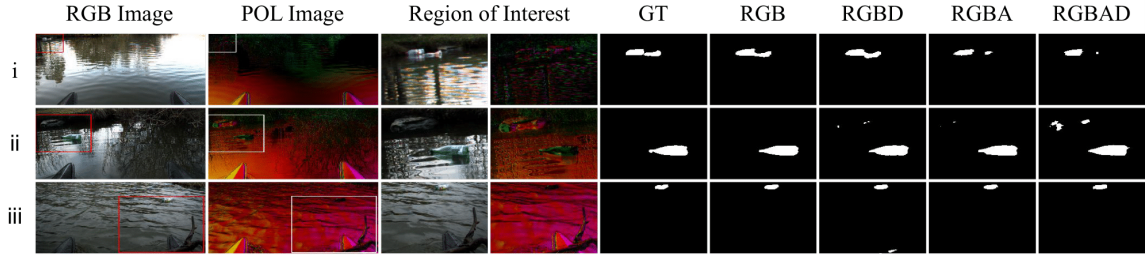
Model	RGB	RGBD	RGBA	RGBAD
U-Net	5.2 ms	-	-	-
DeepLab	**5.0 ms**	-	-	-
CMNeXt	24.3 ms	37.7 ms	37.8 ms	39.6 ms
MMSFormer	**18.2 ms**	**24.2 ms**	**24.3 ms**	**24.5 ms**
StitchFusion	22.1 ms	45.4 ms	45.7 ms	83.9 ms

Table 3.3: Mean Inference Times (ms) for the PoTATO-Seg.

For qualitative assessment, we examine StitchFusion, the best-performing model. Figure 3.3 gathers scenes with the largest IoU gaps between modalities, complementing the quantitative results. Figure 3.3a shows examples when polarization enables StitchFusion to detect low-contrast objects and suppress reflection-induced false positives. Figure 3.3b highlights failure cases in which the added channels introduce false positives or miss true positives in regions with weak polarization signals.



(a) Examples when polarization enables detection of a bottle under low-contrast lighting (i), suppresses a strong reflection mistaken for an object in RGB (ii), and recovers a bottle whose color matches the water surface (iii, iv), all producing sharper masks for the RGBAD input.



(b) Polarimetric fusion limitations. (i) Failure to improve the detection of a distant bottle located in a region with weak polarization signals. (ii) Shoreline rocks and (iii) tree branches produce polarimetric signatures similar to floating bottles, resulting in false positives.

Figure 3.3: Qualitative Examples of Polarimetric Fusion using StitchFusion.

3.5 Discussion

Baseline experiments demonstrate consistent results with what is presented in the previous chapter, and fusion models gain most in regions with strong polarimetric information. *DoLP* and *AoLP* information can help to suppress glare and reveal color-concealed targets, with the largest improvements observed for StitchFusion, whose cross-modal blocks reduce contour error and increase IoU metrics.

In some cases, extra channels can introduce false positives when background materials produce similar polarimetric responses and provide little benefit where polarization is weak. This highlights the opportunity for better fusion strategies. The lack of ImageNet-style pre-training forces the network to learn polarimetric features from scratch, slowing convergence. Additionally, the available polarimetric datasets are rather small. In our experiments, validation curves start to indicate overfitting after about 50 epochs, suggesting that the long schedules common in the fusion literature could be excessive for the dataset used in this evaluation.

Despite these training challenges, the contour analysis presented in Figure 3.2 indicates that polarimetric fusion yields high-precision masks. On the other hand, the extra channels enlarge the input tensor, increasing model size and latency. Inference time experiments confirm that these costs must be balanced against the accuracy gains, and this should be carefully considered, especially for applications with real-time requirements and limited computing power.

3.6 Conclusion

In this chapter, we evaluated how polarimetric cues can enhance the semantic segmentation of floating objects. We provide a systematic and reproducible evaluation baseline that includes two standard segmentation models and three modern general-purpose fusion models. Our experiments indicate that integrating *DoLP* and *AoLP* information raised the mean IoU and lowered the contour error by suppressing reflection and increasing contrast. Nonetheless, the fusion models sometimes produced false positives when background materials had similar polarimetric signatures and contributed little in areas with weak polarization. We conclude that, in general, polarimetric information is most useful for applications that demand precise masks and have strong potential in scenes with rich polarimetric information, but they incur additional training complexity and increased inference time. As the downstream objective is to generate goal positions for floating-waste collection, the remainder of this thesis focuses on object detection as the primary perception task. Future work should explore larger and more diverse polarimetric datasets, tailored training strategies, and lightweight fusion modules optimized for real-time deployment.

Part II

Reinforcement Learning-based Navigation

4

Simulation and Reinforcement Learning Framework

This chapter provides preliminaries for the reinforcement learning framework used throughout this dissertation. We motivate the adoption of NVIDIA's Isaac ecosystem for maritime robotics simulation, explain how it meets requirements for scalable GPU-parallel training, and situate this choice relative to existing maritime simulators. This chapter also documents the framework's progression across releases. For clarity, we map the framework versions to their use in Chapters 5 to 7. By consolidating these implementation details, we provide preliminary context for the reinforcement learning methods introduced in Chapter 5 and further developed in Chapters 6 and 7. Finally, this chapter also references the evolution of the implemented framework in collaboration with other groups, highlighting the framework's generality and ongoing extensions that support broader mobile robotics research.

This chapter was written for this thesis and cites the following two articles [2, 58], to which the author contributed.

- El-Hariry, M.*, Richard A.*, Castan, R.M.*, **Batista, L.F.W.***, Geist, M., Pradalier C., Olivares-Mendez M. "RoboRAN: A Unified Robotics Framework for Reinforcement Learning-Based Autonomous Navigation". Transactions on Machine Learning Research, 2025.[4]
- Chung, L.K., Jung, W., Pullabhotla, S., Shinde, P., Sunil, Y., Kota, S., **Batista, L.F.W.**, Pradalier, C. and Kousik, S. "Guaranteed Reach Avoid for Black-Box Systems through Narrow Gaps via Neural Network Reachability". In: 2025 IEEE International Conference on Robotics and Automation (ICRA). 2025.[5]

[4] *Statement of contributions:* Luis contributed to the hydrodynamics, buoyancy, and propeller implementation for the reinforcement learning framework, integration and execution of tests in simulation and deployment to Kingfisher ASV and field tests.

[5] *Statement of contributions:* Luis contributed to supporting the use of the RL framework as well as the execution of simulation and field tests for the ASV platform.

4.1 Introduction

Although several simulation environments for marine robotics exist, options can be limited in scope and maturity. An actively developed framework based on Gazebo is described in [59], which builds upon earlier efforts to simulate underwater dynamics, including buoyancy and hydrodynamic forces [60]. This environment provides a robust foundation for marine robotics research and offers a practical entry point for most simulation needs.

Stonefish [61] marks a substantial step forward, providing real-time simulation well suited to control and rapid prototyping. However, the absence of GPU acceleration is a major constraint, particularly when scaling to the computational demands of deep reinforcement learning (DRL) training pipelines.

In contrast, simulation frameworks that integrate well with RL workflows are more commonly found in other domains. For example, the framework proposed in DRIFT [62] expands on NVIDIA's Isaac ecosystem and effectively addresses similar navigation tasks in the context of satellite systems. Nonetheless, such frameworks typically lack the specialized modeling of hydrodynamics and buoyancy that are critical for marine robotics.

Although simulation tools exist for maritime applications and highly parallelizable RL frameworks are available in other domains, a simulation environment that combines both highly parallel simulation and maritime-specific modeling remains absent. This gap motivates the development of the simulation and reinforcement learning framework used throughout this thesis.

There is strong industry-supported research and development on RL using NVIDIA's Isaac-family frameworks, motivated by the need for standardized interfaces, high-throughput GPU simulation, and tight integration with modern RL libraries. As a result, these frameworks are widely adopted across manipulation, locomotion, and navigation tasks, where reproducibility and sim-to-real transfer are central concerns. As a fast-evolving platform, the Isaac stack has undergone rapid iteration, transitioning from Isaac Gym to Isaac Lab, deepening integration with Isaac Sim/Omniverse, and improving support for distributed and multi-GPU training.

For this thesis, the pace of change has had mixed effects. On the positive side, it lowered the barrier to implementing and comparing new algorithms, provided robust baselines, and reduced boilerplate around data pipelines, logging, and experiment management. On the other side, it necessitated frequent upgrades, refactoring of extensions to track API changes, and additional software-engineering effort to maintain compatibility and reproducibility.

In this chapter, we outline the evolution of the Isaac-based RL framework used in this thesis, specify the versions adopted in each experimental chapter, and highlight external studies that have built upon the extensions introduced in our work.

4.2 Reinforcement Learning Framework

We chose the Isaac Sim–based simulation environment for this work due to its ability to run many environments in parallel with GPU-accelerated physics, which is essential for accelerating reinforcement learning (RL) training. Developed and maintained by NVIDIA, Isaac Sim and its RL stacks have seen rapid evolution. The frameworks progressed from Isaac Gym (Preview) [63] to OmniIsaacGymEnvs (OIGE) [64] and finally to Isaac Lab [65]. Each stage is paired tightly with specific Isaac Sim versions, reflecting concurrent refactors in the simulator (e.g., the extension namespace changes landing in 4.5) and codified migration guidance across releases. This coupling has directly shaped the design and maintenance of our extensions.

- **Isaac Gym:** The initial GPU-accelerated platform placed both physics and learning en-

tirely on the GPU, enabling orders-of-magnitude throughput gains for continuous-control tasks and establishing the case for highly parallel RL in robotics.

- **OmniIsaacGymEnvs:** As an evolution of Isaac Gym, OmniIsaacGymEnvs integrated NVIDIA Omniverse capabilities to offer more realistic and extensible environments for robotics research. This version provided native compatibility with the RL framework rl-games [66].
- **Isaac Lab:** The current framework unified OIGE and Orbit [65], providing a modular RL stack that supports multiple RL libraries, including SKRL [67], RSL-RL [68], and rl-games [66]. It represents the most flexible and extensible version to date, facilitating a wide range of RL experiments and applications.

This evolution occurred in parallel with the development of this thesis. Table 4.1 summarizes key releases of the RL stack, their recommended Isaac Sim pairings, and the chapters in which each is used.

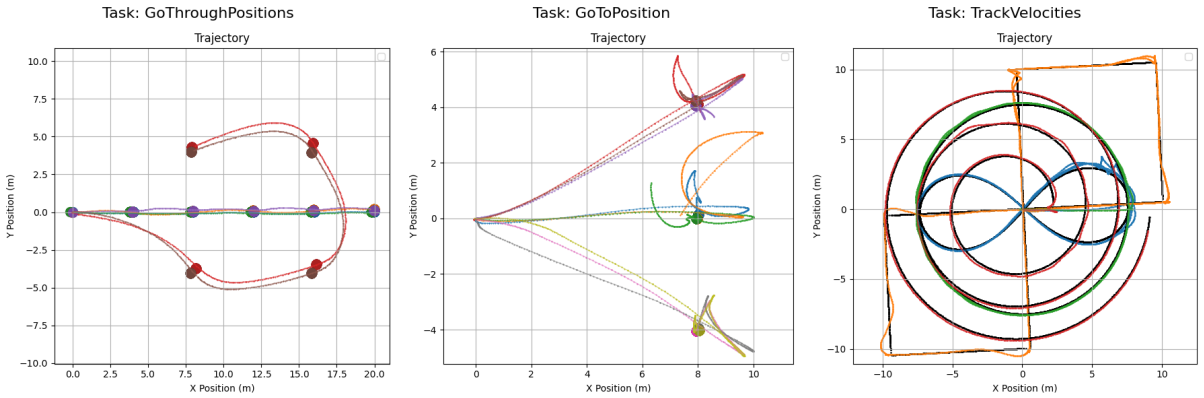
RL Framework (version)	**Release date**	**Isaac Sim**	**Used in**
OIGE 2022.2.1	Mar 2023	2022.2.1	Chapter 5
OIGE 2023.1.0	Oct 2023	2023.1.0	—
OIGE 2023.1.1	Dec 2023	2023.1.1	Chapter 6
OIGE 4.0.0	Jun 2024	4.0.0	Internal tests
Isaac Lab v1.0.0	Jun 2024	4.0	—
Isaac Lab v1.1.0	Jul 2024	4.1	—
Isaac Lab v1.2.0	Sep 2024	4.2	—
Isaac Lab v1.3.0	Nov 2024	4.2	Internal tests
Isaac Lab v2.0.0	Jan 2025	4.5	—
Isaac Lab v2.1.0	Apr 2025	4.5	Chapter 7
Isaac Lab v2.2.0	Aug 2025	5.0	—

Table 4.1: RL framework evolution (OIGE stands for OmniIsaacGymEnvs), release timing, recommended Isaac Sim pairing, and corresponding chapter where it was used.

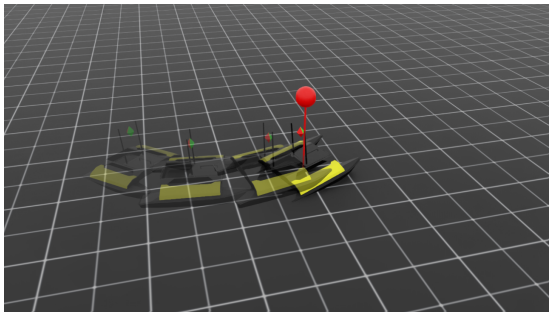
To enable the development of RL policies for ASVs, we extended the RL framework from NVIDIA, adding buoyancy, hydrodynamics, and thruster dynamics models, with a vectorized implementation for batched execution across many parallel environments. These components were iteratively ported and validated across the OIGE to the Isaac Lab transition, while adopting upstream API refactors. Although tracking the evolving platform required periodic refactoring, it provided timely bug fixes, better sensor and physics fidelity, and improved training stability, which we consider critical for our use and which also increases its potential for adoption by the broader research community.

4.3 Framework evolution beyond this thesis

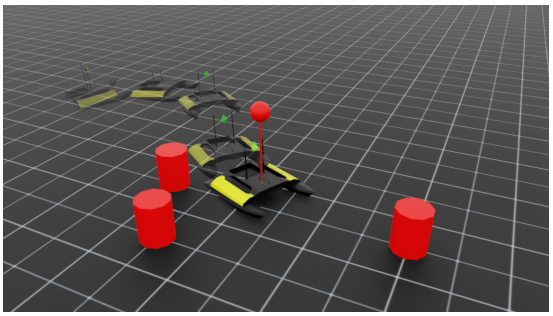
In addition to the work presented in this dissertation, and in collaboration with the SpaceR Lab, the framework evolved into an extensible benchmark and deployment stack called RoboRAN: A Unified Robotics Benchmark for Reinforcement Learning-Based Autonomous Navigation [58]. RoboRAN targets cross-domain, robot–task–decoupled navigation research with an emphasis



(a) Trajectories of repeated field experiments executing three different tasks: going through a sequence of positions, going to a position, and tracking velocities to reach a predefined trajectory.



(b) *GoToPosition* task in simulation.



(c) *GoToPosition* task with obstacles.

Figure 4.1: Example of additional tasks implemented in RoboRAN and evaluated for the Kingfisher ASV in both simulation and field tests.

on reproducibility, standardized evaluation, and sim-to-real transfer across land, water, and microgravity settings.

RoboRAN extends Isaac Lab with (i) a modular design that cleanly separates robots and tasks via standardized APIs; (ii) uniform tasks and metrics for cross-medium evaluation; (iii) sim-to-real validation on multiple platforms; and (iv) the first open-source deployment API that executes Isaac Lab–trained policies on real robots, featuring ROS 2 integration and containerized workflows. The framework supports heterogeneous embodiments, including a TurtleBot 2, a Floating Platform, and the Kingfisher ASV (among others). It implements a common interface with consistent observation structures, reward layers, and a domain-randomization stack to improve transfer to real-world deployment.

For the ASV, RoboRAN incorporates hydrostatics, hydrodynamics, and thruster actuation to capture drag-dominated, inertial behaviors in water, enabling scalable training and evaluation of control policies under realistic disturbances. Within the benchmark's task suite, the ASV was evaluated in simulation and in field trials for a variety of tasks with examples shown in Figure 4.1. The benchmark reports standardized success metrics and continuous-control indicators (e.g., distance and heading errors, time-to-target, velocity-tracking errors), and documents sim-to-real trends for the ASV alongside the other robots. Detailed experiment design, figures, and per-task results are provided in the RoboRAN manuscript [58] and are not included in the scope of this thesis.

4.4 Research Collaboration

The reinforcement learning (RL) framework developed in this thesis has been useful in advancing research across multiple areas, particularly in safety-critical applications. Notably, it supported the validation of NeuralPARC [2]—a method that addresses the reach–avoid problem for black-box systems with complex dynamics. Using the framework introduced here and further detailed in Chapter 5, an ASV was trained in disturbance-rich settings, and thousands of trajectories were generated for data-driven modeling. Our learning framework provided a valuable opportunity to validate NeuralPARC's reachability guarantees under realistic, high-dimensional conditions, effectively bridging data-driven learning and formal safety verification for ASV applications as well.

4.5 Conclusion

This chapter introduced the reinforcement learning framework that underpins the remainder of this thesis. It also provided a broader perspective on how the framework is being used beyond the scope of this dissertation.

The framework evolved together with the thesis. As requirements matured, we improved implementation, expanded functionality, refined the reward design, and strengthened the simulator-to-robot deployment process to support reproducible experiments and transfer to the Kingfisher platform. This iterative process is reflected in the next three chapters, each applying the framework at a distinct stage: Chapter 5 uses a first version, detailing the methodology, defining the core task, and validating it with field deployments. Chapter 6 employs a second version, extending the stability and robustness evaluation and identifying failure cases. Chapter 7 adopts a third version, following a large codebase update. It addresses the previously identified failure cases and focuses on integrating the full pipeline, combining perception, and deepening the analysis of sim-to-real transfer and robustness during deployment on the physical platform. Together, these chapters apply the framework progressively, introducing incremental improvements towards a more robust policy.

5

Deep Reinforcement Learning Framework for ASV Navigation

This chapter introduces the reinforcement learning methodology used to navigate an Autonomous Surface Vehicle to "capture" floating targets under uncertainty. Building on the Isaac-based simulation framework presented in Chapter 4, we formalize the task, observation, and action spaces, and training procedures. We first integrate buoyancy and hydrodynamics models into a modern reinforcement learning framework to reduce training time. Next, we show how system identification coupled with domain randomization improves the RL agent performance and narrows the sim-to-real gap. Real-world experiments for the task of capturing floating waste show that our approach lowers energy consumption while simultaneously reducing task completion time. This chapter provides the learning backbone that is further extended in Chapter 6 for robustness evaluation and integrated perception for full-system evaluation in Chapter 7.

This chapter substantially reuses content from the following research article [69].

- **Batista, L.F.W.***, Ro, J.*, Richard, A., Schroepfer, P., Hutchinson, S. and Pradalier, C. "A Deep Reinforcement Learning Framework and Methodology for Reducing the Sim-to-Real Gap in ASV Navigation". In: 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2024, pp. 1258–1264.[6]

Compared with the original article, this chapter contains minor modifications to improve its integration with the thesis.

[6] *Statement of contributions:* Luis contributed to project definition, experimental design, simulation tests, code development, robot deployment, and manuscript writing. Jungwhan, as an equal contribution author, focused on implementing and optimizing the reinforcement learning framework and contributed significantly to coding, field experiments, and manuscript writing. Antoine provided critical guidance on the reinforcement learning framework, feedback on the design of the reinforcement learning task and editing parts of the manuscript. Pete contributed to manuscript editing and provided feedback. Seth and Cedric offered guidance and feedback throughout the research process.

5.1 Introduction

As introduced in Chapter 2, plastic pollution poses a significant threat to global sustainability, impacting marine life, ecosystems, and human health [13]. To restore ecological balance, it is necessary not only to halt pollution but also to reverse its effects. As a solution, Autonomous Surface Vehicles (ASVs) are being increasingly recognized for their critical role in addressing this issue for both cleaning and monitoring water quality [70, 71]. The task of cleaning water surfaces autonomously has gathered attention not only from the academic sphere, [71, 72] but also from the industrial sector in companies such as IADYS, Clearbot, WasteShark, and OrcaUboat. While ASVs offer potential solutions for floating waste management, the autonomy of these solutions is often compromised by the complexity of the marine environments.

The advancement of ASVs depends on the development of robust and efficient control systems capable of handling highly dynamic and often unpredictable marine environments. As these ASVs often need to explore large areas without access to charging points, it is also critical that control algorithms operate efficiently to reduce energy consumption. This reduction in battery energy consumption allows for extended operational range, reduced costs, and lower environmental impact.

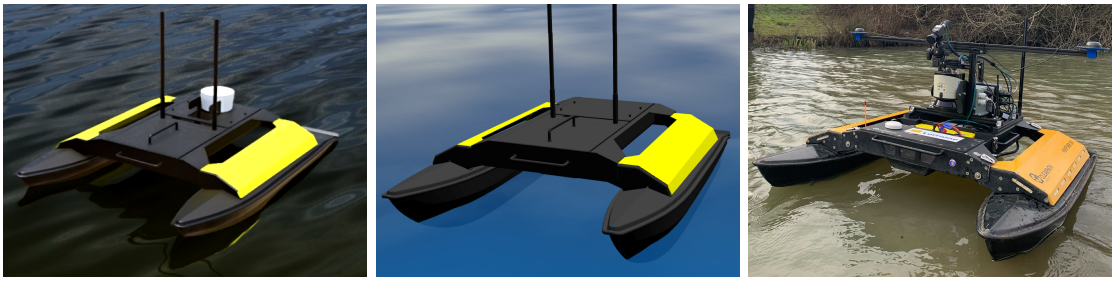
At the same time, Deep Reinforcement Learning (DRL) has emerged as a powerful tool in robotics, providing flexible and robust control strategies, and it has growing applications in the field of marine robotics and ASVs [6]. Recent work has even shown that the DRL-based approach is capable of outperforming traditional control algorithms in real-world trials [11]. Yet, field applications of DRL for ASVs are scarce, with most studies not extending beyond simulations, especially in the context of energy-efficient agents.

Despite its potential, the adoption of DRL for ASVs is hindered by two significant barriers. Firstly, there is a lack of high-performance marine simulation environments. Current options, such as Gazebo's UUV plugin [59] incorporate foundational buoyancy and hydrodynamics simulations but do not meet the extensive parallelization needs of DRL frameworks. Secondly, there is difficulty in validating and interpreting the internal decision-making processes of RL agents due to their 'black-box' nature. This opaque nature creates uncertainty with respect to RL agent reliability and safety in mission-critical applications, which can hinder the adoption and acceptance of such systems [73–75].

With these challenges in mind, we aim to narrow the gap between simulation and the real world through methodical system identification coupled with domain randomization. Reducing this gap improves the efficiency of DRL agents for sustainable applications of ASVs, addressing issues such as thruster dynamics, battery charge effects, and external disturbances, thus reducing behavior uncertainty as they are trained under more realistic conditions for real-world deployment. This chapter's main contributions include:

- Open-source, highly parallelized hydrodynamics and buoyancy implementation suitable for reinforcement learning framework[7].
- Methodology to reduce the sim-to-real gap by combining System Identification with Domain Randomization for Deep Reinforcement Learning.
- Real-world experimental evidence illustrating that our approach can minimize battery energy consumption and simultaneously speed up task completion.

[7] `https://github.com/luisfelipewb/RL4WasteCapture/`



(a) Isaac Sim (b) Gazebo (c) Real ASV

Figure 5.1: Simulated and Real ASV Platforms Used Through our Experiments

5.2 Related Work

There is a growing adoption of Deep Learning (DL) on ASVs in the areas of perception, control, and multi-robot collaboration [6]. Despite this trend, the utilization of DL-based controllers for ASV is still in the early stages. One of the main barriers to adoption lies in the challenge of developing agents capable of handling the complex hydrodynamic water-body interaction.

To address the complex dynamics, DRL-based controllers have been explored as a potential solution. In [76], a DRL-based controller was used for path following and demonstrated that in simulation, a DRL-based controller was capable of controlling ASVs in extremely complex systems. Similarly, [77] proposes an IQN-based local path planner to compensate for unknown currents and avoid obstacles. RL has also been proposed for ASV control problems in several other studies [7–9]. However, evaluation in these studies is often performed only on simulation and lacks a field test validation, which would demonstrate the ability to move from a training environment to the real world.

The studies that include the evaluation of DRL agents in real-world scenarios are slowly growing, but more are needed in different subdomains of DRL agent task performance. Wang et al. claim that DRL outperforms NMPC in trajectory tracking, showing lower tracking error and better disturbance rejection in river environments [11]. An RL-based strategy to integrate guidance and heading control, improving tracking accuracy and robustness of the controller is proposed in [78]. While this existing literature shows promising results, most of the research is still in the early stages and often focuses on the path-tracking task. As such, further real-world testing and validation are necessary to continue to fill the research gaps regarding how DRL controllers perform outside of the simulated training environment, in particular, on how changes in the training environment or algorithms can improve real-world operation performances.

While simulation environments exist, the options are rather limited. The most up-to-date option that is under active development is detailed in [59]. They extend previous initiatives to simulate buoyancy and hydrodynamics [60] and provide a solid starting point. Cieślak introduced Stonefish [61], an advanced simulation tool capable of real-time performance. However, it lacks GPU acceleration, limiting its scalability for large-scale simulations required for training DRL agents. Such frameworks suitable for the RL pipeline are available in other domains. For instance, [62] solves similar navigation tasks for satellites, but it misses the hydrodynamics and buoyancy simulation.

In summary, while the existing literature provides a solid foundation, it could be further enhanced by the inclusion of accessible open-source implementations, comprehensive analysis, and robust field test validation. Our work aims to enrich the field by offering accessible code and evaluating the impact of system identification to minimize the sim-to-real gap, including field

test validation.

5.3 Problem Formulation

5.3.1 Capture task

Capturing floating waste requires navigating the ASV over the waste, where a fixed net between its hulls traps it during passage. The target position is provided in a two-dimensional space, and the agent must be capable of navigating toward it by controlling the actions of each thruster. The goal is considered achieved when the ASV approaches the target head-on and passes over it with a tolerance of up to 0.3 m distance between the target's position and the center of mass (COM) of the vessel. We considered an environment free of obstacles to isolate the evaluation of the capture task. Anticipating future integration of perception capabilities of the onboard camera, the task has been designed to reflect a limited field of view of 90 degrees and capping the operational range at 10 meters.

5.3.2 Dynamic Model

The dynamic model of the ASV in the simulation is based on Fossen's six degrees of freedom model [3].

$$\mathbf{M}\dot{\nu} + \mathbf{D}(\nu)\nu + g(\eta) = \tau_{\text{thruster}} + \tau_{\text{disturbance}} \tag{5.1}$$

Where $\eta = [x, y, z, \phi, \theta, \psi]^T$ is the position vector, and $\nu = [u, v, w, p, q, r]^T$ is the velocity vector, each comprised of the following terms: surge, sway, heave, roll, pitch, yaw. $\mathbf{M}$ is the system inertia matrix.

$$\mathbf{D}(\nu) = \mathbf{D}_l + \mathbf{D}_q(\nu) \tag{5.2}$$

$\mathbf{D}$ denotes the hydrodynamic damping matrix, consisting of linear term $\mathbf{D}_l$ and quadratic term $\mathbf{D}_q(\nu)$. g is the vector of gravitational and buoyancy forces and moments. τ_{thruster} refers to the forces and moments caused by the ASV's propulsion system, and $\tau_{\text{disturbance}}$ to the forces and moments caused by external forces such as waves and wind. The Coriolis force's impact is considered negligible within our ASV's operational speed range ($< 2\ m/s$).

5.4 Methodology

Our proposed methodology shown in Figure 5.2, is composed of four main steps: System Identification, DRL training, Simulation Evaluation, and Real-world Evaluation. In the following subsection, each one is explained in more detail.

5.4.1 System Identification

In the system identification process, we utilized a streamlined dynamics model premised on the ASV's movement within a two-dimensional plane. This approach is substantiated by the typical operational environment of the Kingfisher ASV, which is relatively calm waters where the effects of roll and pitch on movement are minimal and thus can be disregarded for 2D motion considerations. The hydrodynamic damping matrix $\mathbf{D}$ has been tailored as follows.

$$\mathbf{D}_l = \begin{bmatrix} X_u & 0 & 0 \\ 0 & Y_v & 0 \\ 0 & 0 & N_r \end{bmatrix} \tag{5.3}$$

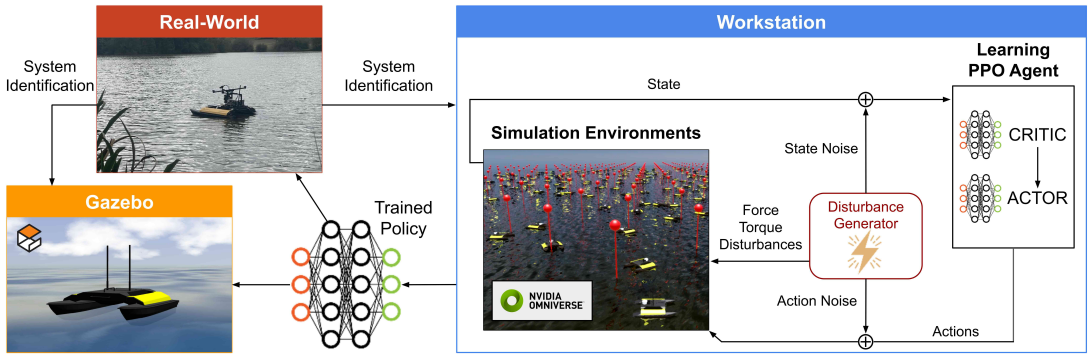


Figure 5.2: Integrated pipeline for DRL-Based ASV Control consisting of four main steps. 1. Real-world experiments for system identification; 2. Train policy in simulation; 3. Tests using an independent simulation environment; 4. Real-world tests.

$$\mathbf{D}_q(\nu) = \begin{bmatrix} X_{u|u|}|u| & 0 & 0 \\ 0 & Y_{v|v|}|v| & 0 \\ 0 & 0 & N_{r|r|}|r| \end{bmatrix} \tag{5.4}$$

where X_u, Y_u, and N_r are the hydrodynamic damping coefficients in surge, sway, and yaw, respectively, capturing the linear relationship. The terms $X_{u|u|}$, $Y_{v|v|}$, and $N_{r|r|}$ represent quadratic damping components that scale with the square of the velocities. For the ASV, system identification tests were conducted to derive the hydrodynamic coefficients as described in [79].

Figure 5.3 (a) shows the surge direction velocity during the acceleration test with various thruster commands, which were implemented to identify surge drag coefficients. It accelerates the ASV from rest and observes its maximum velocity after reaching a steady state. In the steady state, the force generated by the thrust in the surge direction balances out with the damping force, establishing equilibrium. Figure 5.3 (b) shows the yaw rate during the rotation test, which was implemented to identify yaw drag coefficients. In this test, balanced forces were applied to both thrusters so that the ASV could rotate without surge and sway at a single point. The test results were then employed to perform quadratic curve fitting, from which we derived the drag coefficients. For the sway coefficient, circle and zigzag tests were performed. These tests consistently showed the sway velocity remaining below 0.1 m/s, indicating a minor influence under our test conditions. Consequently, the sway damping coefficient was manually adjusted to a high value, ensuring the model aligns with observed behaviors in the real world. The final results are shown in Table 5.1.

Table 5.1: Hydrodynamic Parameters

Parameters	**Nominal**	**Identified**
X_u	16.45	0.00
Y_v	15.80	99.99
N_r	6.00	0.83
$X_{u\|u\|}$	2.94	17.26
$Y_{v\|v\|}$	2.76	99.99
$N_{r\|r\|}$	5.00	17.34

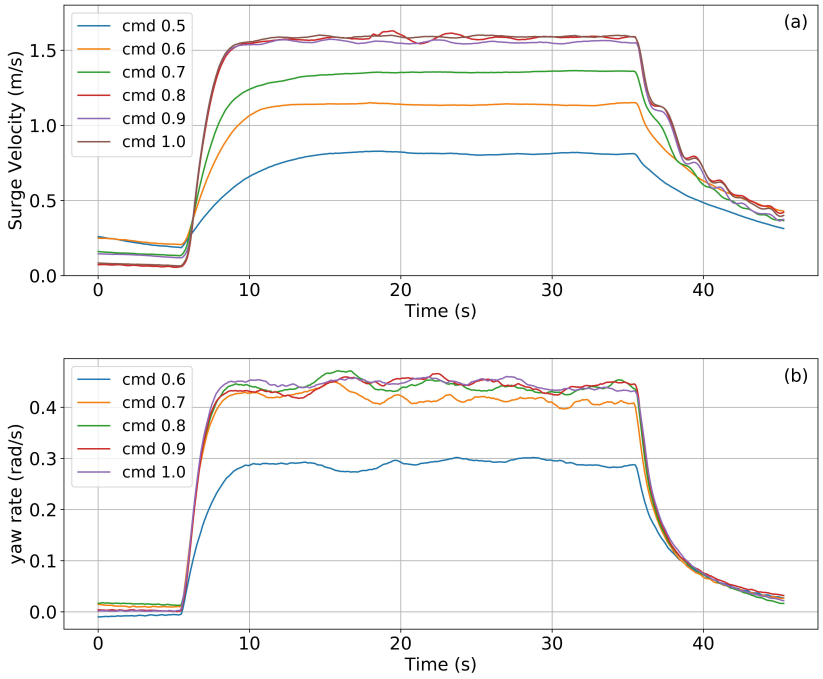


Figure 5.3: System Identification Experiments. (a) shows the surge velocity of the ASV during the acceleration test, and (b) shows the yaw rate of the ASV during the rotation test.

Figure 5.4 illustrates the thruster forces measured through a bollard pull test. The red line represents the nominal thruster force for the Heron ASV as provided by Clearpath Robotics. In contrast, the blue and orange lines represent the actual measured forces for the left and right thrusters, respectively, corresponding to the applied commands. It is noteworthy that the graph reveals a minor disparity between the performance of the two thrusters, which may pose challenges in learning precise control for the ASV due to its asymmetry.

5.4.2 RL Framework for ASV

Expanding upon the established methodologies from [80] and [81, 82], our work centers on developing an RL framework tailored for large-scale ASVs simulation. In this enhanced training environment, we incorporate buoyancy and hydrodynamics into Isaac Sim to simulate realistic marine conditions. In particular, we performed a GPU-accelerated port of the UUV gazebo plugin [60] to Isaac Sim. This port allows, for instance, the ability to compute the hydrodynamic forces of thousands of systems seamlessly on a GPU. Similar to [80], this simulator is built on top of Nvidia's Isaac Sim and OmniIsaacGym [63], a simulator capable of running thousands of simulation environments in parallel. This framework marks a novel contribution to ASV development, facilitating the training and evaluation of RL agents. Moreover, thanks to its increased training speed and broader exploration capabilities, the proposed framework offers a significant leap in training capabilities for ASV systems. It allows model-free methods such as PPO [83] to compete against MPC or Optimal Controllers. Finally, by providing an open-source, scalable, and robust platform, this work could serve as a key step towards advanced autonomous marine operations, potentially impacting environmental sustainability efforts.

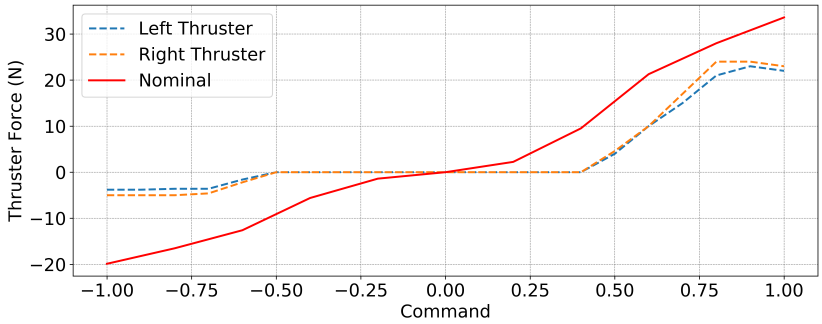


Figure 5.4: Comparison of Nominal and Measured Thruster Forces

5.4.3 Training process

Observation space: The observation space, denoted as $\mathbf{o}_t \in \mathbb{R}^6$ for each time step t, comprises a vector of kinematic and positional features.

$$\mathbf{o}_t = [u_t \quad v_t \quad r_t \quad \cos(\delta_t^{\text{head}}) \quad \sin(\delta_t^{\text{head}}) \quad d_t]^T \tag{5.5}$$

The components are the surge speed u_t, sway speed v_t, and yaw rate r_t, which represent the ASV's linear and angular velocities in the body frame. The ASV's orientation towards the goal was represented by $cos(\delta_t^{\text{head}})$ and δ_t^{head}, ensuring a continuous representation of the target bearing. The last element d_t represents the distance to the goal at time step t (Figure 5.5). This approach of focusing only on the local frame enhances the model's generalizability.

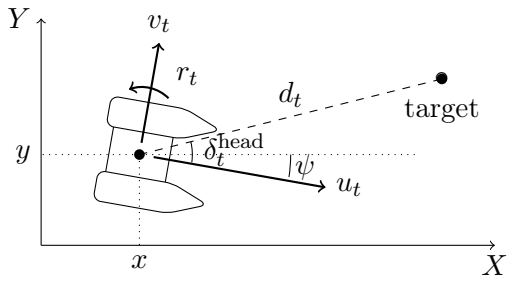


Figure 5.5: Diagram of the ASV's observation space components

Action space: The action space, represented as $u_t = [u_t^{\text{left}} \quad u_t^{\text{right}}]^T$, comprises the command inputs for the left and right thrusters, respectively, at time step t. These commands are converted into thrust forces using the characteristic force curves obtained from system identification experiments. The resultant forces are then applied to the corresponding thrusters in the simulation.

Reward function: To effectively train the ASV agent for the precision capture task, a compound reward function r_t is used to encourage desired behaviors.

$$r_t = r_t^{\text{dist}} + r_t^{\text{head}} + r_t^{\text{energy}} + r_t^{\text{alpha}} + r_t^{\text{time}} + r_t^{\text{goal}} \tag{5.6}$$

Each reward term is detailed as follows:

$$r_t^{\text{dist}} = \lambda_1 \cdot (d_{t-1} - d_t) \tag{5.7}$$

r_t^{dist} encourages the reduction of the distance to the goal and reflects the ASV's progress towards its goal.

$$r_t^{\text{head}} = \lambda_2 \cdot (e^{k_1(\delta_t^{\text{head}})^4} + e^{k_2(\delta_t^{\text{head}})^2}) \tag{5.8}$$

r_t^{head} promotes alignment of the ASV's heading with the goal orientation, with continuous reward, with a pronounced peak near zero.

$$r_t^{\text{energy}} = \lambda_3 \cdot (e^{k_3 E} - 1) \tag{5.9}$$

r_t^{energy} term penalizes excessive use of actuation to encourage energy efficiency, where $E = (u_t^{\text{left}})^2 + (u_t^{\text{right}})^2$.

$$r_t^{\text{alpha}} = \lambda_4 \cdot (e^{k_4|r_{t-1} - r_t|} - 1) \tag{5.10}$$

r_t^{alpha} discourages abrupt changes in angular velocity, thereby facilitating stable and gradual turns.

$$r_t^{\text{time}} = \lambda_5 \tag{5.11}$$

r_t^{time} applies a penalty for each time step, encouraging the agent to reach the goal quickly.

$$r_t^{\text{goal}} = \begin{cases} \lambda_6, & \text{if } d_t < d_{threshold} \\ 0, & \text{otherwise} \end{cases} \tag{5.12}$$

Finally, the r_t^{goal} component grants a one-time reward when the ASV successfully reaches the goal. $d_{\text{threshold}}$ determines the required proximity for successful goal achievement. $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6$ are weights for each reward component, and k_1, k_2, k_3 and k_4 are constants that adjust the sensitivity of the reward components to their respective errors.

Training Details:

Table 5.2: The reward function's parameter values

Weights	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6
Values	1.0	0.02	0.01	1.0	-0.2	30.0
Parameters	k_1	k_2	k_3	k_4	$d_{\text{threshold}}$	
Values	-10.0	-0.1	1.0	-0.33	0.1 m	

The values and parameters of the reward function, shown in Table 5.2, have been carefully calibrated to optimize the agent's behavior. The cumulative reward per time step remains negative prior to the goal being reached, thereby refining the ASV's approach toward the target. A conservative lower value for $d_{\text{threshold}}$ was chosen, which is less than the actual required proximity to capture the target.

Table 5.3 presents the core learning configurations employed in our training. The training process was completed with an Nvidia GeForce GTX 3090 GPU with 24GB RAM for approximately 10 minutes.

To robustly train the DRL agent, we incorporated a comprehensive randomization strategy across the simulation environment. This strategy involved introducing observation noise to replicate the accuracy of sensors employed in the ASV (position: ± 3 *cm*, orientation: ± 0.025 *rad*)

Table 5.3: Reinforcement learning configurations

Parameters	Values
actor neural network	128 × 128
critic neural network	128 × 128
learning rate	0.0001
gamma	0.99
number of environments	1024
sample batch size	16384
max iterations	1000
max steps per episode	3000
time step size	0.02

and perturbing action commands to reflect control imprecision. ±10% random variation in the drag coefficients was introduced to account for uncertainties in its hydrodynamic model, thereby diversifying the training environments and preventing overfitting to idealized hydrodynamic parameters.

We modeled external force and torque disturbances to simulate natural environmental conditions, applying a force field with a random offset of $\pm 2.5\ N$ and a maximum sine force amplitude of 2.5 N, along with torque disturbances having a random offset of $\pm 1.0\ N \cdot m$ and a sine torque amplitude of $1.0\ N \cdot m$. These disturbances, representing about 10% of the ASV's thrust capacity, are significant enough to test control strategy robustness without overwhelming the system, enhancing agents' reliability and performance in actual operational scenarios.

Additionally, we applied up to 50% randomization to thruster forces to account for fluctuations due to battery level variations and discrepancies between the nominal and actual measured thruster forces, ensuring the training process encompasses a realistic spectrum of operational conditions.

5.4.4 Simulation Environment

The initial evaluation of the trained models was conducted in the Gazebo simulation environment, using the UUV [60] plugin. This choice was motivated by the chance to validate the models' performance independence from the Isaac Sim training environment and to facilitate ROS integration for field tests. Gazebo's parameters, calibrated with system identification parameters identified in Section 5.4.1, provided an environment for assessing model behavior.

5.4.5 Autonomous Surface Vehicle Platform Integration

For the field test experiments, the Kingfisher ASV, developed by Clearpath Robotics, was used. It features a catamaran design, with dimensions of 1.35 m in length and 0.98 m in width, with a weight of 35.96 Kg. Its propulsion system is based on a thruster mounted on each hull, as shown in Figure 5.1c. The motors are driven by a proprietary low-level controller board that controls each thruster using an input command of at least 10 Hz between $[-1.0, 1.0]$. As a higher-level processing unit, we used an NVIDIA Jetson Xavier with CUDA integrated into the ASV to provide energy-efficient high-performance computing capable of providing real-time inference for the RL agent. Power was provided using a 4-cell LiPo battery pack with 22 Ah capacity. Despite being equipped with two cameras and a laser, these sensors were not used. For experimental purposes, perception was abstracted, and the goal was provided programmatically. The localization relied on a high precision SBG Ellipse-D integrated IMU and RTK GPS with a

dual-antenna, multi-ban GNSS receiver providing high position, velocity, and heading precision of 0.02 m, 0.03 m/s, and 0.5° respectively.

5.5 Experiments

To evaluate how the integration of system identification with domain randomization affects the model's performance, experiments were conducted in simulation and real-world environments with agents trained using the nominal values (NV) or identified values (SID) as described in Table 5.1. Thruster force domain randomization (DR) was applied in some cases, resulting in four different tests (Table 5.4).

Table 5.4: Agents training configurations for experiments

Name	Thruster and Hydrodynamic parameters	Thruster Force Randomization
NV	Nominal	0%
NV-DR	Nominal	50%
SID	Identified	0%
SID-DR	Identified	50%

5.5.1 Simulation Evaluation

Target locations were generated with distances set between 3 to 9 meters at 1-meter intervals and angles from -45 to 45 degrees in 5-degree steps. This setup enabled a focused evaluation of navigational efficiency necessary for capturing waste. The Simulation results, detailed in Figure Figure 5.6, show that NV struggled to reach the target in the first approach, having to turn around and correct the trajectory, while other agents presented better results.

5.5.2 Real-world Evaluation

To validate simulations and assess model applicability, field tests were conducted on a day with mild wind speeds (reported maximum of 6 km/h) per local weather. Focusing on controlled yet realistic conditions, these tests covered a subset of predefined simulation goals due to battery constraints. Results are shown in Figure 5.7 where SID-DR provides smoother trajectories while reaching the target faster.

5.5.3 Evaluation Metrics

Each model's performance was evaluated based on two main metrics: completion time (T_c) and accumulated energy consumption E_{acc}. In general, thruster power consumption is positively correlated to its thrust. Hence, the accumulated power consumption E_{acc} of the controller can be represented by Equation (5.13), where N is the number of data points and a_l^i and a_r^i are the action commands on the left and right thrusters.

$$E_{\text{acc}} \propto \sum_{i=1}^{N} (|a_l^i| + |a_r^i|) \tag{5.13}$$

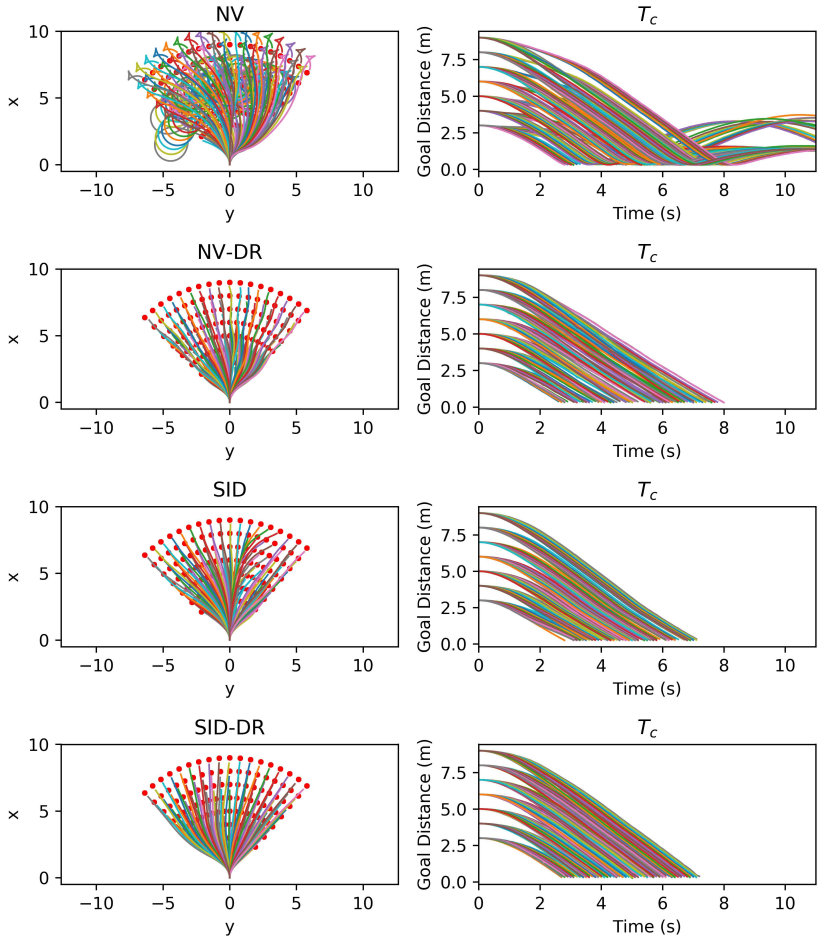


Figure 5.6: Simulation tests showing trajectories (left) and goal distance over time (right). NV exhibits difficulty in reaching the target (red dot), while SID-DR completes the task faster.

5.5.4 Results

Our field experiments demonstrated a 13.1% reduction in energy consumption and a 7.4% improvement in task completion efficiency when comparing the performances of NV-DR and SID-DR approaches. These numbers are extracted from the average across the different initial distances presented on Table 5.5 and underscore the efficacy of incorporating SID in conjunction with DR to optimize autonomous operations.

Real-world conditions, characterized by dynamic elements such as waves and wind, naturally require more energy and time to complete the task than the simulated environments test, where such disturbances were absent. This can also be observed on Table 5.5.

The data points gathered from field tests are comparatively limited. Therefore, the evaluation of energy consumption was also conducted in simulation to obtain metrics of greater statistical significance. The results shown in Figure 5.8 corroborate the trends observed in real-world tests, specifically when comparing NV-DR and SID-DR across various initial distances.

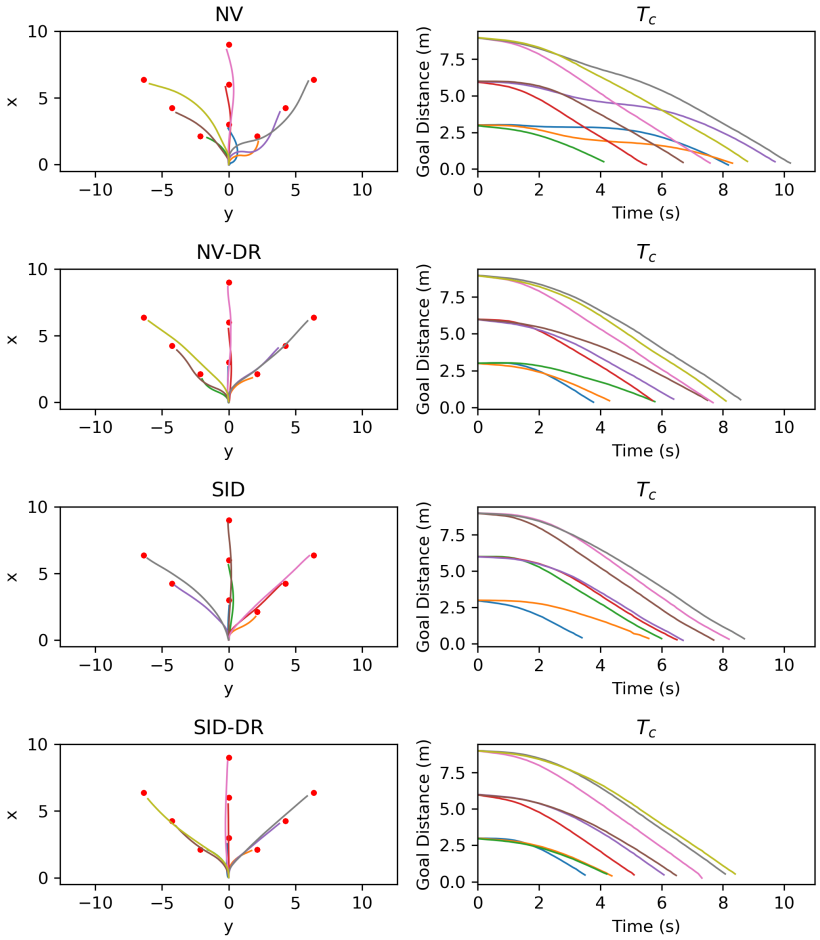


Figure 5.7: Field tests demonstrate that all models are capable of reaching the target while SID-DR completes the task faster with smoother trajectories.

5.6 Discussion

The field tests of DRL agents for autonomous waste collection vessels revealed critical challenges rarely observed in simulation. The initial issue was the significant discrepancy between simulated and real-world performance during preliminary trials, which highlighted the need for improved system identification and domain randomization techniques presented in this work. Primarily, adjusting for thruster dynamics was crucial to bridge the reality gap and enhance agent adaptability. In addition, the nonlinear relationship between battery charge levels and motor power output significantly affected preliminary results. External disturbances and sensor noise further compounded these challenges, necessitating a minimal domain randomization level to train agents capable of generalizing to real-world deployment. Finally, the intricacies of reward shaping proved to be a substantial hurdle. Developing a reward function that effectively guides the agent towards efficient waste collection while avoiding local optima requires careful calibration and thoughtful consideration.

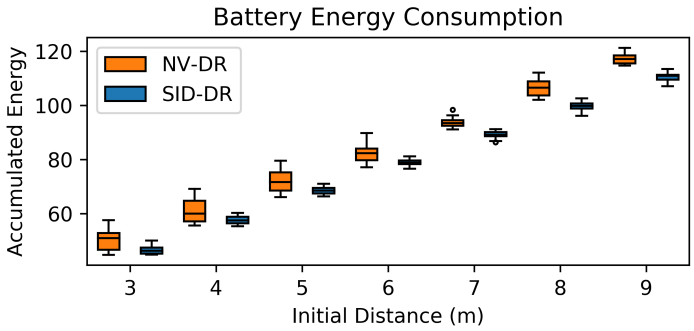


Figure 5.8: Large-scale Simulation Analysis Shows that SID-DR Outperforms NV-DR in Battery Efficiency.

Table 5.5: Time and Energy metrics

		$T_c(s)$			E_{acc}		
	Goal distance	$3m$	$6m$	$9m$	$3m$	$6m$	$9m$
Sim	NV	4.0	15.4	16.9	63.2	236.6	269.8
	NV-DR	3.2	5.3	7.3	50.2	82.3	117.5
	SID	3.3	5.0	6.9	44.7	81.0	116.1
	SID-DR	3.0	4.9	6.7	46.6	78.9	110.5
Real	NV	6.9	7.3	8.9	105.8	114.9	139.1
	NV-DR	4.6	6.5	8.1	77.8	105.4	131.9
	SID	4.5	6.4	8.2	74.9	88.6	123.3
	SID-DR	4.0	5.9	7.9	57.8	90.8	125.2

5.7 Conclusion

In this chapter, we present a highly parallelizable RL framework with the integration of buoyancy and hydrodynamic models. Furthermore, we share a methodology that combines system identification with domain randomization, effectively narrowing the sim-to-real gap in ASV simulations. Our approach is validated through both simulation and real-world trials, specifically focusing on the task of capturing floating waste. Results demonstrate not only a reduction in task completion time but also a decrease in energy consumption. In subsequent chapters, we extend the learning framework (see Chapter 4 for version details) and evaluate the capture task for robustness.

6

Robust ASV Navigation

This chapter builds on the framework and methodology presented in Chapter 5 with a focus on evaluating the robustness of trained policies under parametric uncertainty and external disturbances. We present a stress-testing protocol that combines large-scale simulation with controlled field trials, including variations in mass and drag, and reports performance metrics. The experiments are based on an upgraded framework with refinements to the methodology and training parameters introduced in Chapter 5. This chapter also presents a model predictive control baseline (MPC) and subjects it to the same robustness evaluation. A direct performance comparison between the MPC and the trained policy is inappropriate due to differences in objective functions and reward structures. Instead, we report, in parallel, how external disturbances degrade each approach. The robustness tests delineate safe operating envelopes and expose key limitations, which are addressed later in the integrated system in Chapter 7.

This chapter substantially reuses content from the following research article [84].

- **Batista, L.F.W.**, Aravecchia, S., Hutchinson, S. and Pradalier, C. "Evaluating Robustness of Deep Reinforcement Learning for Autonomous Surface Vehicle Control in Field Tests". In: 2025 Workshop on Field Robotics (ICRA). 2025 [8]

Significant modifications were made to adapt the text to the context of this dissertation. To avoid extensive redundancy, this chapter reports only the differences in problem formulation, RL-specific methodology, and conclusions, referring the reader to the previous chapter.

[8] *Statement of contributions:* Luis contributed to RL policy implementation, simulation tests, field tests, and manuscript writing. Stéphanie contributed to MPC implementation, manuscript writing, and field tests. Seth and Cédric offered guidance and feedback throughout the research process.

6.1 Introduction

Autonomous Surface Vehicles (ASVs) have increasingly gained recognition for their potential in water quality monitoring and waste removal [70, 71]. Autonomous cleaning tasks using ASVs have attracted attention from both academic [72] and industry [5] research. Despite their potential, the effectiveness of ASVs for floating waste management can be limited by several complexities.



(a) Jellyfishbot from IADYS (b) Research Platform

Figure 6.1: Examples of ASVs equipped with nets for capturing floating waste. The changing payload due to waste accumulation affects the ASV's hydrodynamics and maneuverability.

A particular challenge in capturing floating waste is the dynamic variation of the payload. As illustrated in Figure 6.1a, one approach is to equip the ASV with a net to trap debris as it navigates the water. As waste accumulates in the net, the changing payload affects the ASV's mass, moment of inertia, and drag characteristics, which in turn significantly alter its dynamics. Additionally, the ASV is also exposed to external disturbances such as wind and water currents. These changes necessitate real-time adjustments to the control strategy to maintain accurate and stable operation during the waste collection process.

Traditional control techniques, such as Adaptive Control and Model Predictive Control (MPC), are widely used for ASV navigation. While effective in many applications, these methods face significant challenges, particularly in accurately estimating system dynamics, tuning parameters, and accounting for external disturbances. MPC, in particular, relies on precise models to optimize a cost function over a receding horizon, but precisely modeling the system dynamics for ASVs remains complex. This modeling is typically done through system identification (ARMA [85]), or the model is learned with a neural network, as shown on an ASV by [86]. The main drawback of such methods is that they rely on a model, and therefore are not designed to cope with a system whose dynamic model may change. Recent advances have introduced adaptive MPC techniques that account for payload variations, as demonstrated in the Roboat III project [10]. However, the described approach requires manual input for payload data, with automatic payload estimation planned for future work. To improve robustness, novel methods based on MPC estimate online either external perturbations, such as [87], or unknown parameters of the model, such as [88], but they typically focus on a single variable. In contrast, our work addresses multiple dynamic changes simultaneously, including mass, inertia, drag, and environmental forces like wind and currents, to enhance the robustness of ASV control systems.

Simultaneously, Deep Reinforcement Learning (DRL) has recently emerged as a powerful tool in robotics, offering flexible and robust control strategies with growing applications in marine robotics and ASVs [6]. For instance, [76] used a DRL-based controller for path following, demonstrating its effectiveness in managing ASVs within highly complex systems in simulation.

Similarly, [77] introduced an IQN-based local path planner capable of handling unknown currents and avoiding obstacles. Various other studies have also explored RL techniques for ASV control problems [7–9], though many of these focus predominantly on simulations, with limited real-world testing. In contrast, some research has emphasized evaluating DRL agents in real-world environments. For example, [82] trained a model-based RL agent for ASV trajectory following, demonstrating its robustness in simulation and its adaptability to real-world conditions in a zero-shot setup. Additionally, an RL-based approach combining guidance and heading control to improve tracking accuracy and controller robustness was proposed by [78]. Wang et al. further showed that DRL outperformed Nonlinear-MPC in trajectory tracking, achieving lower tracking error and better disturbance rejection in river environments [11].

Although these studies show promising results, they mainly focus on path-following tasks and are predominantly evaluated in simulation. While some agents are tested in real-world trials, their performance and robustness under varying conditions are not comprehensively analyzed.

In this chapter, we aim to explore and validate the robustness of an RL agent designed for capturing floating waste under different disturbance conditions. The main contributions are:

- Evaluation of the robustness of a DRL-based controller for ASVs through simulations and field tests, demonstrating its resilience against variations in center of mass, rotational drag, and external disturbances.
- Field experiments validating the DRL agent's performance in realistic scenarios, showing minimal degradation compared to traditional controllers such as MPC.
- Practical insights into the strengths and limitations of DRL-based control strategies alongside the code used to train the RL models, serving as a baseline for future ASV control research and development.

6.2 Methodology

6.2.1 Problem Formulation

This chapter adopts the problem formulation introduced in Chapter 5. We use the same ASV dynamic model employed in simulation, namely Fossen's six-degree-of-freedom model [3], with implementation details provided in Section 5.3.2. As before, the reinforcement learning task is restricted to planar motion (two dimensions; see Figure 5.5), and we evaluate the capture task introduced in Section 5.3.1. Unless stated otherwise, the notation and assumptions are identical to those in Chapter 5.

6.2.2 Platform

To train the RL-based control policy, we used the highly parallelized framework detailed in Section 5.4.2. As described in Chapter 4, we upgraded Isaac Sim and OmniIsaacGymEnvs (OIGE), refactored the codebase, and introduced curriculum learning to improve training stability.

We also used a separate simulation environment based on Gazebo (Section 5.4.4) because it provides independent physics and hydrodynamics implementations, enabling cross-validation of the RL agent's performance and helping diagnose overfitting to the Isaac Sim physics stack. Additionally, Gazebo's ROS integration facilitates a seamless transition to the real-world platform for testing.

In both simulation environments, the linear and quadratic hydrodynamic damping coefficients in Equation (5.2) were set to the values obtained from the system identification experiments in Section 5.4.1. For completeness, the coefficients are: $X_u = 0.00$, $Y_v = 99.99$, $N_r = 5.83$, $X_{u|u|} = 17.26$, $Y_{v|v|} = 99.99$, $N_{r|r|} = 17.34$.

For the field tests, we used the same Kingfisher ASV from Clearpath Robotics; details including dimensions, weight, and control interface are provided in Section 5.4.5.

6.2.3 Control approach

RL

The RL-based control approach follows the details introduced in Section 5.4.3. The observation and action spaces are unchanged. A minor simplification was implemented in the heading reward (Equation (5.8)) by replacing it with Equation (6.1) with no measurable impact on training performance. The remaining rewards remain unchanged.

$$r_t^{\text{head}} = \lambda_2 \cdot \left(e^{k_1, \delta_t^{\text{head}}} \right) \tag{6.1}$$

We also introduced curriculum learning. Task difficulty increased gradually over the first 300 epochs and then remained fixed for the final 200 epochs to enable policy optimization. The final policy weights are used in all experiments. This reduced the total number of training epochs by half relative to Chapter 5 (from 1000 to 500). Other relevant training hyperparameters are as specified in Table 5.3.

MPC

To provide a baseline comparison, we implement an MPC controller for the ASV.

The dynamic model of the ASV is a 2D simplification of Equation (5.1), as proposed in [89], where only the linear term from the hydrodynamic damping matrix (Equation (5.2)) is kept. This model is given by :

$$\mathbf{M}\dot{\nu} + \mathbf{D}_l \nu = \tau_{\text{thruster}}, \quad \text{where} \quad \nu = [u \quad v \quad r]^T \tag{6.2}$$

The kinematic equation relating velocities in the body and global frames is given by $\dot{\eta} = \mathbf{R}(\psi)\nu$, where $\mathbf{R}(\psi)$ is the rotation matrix from body frame to global frame and $\eta = [x, y, \psi]^T$ is the pose vector in the global frame (Figure 5.5).

Finally, the applied forces are described by

$$\tau_{\text{thruster}} = \mathbf{B}\mathbf{u} = \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ a/2 & a/2 \end{bmatrix} \begin{bmatrix} \mathbf{u}^{\text{left}} \\ \mathbf{u}^{\text{right}} \end{bmatrix} \tag{6.3}$$

All the other terms are defined following the convention presented in Chapter 5.

From these equations, we can reformulate the dynamic model of the ASV:

$$\dot{\mathbf{q}}(t) = f(\mathbf{q}(t), \mathbf{u}(t)) \tag{6.4}$$

Where $\mathbf{q} = [x, y, \psi, u, v, r]^T$ is the state and $\mathbf{u}$ is the command (Equation (6.3)). The hydrodynamic parameters from $\mathbf{D}_l$ are identified with the system identification method described in Section 6.2.1. The mass m is measured, and the inertia I_z is estimated from the weight and shape of the robot, allowing for estimating $\mathbf{M} = diag(m, m, I_z)$.

Finally, we formulate the optimal control problem (OCP) for the MPC as the least squares function that minimizes the deviations of state and control from the reference over the given time horizon. In this problem, the reference is simply the (x, y) coordinates of the target. The OCP is solved with the Acado [90] framework. Following [91], we execute one iteration of the optimization at each control loop, using as the initial state the last available state, and the previous solution as initialization to the next optimization. That allows the optimization to run in real-time, at the ASV control frequency (20Hz), with a 3-second time horizon.

6.2.4 Evaluation Methodology

The evaluation methodology focuses on comparing the performance degradation of MPC and RL-based controllers under varying disturbances.

The RL agent was trained in Isaac Sim, using domain randomization to simulate disturbances. However, both the RL and MPC were evaluated under the same disturbance conditions in the Gazebo simulation environment, ensuring consistency in the physics simulation.

Field tests were conducted in real-world settings with natural disturbances, such as wind, as well as controlled disturbances. In the controlled tests, two additional disturbance scenarios were introduced: (1) a 4 kg weight was attached to the right of the center of mass, and (2) a floating net containing partially filled water bottles, creating a drag force with an offset 20 cm from the center. These disturbances combined created asymmetric perturbances to the vessel and are shown in Figure 6.2



Figure 6.2: Disturbance applied during field tests: additional mass attached to the right of the center of mass and a floating net with water bottles creating asymmetric drag.

To assess performance, we measured three main metrics: the duration to reach the goal normalized by the target distance, T_{norm}, the accumulated energy normalized by the target distance, $E_{\text{acc}}^{\text{norm}}$, and the increase in trajectory length compared to a straight line Δ_d. Given d, the distance to the goal when it is first received, T, the time to reach it, d_{real} the length of the actual trajectory, $d_{threshold}$ the distance threshold below which the target is reached, and N the total number of control actions until the goal is reached, the metrics are:

$$T_{norm} = T/d \tag{6.5}$$

$$\Delta_d = d_{real} - (d - d_{threshold}) \tag{6.6}$$

$$E_{\text{acc}}^{\text{norm}} = (\sum_{i=1}^{N}(|\mathbf{u}_t^{\text{left}}| + |\mathbf{u}_t^{\text{right}}|))/d \tag{6.7}$$

Robustness was evaluated by comparing each controller's performance in undisturbed and disturbed conditions, with MPC being compared against disturbed MPC, and RL against disturbed RL. This approach ensures a fair comparison between each controller's robustness to disturbances, as their objective functions and optimization processes differ.

6.3 Experiments

6.3.1 Simulation

The simulation in Gazebo was used to assess the agent's robustness to variations in key parameters, including surge drag (X_u), rotational drag (N_r), mass (M), center of mass (CoM), and moment of inertia about the Z-axis (I_z). Among these, we selected for detailed analysis the two with the most significant impact on the agent's performance during preliminary investigation, CoM and N_r. The CoM shifts from 0.0 to 12.5 cm along the Y-axis, and N_r from 0 to 20.

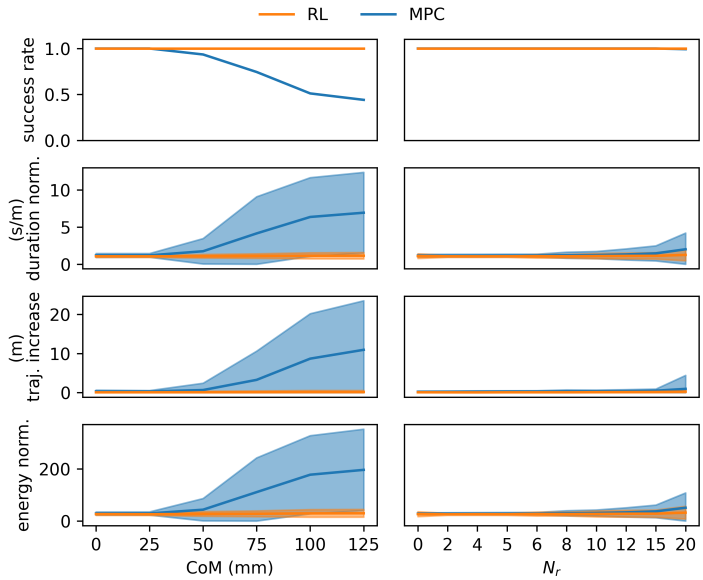


Figure 6.3: Impact of changes in center of mass and rotational drag on agent performance in simulation. Mean and standard deviation of success rate, normalized duration, T_{norm}, and trajectory increase, Δ_d, and normalized accumulated energy, $E_{\text{acc}}^{\text{norm}}$, are shown.

No additional disturbances (e.g., wind or currents) were applied in the simulation to isolate and evaluate the effects of the parameter variations. To conduct our evaluation, we compare the performances of the agents with a set of goal and initial speed conditions, when we vary either CoM or N_r. For a given value of CoM and N_r, the goal distances were set at 1-meter intervals between 3 and 9 meters, with bearing angles ranging from -45 to 45 degrees in 5-degree increments. Also, we tested three different initial speeds (0, 0.5, and 1 m/s), resulting in a total of 399 goals per environment.

The mean and standard deviation for each experiment are shown in Figure 6.3 for the three metrics, normalized duration T_{norm}, trajectory increase Δ_d, and normalized accumulated energy, $E_{\text{acc}}^{\text{norm}}$. The figure also shows the success rate. It is possible to observe that the RL agent maintained a consistent success rate near 100% across all variations in the center of mass (CoM) and rotational drag (N_r), whereas the MPC controller experienced a significant drop in performance as CoM shifted.

As the CoM shifted, the MPC's success rate declined sharply, and all three metrics grew substantially, indicating its sensitivity to these changes. In contrast, the RL agent exhibited robustness across all conditions showing minimal changes in normalized duration and accumulated energy, and experienced negligible trajectory increase, even under large CoM shifts and variations in N_r.

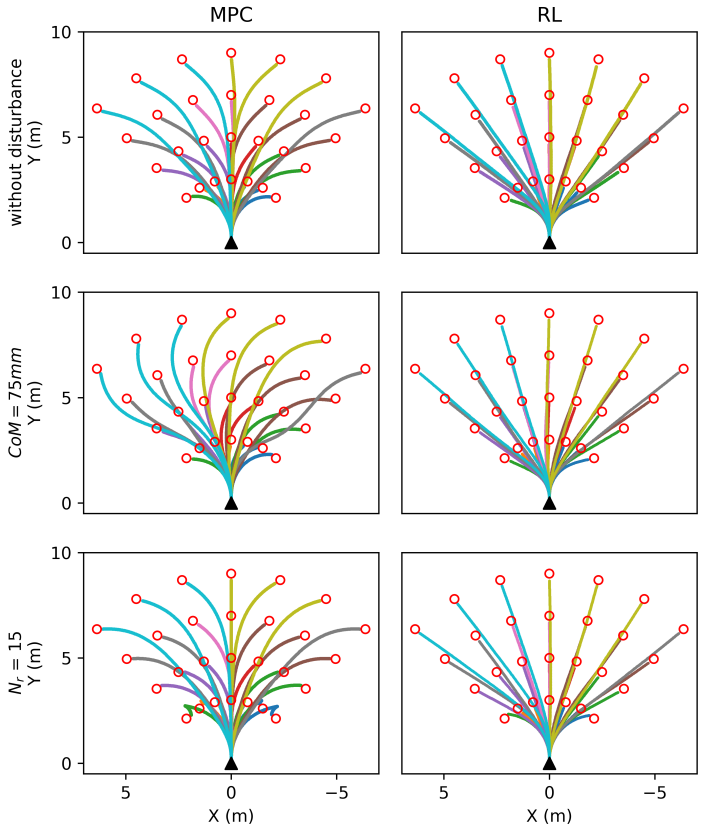


Figure 6.4: Trajectories for a subset of goals (red circles) in simulation under varying disturbances. The RL agent maintains consistent trajectories, while the MPC shows varied deviation patterns as disturbances grow.

The impact of the disturbances on the trajectories is shown in Figure 6.4, which displays three scenarios: normal conditions, $CoM = 75$ mm, and $N_r = 20$. The RL agent consistently maintains a straight trajectory despite varying disturbance conditions. Notably, the CoM disturbance causes a distorted trajectory, while the high N_r disturbance forces the MPC agent to overshoot and backtrack, particularly at the 3 m and 45-degree goals.

6.3.2 Field tests

We present the results of field tests conducted under two conditions: (i) a baseline scenario, referred to as *normal*, where the boat operated solely under natural weather disturbances, and (ii) a perturbed scenario, referred to as *disturbed*, in which additional mass and drag were applied, as depicted in Figure 6.2. In the disturbed test, the 4 kg mass was attached with an offset of approximately 38 cm to the right side, changing the ASV's center of mass, and a floating net with water bottles partially filled with water (weighing 2.8 kg outside of water) was attached behind the ASV to simulate additional drag. The attachment point, as indicated in Figure 6.2, was offset by approximately 21 cm to the right. These offsets combined, produced an asymmetric disturbance, causing a counterclockwise torque to the Z axis of the vessel as it accelerates forward.

We quantified the weather disturbances, mostly wind, by recording RTK IMU measurements over 8 minutes without input commands. The ASV reached angular speeds of -0.14 to 0.07 rad/s, surge speeds of -0.15 to 0.34 m/s, and sway speeds of -0.13 to 0.05 m/s. These disturbances are significant compared to the ASV's maximum velocity of 1.6 m/s, highlighting the impact of environmental forces on performance.

The field test trajectories in Figure 6.5 show the RL agent maintaining a straight path, while MPC reacts slowly to disturbances, similar to simulation.

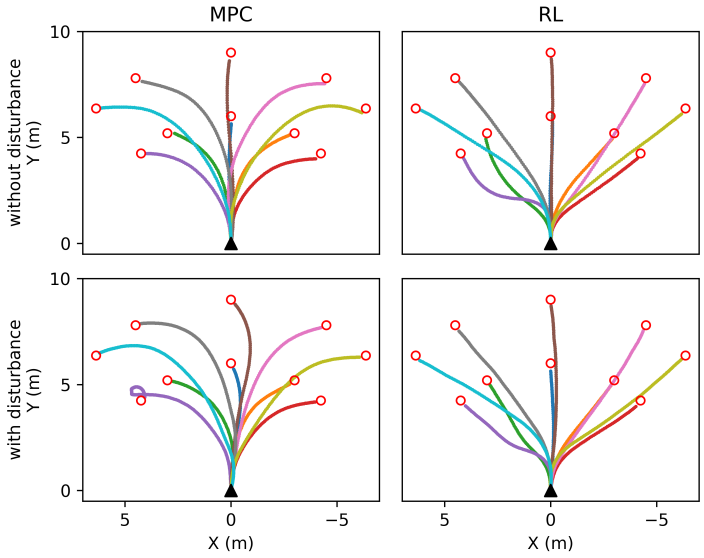


Figure 6.5: Field test trajectories for the ASV under *normal* and *disturbed* conditions. While the RL agent mostly retains trajectory stability, the MPC shows a tendency for deviations aligned with the direction of the external disturbances.

Finally, Table 6.1 highlights the robustness of the agents to the disturbances. The table displays the mean value of each metric across all goals. It shows that the degradation in normalized time, T_{norm}, is similar for both agents, although slightly better for the RL. Nonetheless, the mean value of T_{norm} remains better for the RL agent when compared to the MPC, both in *normal* and *disturbed*. It is also worth noting that the RL agent shows an improvement, i.e negative degradation, in the trajectory deviation metric (Δ_d). The reason is that the drag, added in *disturbed*, slows the vessel, allowing it to initiate earlier rotations toward the target, thereby reducing overall trajectory deviation. Nonetheless, the drawback is the increase in en-

ergy consumption (E_{acc}), which is likely due to its active counteraction of torque and drag forces to maintain straight trajectories.

Agent	Metric	Normal	Disturbed	Degradation %
RL	T_{norm}	1.094	1.247	13.96
	E_{acc}	28.619	36.248	26.66
	Δ_d	0.246	0.160	-34.88
MPC	T_{norm}	1.175	1.350	14.89
	E_{acc}	33.457	36.457	8.96
	Δ_d	0.697	0.775	11.23

Table 6.1: Robustness evaluation: Field Tests Metrics for RL and MPC under *normal* and *disturbed* Conditions

6.4 Discussion

Our results demonstrate that the RL agent effectively reacted to various sources of disturbances, maintaining straight trajectories toward the goal without explicitly modeling or directly observing these disturbances. This outcome suggests that domain randomization, combined with a well-designed simulation environment, can be efficient for real-world ASV applications, particularly in scenarios where disturbances are complex or difficult to quantify.

Nonetheless, achieving robust performance required iterative refinement of the simulation environment and field validation. The primary challenges centered on reducing the simulation-to-reality gap and carefully shaping the reward function to guide the agent toward desired behaviors. Inaccuracies in the simulation model or improper reward settings often led the RL agent to develop unexpected behavior.

By contrast, the primary advantage of the MPC is its structured approach and explainability. For example, by tuning some coefficients from the model (in particular N_r), we were able to achieve the best results for the *normal* condition experiments presented in this chapter.

Finally, the primary limitation observed for the RL agent in field tests was its difficulty to perform in-place rotations, which are necessary for reaching nearby goals with high bearing angles (Figure 6.6). These maneuvers pose significant challenges even for human operators due to the ASV's underactuated and non-holonomic dynamics. However, we observed that the RL policy struggled with in-place rotations significantly more in the field than in simulation. Upon careful analysis, we attribute this discrepancy to differences between the real and simulated actuators, which are further discussed in Section 6.5. This mismatch caused the RL agent to overshoot in-place rotations, ultimately preventing the agent from efficiently reaching the target. In contrast, the MPC adopted a more conservative strategy, either executing slower rotations to align with the target or reversing to reach the desired position.

6.5 Challenges and Lessons Learned

Conducting field experiments with the Kingfisher ASV presented several challenges, primarily related to hardware limitations, control delays, and environmental disturbances. Addressing these issues was essential to ensure reliable data collection and validate our experimental results.

One illustrative issue we encountered was broken propeller blades, as shown in Figure 6.7a. The Kingfisher ASV we used is an aging platform that Clearpath discontinued over a decade ago.

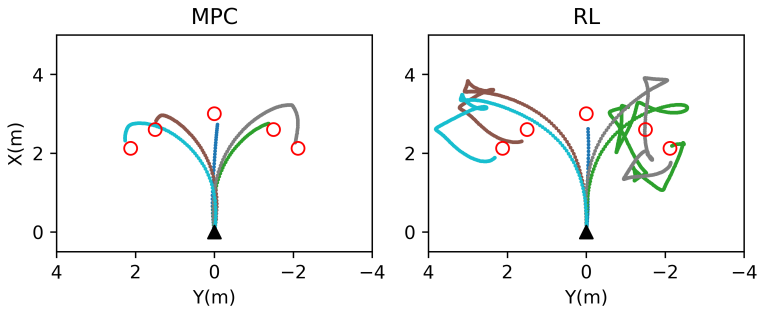


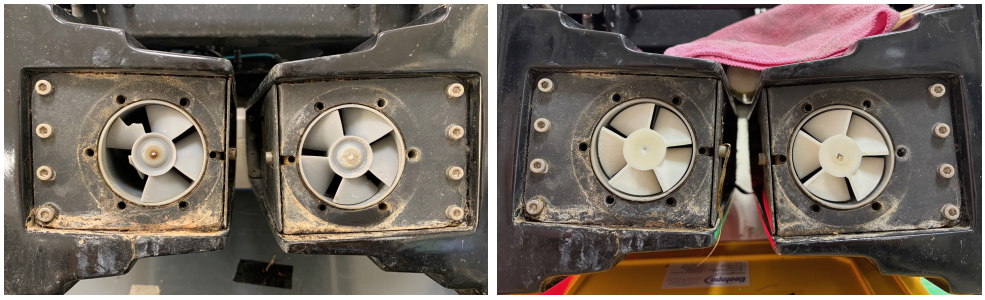
Figure 6.6: Illustration of limitations encountered during field tests. The RL agent struggles with in-place rotations when approaching nearby goals with high bearing angles.

Although it remains operational and robust, regular maintenance is essential to ensure reliable performance. Having the necessary tools and technical expertise for field repairs is indispensable when working with such platforms. In this particular case, replacing the damaged propellers required repeating system identification experiments to update the propeller characteristic force curve, ensuring consistency with our simulation environment. When planning ASV experiments, it is important to allocate repair and maintenance time into the schedule and consider the robot's lifespan when referencing benchmark studies, such as [92].

One of the most significant challenges we encountered was diagnosing the limitation shown in Figure 6.6. Specifically, discrepancies between simulation and field tests raised questions about whether in-place rotation issues stemmed from inaccuracies in hydrodynamic modeling, ASV inertia, or thruster response characteristics. Since these factors are interrelated, isolating their individual contributions based solely on field-collected data was challenging.

The audible response of the propellers to manual commands indicated a potential actuation delay. This prompted further investigation into the proprietary microcontroller responsible for converting input commands into Pulse Position Modulation (PPM) signals for the motor controllers. Using an oscilloscope, we measured the signal propagation and precisely analyzed the commands received by the motor controllers. Our findings revealed that the applied command is rate-limited, meaning that each update is constrained by a maximum allowable change from the previous command. This rate limitation introduced a cumulative delay, resulting in a total response time of approximately 2 seconds when changing the input from -1.0 to 1.0. Understanding this delay is crucial for better modeling, particularly when implementing controllers that require precise actuation timing.

Environmental factors also played a significant role in system performance. One challenge was quantifying the weather disturbances affecting the ASV without specialized sensors such as an anemometer. Initially, we relied on local weather forecasts for wind conditions. However, this approach proved impractical, as localized wind effects in the test area often deviated from forecasted conditions, and the unknown aerodynamics of the ASV made it difficult to directly relate wind velocity to the forces and torques acting on the vessel. To address this, we adopted a simple yet effective alternative: before initiating new experiments, we allowed the ASV to drift freely while recording its position. By analyzing data from the onboard RTK-GPS and IMU, we estimated the accelerations and peak velocities induced by external disturbances. This method provided a practical way to quantify environmental forces acting on the vessel and assess the robustness of the RL policy in real-world conditions.



(a) Broken original propeller (b) 3D-printed replacement

Figure 6.7: The replacement propeller required redesign for feasibility

6.6 Conclusion

This chapter assessed the robustness of a DRL-based controller for ASVs operating in complex and disturbed environments. We trained with domain randomization and conducted extensive simulations in which the agent maintained trajectory stability under variations in mass, rotational damping, and environmental disturbances. Field trials confirmed these findings, demonstrating consistent execution of the floating-waste capture task and reduced task completion time and energy consumption.

Field experiments highlighted one principal limitation: incomplete actuation fidelity in simulation due to missing rate-limited thruster dynamics, which led to overshoot and slow heading corrections. In Chapter 7, we upgrade the simulator to include a rate-limited thruster model that reproduces the delay characterized in Section 6.5 to mitigate the overshoot observed in Figure 6.6. We also integrate a camera-based perception module that detects and classifies floating waste and generates goal positions for the DRL agent, enabling real-time perception and decision-making. The next chapter presents this integrated stack and evaluates its robustness and transferability in simulation and field tests.

Part III

Integration

7

Integrated Autonomous Floating Waste Collection

As established in earlier chapters, autonomous surface vehicles for floating-waste removal operate under variable hydrodynamics, external disturbances, and challenging water-surface perception. This chapter presents a field-validated system that couples the polarimetric vision developed in Chapter 2 with a lightweight learned controller for the detection and capture of floating waste. Camera-based perception converts detections into target points on the water surface. Addressing the shortcomings identified in Chapter 6, we deploy a policy trained entirely in simulation that navigates the vessel to these targets with zero-shot transfer on a retrofitted ASV platform. We conduct matched simulation and field experiments across 14 disturbance regimes, evaluating performance with 6 complementary metrics. While results show near-perfect success and centimeter-level terminal accuracy in most conditions, this chapter focuses on analyzing what enables robustness and what undermines it. The system is further evaluated in a search-and-capture application using onboard camera detections under real-world conditions. In addition to reporting experimental results, we share practical lessons for reaching zero-shot sim-to-real transfer of learned policies. These include methodology details and evaluation protocols that expose failure modes and direct improvements in simulation fidelity where they are most impactful. Key enablers include improved actuation model fidelity, targeted domain randomization, and careful handling of latency and timestamps across modules. We also outline current limitations that motivate decision-layer refinements for managing multiple targets in more complex applications.

This chapter contains the content of an article [93] that is under preparation for submission to IEEE Transactions on Field Robotics with modifications to better integrate with the thesis content.

- **Batista, L.F.W**, Aravecchia, S., and Pradalier, C. "Sim-to-Real Transfer and Robustness Evaluation of Reinforcement Learning Control with Integrated Perception on an ASV for Floating Waste Capture". 2025.[9]

The content has been revised to reduce redundancy, especially in the related work section, while still remaining self-contained. Some deliberate overlap with previous chapters is retained to preserve coherence and avoid excessive fragmentation. Where content recurs, the text cites the relevant earlier chapters and restates key definitions and assumptions for completeness.

[9] *Statement of contributions:* Luis contributed to the study conception, methodology, implementation, and

7.1 Introduction

As discussed in previous chapters, achieving robust autonomy for surface vehicles for floating waste removal presents several challenges. A central difficulty is payload variability during operation. As illustrated by the commercial platform shown in Figure 7.1b, a possible option to capture debris is to equip the ASV with a towed net. As the net fills, changes in payload modify the vehicle's mass, moment of inertia, and hydrodynamic drag, thereby shifting its maneuvering dynamics. Concurrently, exogenous disturbances from wind and currents act on the hull. These factors necessitate real-time control adaptation to maintain accurate and stable operation throughout the cleanup process.

Reinforcement Learning (RL) techniques enable the development of control policies that adapt to model uncertainty and environmental disturbances, with growing adoption in marine robotics and ASVs [6]. Within this context, point-goal navigation formulations for floating-debris interception have been explored in Chapters 5 and 6, showing promising robustness to environmental and payload variability.

With respect to the perception, machine-learning–based object detection has been widely used to quantify floating plastics in natural waters [16]. However, outdoor aquatic scenes exhibit variable illumination with direct sun, shadows, and specular reflections that can degrade camera-based detection. Polarization-aware imaging can exploit the polarization state of reflected light to suppress glare and enhance object contours on the water surface. Results in polarimetric object detection presented in Chapter 2 demonstrate improved robustness to specular highlights in aquatic environments, thereby strengthening detection reliability under challenging illumination.

The goal of this chapter is to advance the robustness of ASVs for autonomously capturing floating waste. Here, we target robustness in perception-driven, point-goal navigation, presenting: (i) the design and implementation of a research platform adapted for capturing floating waste; (ii) a polarimetric target-identification pipeline with quantitative performance characterization; (iii) controlled field experiments that isolate key disturbance sources such as payload variation, perception errors, and mismodelled dynamics to quantify their impact on controller performance; and (iv) end-to-end field trials integrating perception and control in autonomous operation for evaluating the viability of the proposed approach.

By detailing our methodology, observed challenges, and practical lessons from field trials, we present evaluation protocols, development practices, and design choices that enabled zero-shot sim-to-real policy transfer and robust operation. These practical considerations can assist researchers and practitioners in implementing and deploying RL policies on physical platforms. While our application targets ASV architectures, the underlying principles may generalize to field-deployed mobile robots operating under real-world disturbances.

7.2 Related Work

Autonomous floating-waste collection is multidisciplinary and involves perception, control, and systems engineering. In this section, we briefly survey related areas and discuss how they relate to our problem and methodology.

training, designed and executed the simulation experiments, conducted field trials, and drafted the manuscript. Stéphanie contributed to formalizing the robustness definition and manuscript editing. Cédric offered guidance and feedback throughout the entire research process.



(a) Kingfisher Research Platform (b) Jellyfishbot (Courtesy of IADYS)

Figure 7.1: Autonomous surface vessel fitted with nets for retrieving floating waste. The growing load from collected debris changes vessel hydrodynamics and handling.

7.2.1 Autonomous Surface Vehicles and Challenges

Autonomous Surface Vehicles (ASVs) are increasingly deployed in complex, dynamic waterways, where practical autonomy faces several challenges.

As previously introduced, one example is Roboat III [10], an autonomous urban vessel designed for passenger and goods transport. The system employs conventional perception to build an occupancy grid and uses an adaptive Model Predictive Controller (MPC) whose behavior adjusts with payload. This adaptability relies on estimating the payload from the draft and the corresponding underwater volume. In the waste-collection setting, payload changes as debris accumulates in the net, and its estimation is not reliably available, making direct compensation nontrivial.

Huang et al. [94] present a stochastic planning approach for ASV navigation informed by satellite imagery and validated in field trials. Their stack fuses camera and sonar to populate an occupancy grid, projects the waterline to aid local planning, and uses labels derived from the Segment Anything Model [1] for binary water masks. Although the application differs from debris capture, their discussion underscores the challenges that can arise in image-based pipelines operating in unconstrained outdoor conditions. A challenge shared by camera-based perception in our domain.

Other studies explore ASV systems in field deployments, yet focus on applications and tasks different from floating waste collection. For instance, [95] proposes a robust control strategy for intercepting a moving target ASV using traditional model-based controllers. Accurate vessel-dynamics modeling is a core difficulty. The approach localizes the target via GPS without an onboard perception module, highlighting a distinct operating regime from perception-driven target identification.

In this work, we focus on an ASV designed for floating-waste collection. Despite application differences, complex dynamics, payload variability, environmental disturbances, and perception uncertainty are shared. Our work presents design decisions and evaluation methodologies tailored to this application and analyzes their impact in field trials.

7.2.2 Environmental Application

The development of autonomous surface vehicles (ASVs) for environmental applications, particularly for the task of collecting floating waste, has drawn growing interest in recent years. Many works in this area primarily emphasize the mechatronic design of the vehicles tailored for environmental monitoring or waste removal [96–98]. These efforts provide valuable contributions to platform development and environmental applications, yet often lack comprehensive integration of perception and navigation suitable for autonomous operation in complex, real-world environments.

One example is the low-cost ASV for plastic waste collection presented by [99]. While the system is purpose-built for waste retrieval, its focus is on affordability and mechanical design, with limited evaluation of autonomy or perception-driven navigation capabilities.

The cleanup application is also addressed by [100], who integrate perception-based target detection and tracking with an informative path-planning framework validated in field trials. While the context is similar, their focus is on higher-level planning. In contrast, we prioritize quantitative robustness evaluation of a DRL-based control pipeline, including the coupling from perception to control.

7.2.3 Perception

For water-surface applications, Chapter 2 introduced a dedicated dataset with polarimetric imagery to address the challenge of detecting floating debris in outdoor conditions. Empirical results showed that leveraging polarization to suppress water-surface reflections improves detection accuracy. The same dataset was subsequently used in Chapter 3 for semantic segmentation, with studies comparing multiple fusion strategies and reporting increased performance (and inference time) as additional modalities are incorporated. In our setting, a model that outputs the approximate center position of a target is sufficient. With that in mind, this chapter focuses on an object detection pipeline that leverages polarimetric cameras to obtain reflection-attenuated images. For detailed related work on polarimetric-based perception, we invite the reader to refer to Chapters 2 and 3.

7.2.4 Reinforcement Learning-based Control

Deep reinforcement learning (DRL) has demonstrated robust real-world control across mobile platforms, including quadrupeds, full-size humanoids, and vision-based quadrotors [101–103]. In contrast, DRL for ASVs remains at an early stage with limited hardware validation; Chapters 4 to 6 surveyed related work specific to ASVs. The key observations are that adoption is constrained by the difficulty of accurately modeling nonlinear hydrodynamic interactions and environmental disturbances, so existing studies tend to confine evaluation to simulated environments, with few field validations to establish reliable sim-to-real transfer. Although field evaluations are increasing, most real-world studies remain early-stage and focus primarily on path tracking, leaving other ASV tasks underexplored.

Capturing floating waste can be considered as a point-goal navigation task without a terminal-velocity constraint. Chapter 5 introduced a software framework for developing capture tasks and included preliminary field experiments. Chapter 6 subsequently evaluated the robustness of DRL policies in field trials using programmatic goals and identifying limitations in actuation modeling. In this chapter, we improve actuation modeling and perform real-world testing that includes camera-based perception in the loop with a focus on evaluating and understanding the robustness of the trained policy.

7.2.5 Robustness and Policy Transfer

We aim to train a policy that remains robust to external disturbances. In the literature, the robustness of RL agents is often intrinsically linked to uncertainty. As shown in [104, 105], each element of the Markov Decision Process (MDP) is subject to uncertainty: states, actions, transition dynamics, and rewards. Following the taxonomy in [106], we employ Domain Randomization over observations, actions, and dynamics, and apply Reward Shaping to stabilize learning and discourage brittle behaviors (see Section Section 7.5.2).

However, even robust policies may fail to transfer to real hardware; the resulting performance drop is the sim-to-real gap. Complementary surveys [107, 108] attribute this gap to mismatches between simulation and the robot in sensing and appearance, actuation timing and limits, and unmodeled dynamics. In this paper, we evaluate both aspects jointly: we quantify the sim-to-real gap on hardware and probe robustness via controlled perturbations aligned with the MDP components above.

To improve testing methodology and identify failure sources, we implement a simulated perception abstraction described in Section 7.4. Drawing inspiration from Digital Twins in autonomous driving [108], this modularizes interfaces and enables reproducible, component-level validation. Our implementation is a lightweight, targeted abstraction at the perception–control boundary that can be used both in simulation and field tests using the real platform.

Aiming towards zero-shot transfer of a robust policy, our work combines ideas and methodologies from the literature, employing a structured approach to identify performance gaps and improve simulation fidelity where it most affects transfer.

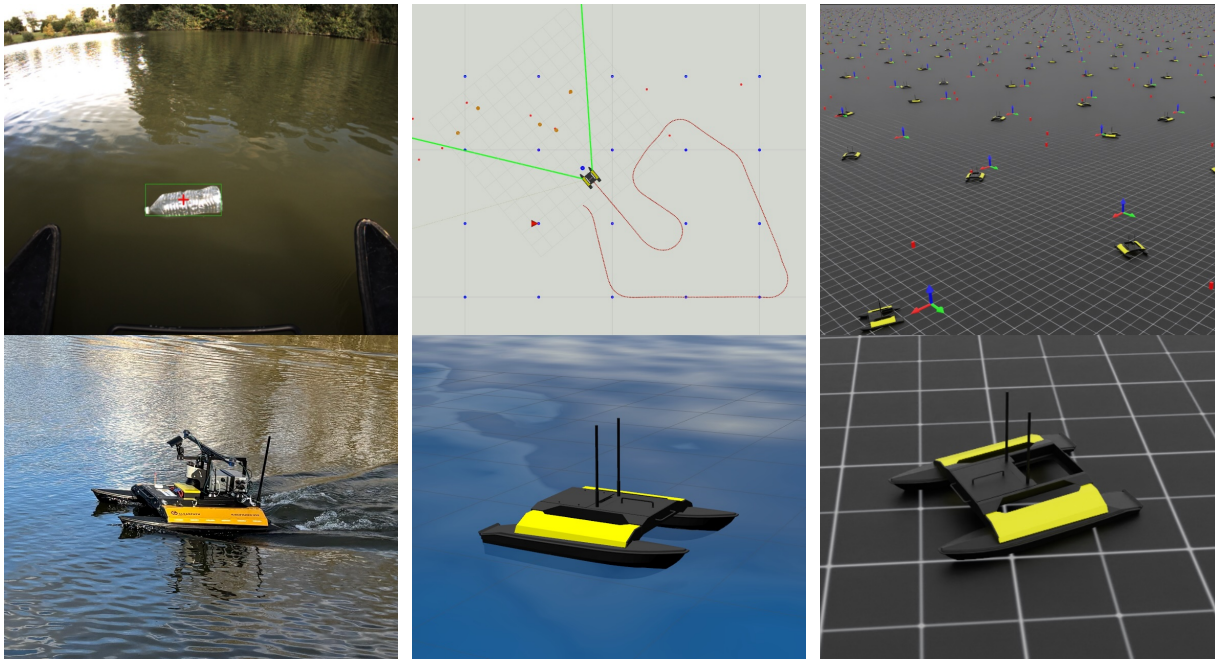
7.3 Experimental Setup

Similar to Chapter 5, this section describes the three environments that are used in this work, illustrated in Figure 7.2. Specifically, we implement an environment based on Isaac Lab to train the reinforcement learning policies; a Gazebo-based simulation for ROS integration and validation; and a Kingfisher ASV research platform for field trials. For completeness, each environment is detailed in a dedicated subsection.

7.3.1 Robot Platform

We conducted field experiments with the *Kingfisher* ASV (Clearpath Robotics), a compact catamaran and predecessor to the *Heron*. This platform was introduced in Section 5.4.5; here we add details and, for completeness, restate all specifications relevant to this chapter. The hull measures 1.35 m (length) by 0.98 m (beam) with a total mass of 35.82 kg. Propulsion is provided by two independently driven thrusters (one per hull), each powered by a 400 W motor. The platform was substantially retrofitted relative to the stock configuration, including replacement of propeller blades, an additional GPU-enabled embedded computer, and additional sensors, as shown in Figure 7.3.

The system is powered by two 22 Ah, 4-cell LiPo batteries, providing over 5 hours of operation in our field trials. A DC–DC step-down converter supplies a regulated 12 V rail. Although the ASV carries a monocular camera and a 2D lidar, these were not used in the experiments reported here; instead, we use a Triton 5.0 MP (TRI050S1-QC) polarimetric camera built with a Sony IMX264MYR sensor, recording color and polarization at 2448×2048 pixels. The unit is paired with a 6 mm lens yielding an angle of view of $80.8°(H) \times 61.6°(V)$ and is mounted forward-facing with a slight downward tilt on the *Kingfisher* to maximize visibility of floating objects up to close



(a) Robot Platform (b) Gazebo Simulation (c) Isaac Lab

Figure 7.2: Three environments used in this work: (a) platform for end-to-end evaluation in field trials; (b) testbed for integration and robustness evaluation in simulation; (c) highly-parallel environment used for policy training.

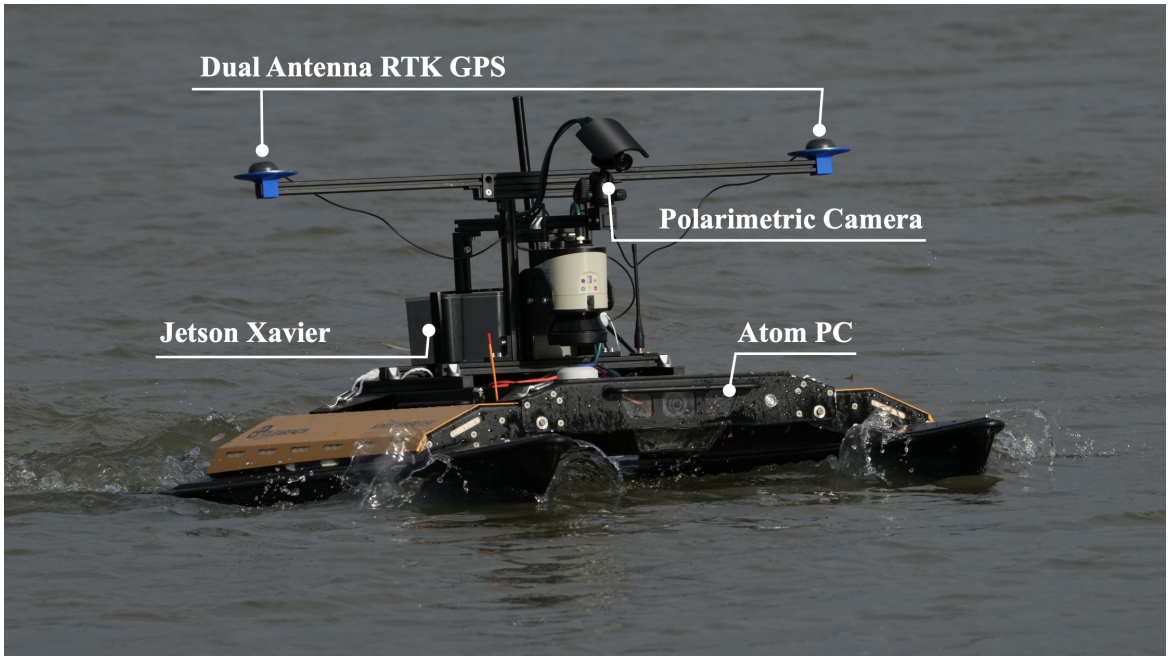
approach (see Figure 7.3). State estimation relies on an SBG Ellipse-D IMU equipped with a dual-antenna RTK-GNSS receiver, achieving position, velocity, and heading accuracies of 0.02 m, 0.03 m/s, and 0.5°, respectively.

Thrust input commands are in the $[-1, 1]$ range. Input commands and thruster force mapping were identified via bollard-pull testing, resulting in the curve presented in Figure 7.4a. Input commands must be provided at a rate of at least 10 Hz and are then processed by a proprietary microcontroller unit (MCU), which outputs PPM (Pulse-Position Modulation) signals to the motor controllers. As identified in Chapter 6, probing the PPM signals revealed that the MCU implements a software rate limiter that constrains command slew and introduces a temporal response shown in Figure 7.4b.

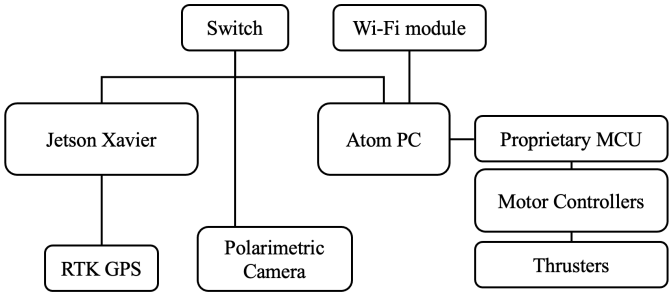
Tasks are distributed between two onboard computers: the Atom PC and the Jetson Xavier. Time synchronization between the two uses NTP during startup. We use the robot operating system (ROS) framework [109] to integrate our components. The Atom PC is connected to the MCU and hosts the ROS master. It handles low-level control, such as issuing thruster commands. The Jetson Xavier handles resource-intensive tasks, including raw image processing, object-detection inference, RL policy inference, and data logging. Access to the ASV during field tests is provided via a Wi-Fi connection, enabling start, stop, and monitoring of experiments from a remote computer simplifying the logistics during field tests.

7.3.2 Integration and Evaluation Testbed

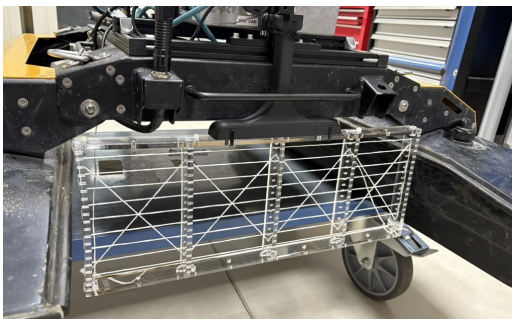
We employ a simulation stack based on *Gazebo* with the *UUV Simulator* plugin [59] to complement the Isaac-based training environment. As previously stated in Chapter 5, this setup



(a) ASV Overview



(b) Block diagram highlighting main components



(c) Frame attached between the hulls

Figure 7.3: The *Clearpath Kingfisher* ASV used in field experiments shown in (a). The ASV is equipped with a diverse set of sensors, including RTK-GPS and a polarimetric camera. It contains two embedded computers for data processing: an Atom PC and a separate Nvidia Jetson AGX with GPU capabilities. The main architecture and system connections are presented in (b). When collecting floating waste, a custom-designed frame was attached to the bottom of the ASV as depicted in (c).

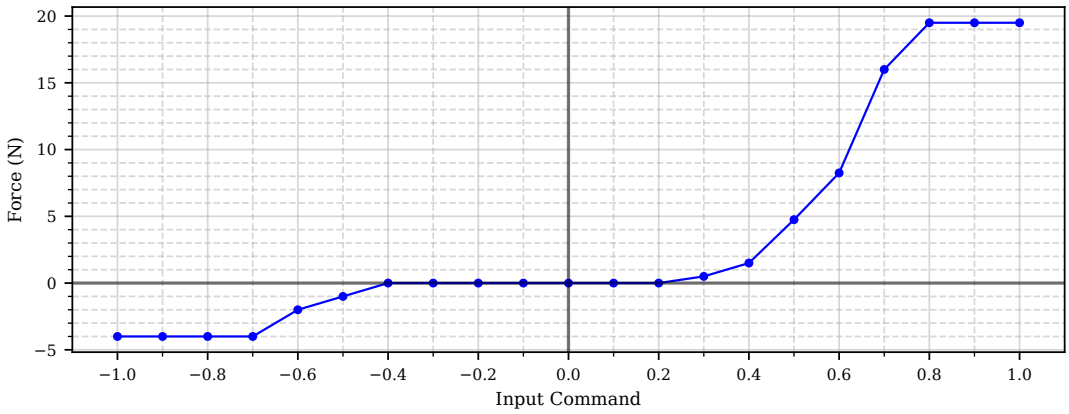
provides an independent physics and hydrodynamics implementation, enabling an initial validation of policy behavior and a check against potential overfitting to the specific dynamics implementation in Isaac Sim. It allows a quick assessment of the consistency of learned behavior before field tests.

Additionally, the Gazebo simulation stack was customized to include a controllable perception-in-the-loop configuration. Instead of a physical camera, we instantiate a calibrated virtual sensor using the intrinsics from camera calibration parameters and a deterministic projection model (Section 7.4.2). Within this pipeline, we can inject controllable detection noise and projection errors to probe robustness to perception bias and uncertainty during policy evaluation in simu-

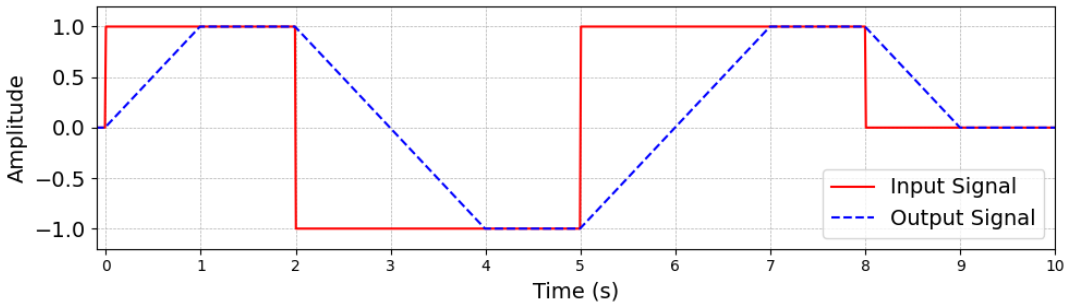
lation.

The UUV Simulator plugin is updated with a new thruster model obtained via system identification experiments on the Kingfisher ASV (Figure 7.4). The force curve is different from Chapter 5 (Figure 5.4) due to the updated propellers (Figure 6.7). The software rate-limit identified in the MCU is replicated as a ROS node that introduces the same response explained in Section 7.3.1 and depicted in Figure 7.4b.

Figure 7.2b illustrates the Gazebo environment used for these experiments, including the vehicle model, simulated targets, and the simulated perception pipeline.



(a) Measured thruster force curve obtained via a bollard-pull test.



(b) Measured MCU command limiting. A full-scale transition from -1 to $+1$ requires up to $\sim 2\,\mathrm{s}$, introducing a significant lag between input and motor controller command.

Figure 7.4: Thruster actuation characterization: (a) static force curve; (b) dynamic command response.

7.3.3 Policy Training Simulator

For training the RL policies, we use an environment based on Isaac Lab [65] due to its batched simulation paradigm that allows massive parallelization. As detailed in Chapter 4, we migrate the core dynamics necessary for ASV simulation (hydrostatics, hydrodynamics, and thruster actuation) on *Isaac Lab 2.1*, allowing for GPU acceleration across many independent rollouts, enabling thousands of environments to be advanced simultaneously. This degree of parallelism substantially increases sample throughput and reduces wall-clock training time, making model-free policy optimization practical for our setting. Figure 7.2 (c) illustrates the environment. The simulation environment used for policy training is implemented in a way that it accepts the same

hydrostatic and hydrodynamic coefficients as the Gazebo-based environment, ensuring parameter parity across platforms. This includes an identical thruster model with an optional software rate limiter to reproduce command-rate constraints. During training, these coefficients and actuation behaviors are exposed to domain randomization, enabling variability while preserving physical consistency. Details of policy training are presented in Section 7.5.

7.4 Polarimetric Perception

This section focuses on the perception itself. The goal here is to detect floating objects in the polarimetric images and convert them into a goal in the robot frame. The detection is done in the image plane, its pixel coordinates are projected onto the water surface, yielding a planar goal in the robot (ASV) frame $\mathbf{g}_r = [x_r \; y_r,]^\top$ for the downstream controller (see Section 7.5). This goal-based abstraction reduces the dimensionality of the state observation and simplifies the coupling between perception and control modules.

7.4.1 Polarimetric Object Detection

We use the polarimetric camera described in Section 7.3.1 to obtain color images with reduced surface specular reflections. This design choice is motivated by the results presented in Chapter 2, which reports that polarimetric sensing improves detection precision in water-rich outdoor environments. To use the polarimetric camera, we first need to extract the color image from the raw image. Our implementation follows the extraction pipeline detailed in Section 2.4.2. Example outputs using the updated camera position are shown in Figure 7.5. In this chapter, we only perform detection on the glare-suppressed images (DIF), which are used throughout our field experiments.

We detect floating waste using *YOLOv5-m* trained on images from the PoTATO dataset (from Chapter 2) with a square input resolution of 1224×1224 pixels. In inference, detections are filtered with a confidence threshold of 0.6 using YOLO's default non-maximum suppression, and the centroid of each bounding box is used as a potential target point for downstream processing. On the PoTATO test split, the model attains mAP@0.5 = 0.921. Despite changes in the camera mounting position, no additional fine-tuning on new images is performed, given that preliminary experiments provided satisfactory results.

7.4.2 Calibration and Projection

In order to establish the relation between image and robot frame coordinates, we first estimate the camera intrinsics. Using checkerboard images spanning the field of view, we calibrate the intrinsic matrix $\mathbf{K} \in \mathbb{R}^{3\times3}$ under the pinhole model, and coefficients $\mathbf{d}$ for the Plumb Bob distortion model.

To compute the extrinsics, we place the ASV in a neutral pose consistent with the pitch observed during field trials and distribute fiducials across three distinct planes at known locations in the robot frame (see Figure 7.6). For each fiducial, we then record its robot-frame coordinates and its undistorted pixel location in the calibrated image, thereby establishing a set of 2D–3D correspondences. Applying a standard PnP solver to these correspondences yields the camera pose with respect to the robot frame, parameterized by the rotation $\mathbf{R}_{cr} \in SO(3)$ and translation $\mathbf{t}_{cr} \in \mathbb{R}^3$, which we use as the extrinsics in the subsequent formulation.

Following projective geometry (see, e.g., [110]), the calibrated intrinsics $\mathbf{K}$ and the extrinsics $(\mathbf{R}_{cr}, \mathbf{t}_{cr})$ can be used to define the pinhole projection matrix $P = \mathbf{K}\,[\,\mathbf{R}_{cr} \mid \mathbf{t}_{cr}\,]$, and $\mathbf{R}_{cr} =$



(a) Normal (RGB) (b) Polarimetric-enhanced (DIF)

Figure 7.5: Example images from the polarimetric camera in the updated position, indicating how polarization can reduce surface reflections.

$\begin{bmatrix}\mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3\end{bmatrix}$. On the water surface we use the plane constraint $Z = 0$, removing the $\mathbf{r}_3$ term and simplifying the plane-induced homography $H = K\,[\,\mathbf{r}_1 \;\; \mathbf{r}_2 \;\; t_{cr}\,] \in \mathbb{R}^{3\times3}$.

With $\tilde{\mathbf{x}}_u = [u \;\; v \;\; 1]^T$ denoting the undistorted image coordinates and $\sim$ the homogeneous equality, we can transform a point ($\mathbf{X}_r = (X, Y, 0)$) from the robot frame to the image plane using Equation (7.1). Similarly, given an undistorted pixel position $\tilde{\mathbf{x}}_u$ and recover water-plane coordinates via Equation (7.2).

$$\tilde{\mathbf{x}}_u \;\sim\; H \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \tag{7.1}$$

$$\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \;\sim\; H^{-1}\,\tilde{\mathbf{x}}_u \tag{7.2}$$

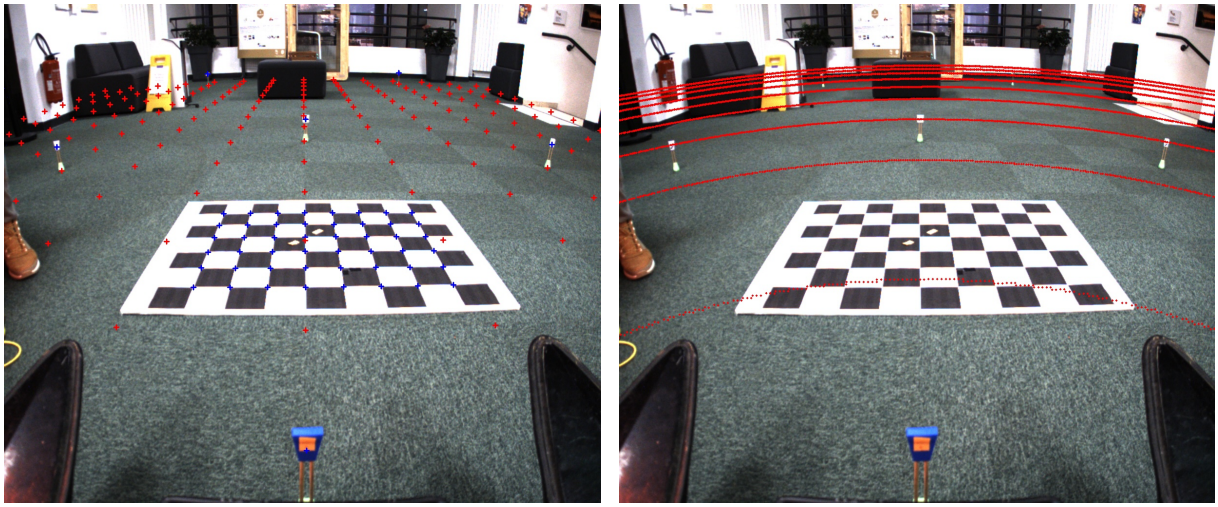


Figure 7.6: Extrinsic calibration and range mapping. *Left*: calibration image with fiducial markers used to estimate the camera–robot transform. *Right*: water-plane distance contours (1–9 m) projected into the image via the calibrated homography.

7.4.3 Perception Pipeline Characterization

In this section, we characterize the perception pipeline in real and simulated settings, reporting onboard end-to-end latency, introducing a simulated perception pipeline that mirrors the real camera, and quantifying water-plane projection errors.

Inference and end-to-end latency on the real platform

We deploy the trained YOLOv5 detector on the Jetson Xavier with TensorRT acceleration and profile the end-to-end latency of the perception stack. The pipeline is decomposed into five stages:

- *Capture*: delay from sensor exposure (message timestamp) until the processing node receives the image; this includes Ethernet switch transit, camera driver, and ROS transport.
- *Extraction*: debayering the RAW polarimetric mosaic and computing the glare-suppressed image using the equations in Section 7.4.1.
- *Preprocess*: conversion to the network tensor (resize/normalize) expected by YOLO.
- *Detection*: TensorRT-optimized inference of the pretrained network on the Jetson Xavier GPU.
- *Projection*: projecting the detections from the image plane to the robot frame via Equation (7.2).

Empirical timings are summarized in Table 7.1. The *Extraction* stage presents the highest latency ($\sim$ 70%). The mean end-to-end latency is $\sim$ 250 ms. In practice, we operate at 2 Hz to preserve headroom and avoid queuing under load. For temporal consistency, each projected goal is published with the original camera sensor timestamp, enabling downstream modules to compensate for the accumulated delay.

Table 7.1: End-to-end perception pipeline latency on a Jetson Xavier: per-stage statistics (ms; mean, std. dev., min, max) computed over ∼500 images.

Step	Mean	Std Dev	Min	Max
Capture	16.03	1.41	15.13	32.96
Extraction	180.00	12.73	169.87	325.34
Preprocess	11.35	1.87	9.86	27.10
Detection	42.52	0.80	41.81	52.95
Projection	0.23	0.37	0.14	7.69
Total Time	248.09	3.73	241.90	261.49

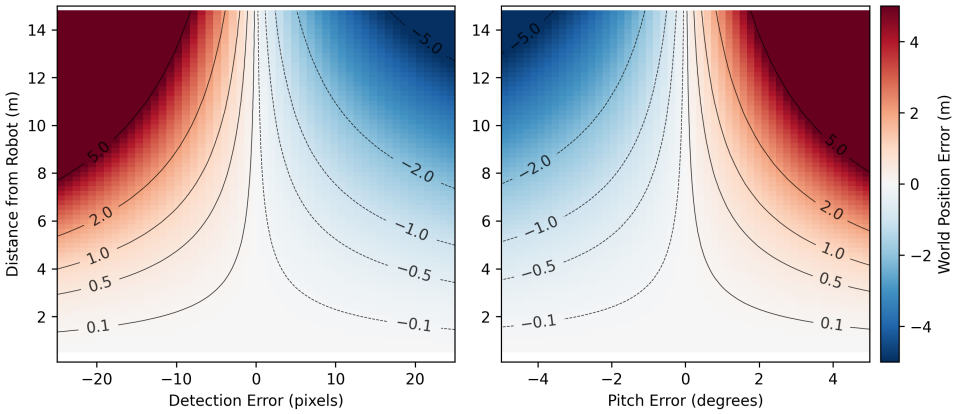


Figure 7.7: Planar position error vs. distance-to-target. Errors are larger at long range and diminish as the ASV approaches the object.

Simulated perception pipeline.

We simulate the perception pipeline by reusing the calibrated projection model from Section 7.4.2, instead of simulating a photorealistic camera running object detection model. Specifically, targets on the water plane $(X, Y, 0)$ are projected to the image with Equation (7.1) and reprojected back to robot-frame goals with Equation (7.2). This design mirrors the real ASV perception characteristics while remaining lightweight and deterministic, enabling controlled studies of how perception imperfections affect control. In particular, we can inject specific error sources such as pixel localization noise in the image plane and pitch bias to quantify their impact on goal accuracy and downstream controller performance.

We also use the simulated perception in some field trials on the real ASV by bypassing the detector and generating programmatic goals with the calibrated projection model. This enables isolating and validating control and dynamics under real disturbances while preserving a deterministic perception abstraction that matches simulation.

Projection error characteristics.

Anticipating the integration with the control component, we provide an analysis of goal detection errors that can arise from the projection model described in Section 7.4.2. Two dominant sources are considered: (i) image-space localization noise (pixel-level uncertainty on the detection center) and (ii) extrinsic calibration errors (pitch bias relative to the neutral calibration). Using the

calibrated intrinsics and extrinsics, we propagate each error source through Equation (7.2) and report the resulting goal position error as a function of target range. The resulting curves (Figure 7.7) show that both effects increase with distance; even small image or pitch perturbations produce large errors at long range, and the induced errors decrease rapidly as the ASV closes in, where the homography becomes well conditioned. Analyzing data from field trials, we observe pitch variation within $\pm 2^\circ$ and detection errors within $\pm 10\,\mathrm{px}$. For these ranges (Figure 7.7) at $r \approx 6\,\mathrm{m}$ the induced error is on the order of 1 m, and within $r \leq 2\,\mathrm{m}$ it falls below 0.1 m. Overall, these effects have a much larger impact on distance estimation at long range and become negligible at close range.

In summary, this subsection measured end-to-end perception delay, defined a lightweight perception abstraction, and characterized projection errors. These results ground controller design (presented in Section 7.5), enable latency compensation, and provide a clear basis for interpreting field behavior. The perception abstraction further supports progressive testing using deterministic goals that are repeatable in simulation and field trials.

7.5 Reinforcement Learning-based Control

This section describes a low-level controller based on reinforcement learning. We present a lightweight policy that achieves zero-shot simulation-to-real transfer and supports robust point-goal navigation. Field experiments quantify performance, surface limitations, and distill key design considerations. The approach closely follows Section 5.4.2 but differs in observations, rewards, a smaller actor–critic architecture, and additional evaluation metrics. To avoid excessive fragmentation with numerous cross-references, we prioritize a complete description, explicitly mentioning important deviations from Chapter 5 at the point where they arise.

7.5.1 Problem Formulation

Our problem is modeled as a Markov Decision Process (MDP; [111]) described by the quintuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, r, \gamma \rangle$. Here, $\mathcal{S}$ denotes the state space, $\mathcal{A}$ the action space, $\mathcal{T}(s' \mid s, a)$ the transition function, $r(s, a)$ a scalar reward, and $\gamma \in [0, 1]$ the discount factor. At each discrete time t, the agent observes a state $s_t \in \mathcal{S}$, samples an action $a_t \in \mathcal{A}$ from its policy $\pi(a_t \mid s_t)$, receives a reward $r_t = r(s_t, a_t)$, and transitions to the next state $s_{t+1} \sim \mathcal{T}(\cdot \mid s_t, a_t)$.

Formally, we do not have access to the full state of the system, but instead to observations. However, for simplification, and because our RL agent is based on Proximal Policy Optimization (PPO) [83], we consider the observation to be the state ($s_t := o_t$). To summarize, at every step the environment provides a state observation s_t, the reward r_t, and a termination flag $d_t \in \{0, 1\}$. The agent's objective is to maximize the expected discounted return over episodes of horizon T.

$$J(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{T} \gamma^t \, r(s_t, a_t) \right]$$

7.5.2 Methodology

Simulation Setup

The dynamic model of the ASV used in simulation follows Equation (5.1) and Equation (5.2) introduced in Section 5.3.2. In summary, the dynamic model is a simplified representation: it groups hydrodynamic effects into linear and quadratic damping in surge, sway, and yaw; includes restoring forces and moments from gravity and buoyancy; accounts for thruster-generated

forces and moments and external disturbances (e.g., wind, waves); and neglects Coriolis effects. The dynamic model used here is a deliberately simplified abstraction of reality, capturing the dominant vessel dynamics while omitting secondary interactions and higher-order couplings.

Task Description

Similar to the approach described in Chapters 5 and 6, the RL agent is based on the PPO algorithm, using an actor-critic architecture with two hidden layers. This setup is designed for efficient control of the ASV's thrusters in dynamic environments.

The dynamic model is implemented in simulation to train an agent capable of autonomously controlling the ASV's thrusters for the task of capturing floating waste. To perform this task, the ASV navigates over the waste, capturing it with a fixed net positioned between its hulls. In simulation, the full 3D state evolves under the physics engine, but the control objective and policy operate on the planar manifold because the water surface is modeled as locally flat, enabling a $2D$ formulation.

As described in Figure 5.5, the target is specified in a two-dimensional space. Accordingly, states, goals, and rewards are computed from planar variables; out-of-plane motions (heave, roll, pitch) are treated as exogenous disturbances that affect dynamics but are not part of the policy input or objective.

The coordinates x and y represent the position of the ASV's center in a fixed world frame, and d_t indicates the planar distance to the goal. The task is considered successful when the ASV passes within 0.15 m of the target's position relative to its center. This is a conservative threshold since the boat's dimensions allow capture of floating waste at distances up to 0.30 m. To focus solely on evaluating the capture task, the environment is kept free of obstacles. Perception field-of-view constraints are not enforced during training; the goal may be placed at any relative bearing and range with respect to the boat.

Action space: The action space, $\mathbf{u}_t = [\mathbf{u}_t^{\text{left}} \quad \mathbf{u}_t^{\text{right}}]^T$, consists of the command inputs for the left and right thrusters, respectively, at time step t. These inputs are transformed into thrust forces based on the characteristic force curves obtained from system identification experiments.

Observation space: At each time step t, the observation space is defined as $\mathbf{o}_t \in \mathbb{R}^8$:

$$\mathbf{o}_t = [\mathbf{u}_{t-1}^{\text{left}} \ \mathbf{u}_{t-1}^{\text{right}} \ u_t \ v_t \ t_t \ \cos(\delta_t^{\text{head}}) \ \sin(\delta_t^{\text{head}}) \ d_t]^T \tag{7.3}$$

The components consist of actions from the previous step $\mathbf{u}_{t-1}^{\text{left}}$ and $\mathbf{u}_{t-1}^{\text{right}}$. This differs from Chapter 6, where the observation vector did not include previous actions. Including previous actions enables the policy to infer hidden state under partial observability and improves learning stability. The surge speed u_t, sway speed v_t, and yaw rate r_t, which describe the linear and angular velocities of the ASV in the body frame. The orientation of the ASV relative to the goal is represented by $cos(\delta_t^{\text{head}})$ and $\sin(\delta_t^{\text{head}})$, providing a continuous representation of the bearing of the target. The final element d_t denotes the distance to the goal at time step t (Figure 5.5).

Reward function: We design a compound per-step reward r_t (Equation (7.4)) to encourage progress toward the goal, alignment of the vessel's heading, smooth and energy-aware actuation, operation within a desired velocity range, time pressure, and a terminal success bonus. The reward function parameters, detailed in Table 7.2, have been meticulously redesigned in comparison to previous chapters to enhance the agent's performance.

In Table 7.2, d_t is the distance to the goal, δ^{head} is the bearing (heading error) to the goal, d_{thr} and δ_{thr} are distance and bearing thresholds, Δt is the step time, $E_{\max}$ normalizes energy, and v_t is the forward body-frame speed constrained by $[v_{\min}, v_{\max}]$. Hyperparameters $\lambda_{1\ldots7}$ weight each

Table 7.2: Per-step rewards, weights, and parameters

Term	Equation	Weight	Parameters (value / unit)	Purpose
r_{1_t}	$(d_{t-1} - d_t)$	$\lambda_1 = 1.0$	—	Reward distance progress.
r_{2_t}	$\left(\cos\delta_t^{\text{head}} - \cos\delta_{t-1}^{\text{head}}\right)$	$\lambda_2 = 5.0$	—	Reward-bearing progress.
r_{3_t}	$max\{0,\, 1 - (\delta_t^{\text{head}}/\delta_{\text{thr}})^2\,\}$	$\lambda_3 = 0.01$	$\delta_{\text{thr}} = 0.1$ rad	Bonus when bearing is within a small angular threshold.
r_{4_t}	$(\lvert\mathbf{u}_t^{\text{left}}\rvert^2 + \lvert\mathbf{u}_t^{\text{left}}\rvert^2)\,\Delta t/E_{\max}$	$\lambda_4 = -0.1$	$E_{\max} = 2.0$ [–], Δt [s]	Penalize excessive actuation.
r_{5_t}	$e^{\,\kappa\,\max\{0,\, v_t - v_{\max},\, v_{\min} - v_t\}} - 1$	$\lambda_5 = 1.0$	$v_{\min} = 0$ $v_{\max} = 0.6$ $\kappa = -10$	Penalize operation outside the desired velocity band.
r_{6_t}	1	$\lambda_6 = -0.05$	—	Apply time pressure to shorten episode duration.
r_{7_t}	$\begin{cases} 1, & d_t < d_{\text{thr}} \\ 0 & \text{otherwise} \end{cases}$	$\lambda_7 = 10.0$	$d_{\text{thr}} = 0.1$ m	One-time success reward upon reaching the goal.

term, and κ sets the velocity-penalty curvature. A low value for d_{thr} was deliberately chosen to be lower than the actual distance required to capture the target. The reward function is then:

$$r_t \;=\; \sum_{n=1}^{7} \lambda_n r_{n_t} \tag{7.4}$$

Policy Training

Table 7.3: Main PPO hyperparameters

Runner / Data Collection	
Horizon length (steps)	48
Max iterations	300
Policy / Networks	
Actor hidden dims	64×64 (ELU)
Critic hidden dims	64×64 (ELU)
Init action std (σ_0)	1.0
Optimization (PPO)	
Learning rate	5×10^{-4}
Schedule	adaptive
PPO epochs per update	5
Mini-batches per update	4
Clip range ϵ	0.2
Value loss coef. c_v	1.0
Value loss clipping	Yes
Discount γ	0.99
GAE λ	0.95
Target / desired KL	0.01

Training was carried out on an NVIDIA GeForce RTX 3080 (10 GB VRAM) and completed in

approximately 10 minutes for 300 epochs. Table 7.3 presents the key learning configurations that were used. We employed *domain randomization* comprising: (i) observation noise (sensor noise) with position $\pm 0.03\,\mathrm{m}$ and orientation $\pm 0.025\,\mathrm{rad}$; (ii) actuation noise (action perturbations); (iii) randomized initial forward velocity $v_0 \sim \mathcal{U}(-0.7,\, 0.7)\,\mathrm{m/s}$; (iv) center-of-mass (CoM) perturbations uniformly sampled within $\pm 0.05\,\mathrm{m}$ to model payload shifts; and (v) external wrench disturbances emulating wind and currents with forces $F_x, F_y \sim \mathcal{U}(-0.3,\, 0.3)\,\mathrm{N}$ and yaw torque $\tau_z \sim \mathcal{U}(-0.3,\, 0.3)\,\mathrm{N{\cdot}m}$. We additionally applied up to 10% multiplicative variation in thruster effectiveness to account for battery state and partial loss-of-effectiveness, while hydrodynamic parameters were kept fixed. Training was repeated with five independent random seeds; because learning curves and final returns converged to comparable values and the learned policies exhibited qualitatively similar behavior, a single seed was selected for all subsequent experiments and evaluation.

Evaluation Metrics

One of the key focus of our work is evaluating the robustness of the deployed RL agents and the sim-to-real transferability.

We test the robustness of the RL agent with respect to different types of disturbances using complementary metrics to assess efficiency, stability, accuracy, and task completion. Although trials may continue until a minimum-distance criterion is met, metrics are computed solely on the trajectory of the first approach to the goal, thereby excluding any post-pass corrections; this choice aligns with the application and prevents metric distortion. The first approach is defined as the first crossing of the line through the goal that is orthogonal to the start–goal segment (see Figure 7.10). All data recorded after this crossing are excluded from metric computation, and the agent's position at the crossing is used to compute the *success rate*. The same procedure is applied consistently in simulation and field experiments. The definitions, units, and interpretations of all metrics are summarized in Table 7.4.

We also evaluate our policies using a sim-to-real gap metric based on [106]. This allows for evaluating transferability and prioritizing improvements that might be needed to increase simulation fidelity.

Let $\mathcal{E}_{sim}$ be the simulated environment and $\mathcal{E}_{real}$ the real world. As introduced in Section Section 7.2.5, the policy trained in simulation solves the MDP $\mathcal{M}_s : \langle \mathcal{S}_s, \mathcal{A}_s, \mathcal{T}_s, r_s, \gamma \rangle$, corresponding to $\mathcal{E}_{sim}$. However, when deployed in $\mathcal{E}_{real}$, the MDP $\mathcal{M}_r$ is different.

Let π_s be the policy the RL agent learned from $\mathcal{M}_s$. Let Ψ be the evaluation metric selected to quantify the performance of the policy: Ψ_s corresponds to the evaluation conducted in $\mathcal{E}_{sim}$, and Ψ_r in $\mathcal{E}_{real}$. Over n experiments, the sim-to-real gap $\mathcal{G}(\pi)$ can be formulated as:

$$\mathcal{G}(\pi) = \frac{1}{n} \left| \sum_{i=1}^{n} \Psi_s^i(\pi_s) - \sum_{i=1}^{n} \Psi_r^i(\pi_s) \right| \tag{7.5}$$

Ψ is measured identically in $\mathcal{E}_{sim}$ and $\mathcal{E}_{real}$. Ψ can be any evaluation metric relevant to the task. In this work, we consider Ψ_{NT}, Ψ_{NE}, Ψ_{ER}, Ψ_{PD}, Ψ_{FD}, Ψ_{SR}, described in Table 7.4.

7.5.3 Experiments

To assess policy robustness, we conducted matched experiments in simulation and in field trials. Specifically, we evaluate the zero-shot robustness of a policy trained exclusively in simulation, and we characterize how different types of perturbations affect performance under realistic conditions.

Table 7.4: Definition of the six evaluation metrics used to assess policy performance. The table reports the symbol, name, mathematical formulation, unit, and brief interpretation of each metric. d_1 and d_N represent the initial and final distances, respectively.

Metric	Equation	Unit	Type	Description
Ψ_{NT} (*Normalized Time*)	$\Psi_{\mathrm{NT}} = \dfrac{T}{d_1}$	$\mathrm{s\,m^{-1}}$	*efficiency*	Episode duration normalized by straight-line distance to goal.
Ψ_{NE} (*Normalized Energy*)	$\Psi_{\mathrm{NE}} = \dfrac{\sum_{t=1}^{N}\left(\lvert\mathbf{u}_t^{\mathrm{left}}\rvert + \lvert\mathbf{u}_t^{\mathrm{right}}\rvert\right)}{d_1}$	$(\mathrm{au})\mathrm{m}^{-1}$	*efficiency*	Accumulated actuator effort normalized by straight-line distance to goal.
Ψ_{ER} (*Excess Rotation*)	$\Psi_{\mathrm{ER}} = \sum_{t=2}^{N}\lvert\psi_t - \psi_{t-1}\rvert - \lvert\delta_1^{\mathrm{head}}\rvert$	rad	*stability*	Total heading change beyond the initial bearing-to-goal offset.
Ψ_{FD} (*Final Distance*)	$\Psi_{\mathrm{FD}} = d_N$	m	*accuracy*	Terminal distance to goal.
Ψ_{PD} (*Path Deviation*)	$\Psi_{\mathrm{PD}} = \sum_{t=2}^{N}\lvert d_t - d_{t-1}\rvert + d_N - d_1$	m	*accuracy*	Excess path length relative to a straight-line distance to the goal.
Ψ_{SR} (*Success Rate*)	$\Psi_{\mathrm{SR}} = \begin{cases} 1, & d_N \le 0.15\ \mathrm{m} \\ 0, & \text{otherwise} \end{cases}$	—	*task completion*	Binary success under a 0.15 m tolerance.

Experiment Description

To conduct our evaluation, each trial consisted of a sequence of programmatic goal positions at distances $3, 6, 9\,\mathrm{m}$ and bearings spanning $[-30°, 30°]$ in $15°$ increments, corresponding to the camera's field of view and covering a representative subset of typical goal configurations. The identical goal set was repeated under 14 experimental conditions to quantify performance degradation under distinct disturbance types. Using the previously defined perception abstraction (Section 7.4.3), we reproduced the same goal sequences and disturbance profiles in both simulation and field trials, achieving experiment parity and isolating the control policy from perception and planning effects. This setup injects controlled disturbances while holding observations constant, enabling direct measurement of performance degradation and assessment of the sim-to-real gap.

In Chapter 6, we observed that the Kingfisher's microcontroller enforces a command rate limit that was absent from the nominal simulator, producing systematic discrepancies between commanded and realized thrust (Section 7.3.1, Figure 7.4b). To isolate the effect of this modeling mismatch, we evaluate two policies trained under identical settings (Section 7.5.2) that differ only in actuator modeling: a *Refined Actuator Policy* that incorporates the empirically measured rate limiting, and a *Naive Actuator Policy* that omits it. The name and context of each disturbance condition are as follows:

Refined Actuator Policy **(trained with rate limit)**:
[No disturbance]: ***01-Ideal:*** Baseline with nominal conditions without any deliberate perturbations.

[Disturbance on the States]: ***02-LocDelay:*** A fixed 100 ms delay is injected into the localization estimate before it is consumed by the policy state. ***10-PNoise05px***, ***11-PNoise25px***,

12-PNoise50px: Image-plane perception noise. Using the calibrated camera model (Section 7.4.2), the simulated target is projected to the image plane; a uniform perturbation within radius $r \in \{5, 25, 50\}$ pixels is applied, and the point is reprojected to the robot frame. Noise is resampled at 2 Hz, matching the camera frame rate.

[Disturbance on the Actions]: ***05-ThrLoE10***, ***06-ThrLoE30***, ***07-ThrLoE50:*** Loss-of-effectiveness on the right thruster where the input command is scaled $u^{right} = (1 - \alpha)u^{right}$ with $\alpha \in \{0.10, 0.30, 0.50\}$, corresponding to $90\%, 70\%, 50\%$ of nominal efficiency. This models thruster imbalance or partial failures (e.g., fouling). Scaling is applied to the command rather than to a thrust model, enabling direct programmatic replication on the real ASV.

[Disturbance on the Transitions]: ***08-DynPert***, ***09-DynPertStr:*** Dynamic perturbations (normal/strong). In simulation we increase total mass from 35 to 41.25 kg, shift the center of gravity laterally ($\text{cog}_y = -0.10$ m for normal; -0.20 m for strong), and increase quadratic surge damping coefficient from 17.26 to 25.89 (normal) or 34.52 (strong). In field trials, we attach a 5.2 kg payload to induce added mass, drag, and torque. We switch between two payload attachment locations for different levels of disturbance: normal (visible in Figure 7.1a) and strong (≈ 10 cm further to the right).

[Combined Perturbations]: ***13-CombPert:*** Moderate combined perturbations, comprising *05-ThrLoE10* (LoE), *08-DynPert* (dynamics), and *10-PNoise05px* (perception noise). Represents a realistic scenario, likely to happen in the application of collecting floating waste. ***14-CombPertStr:*** Strong combined perturbations, comprising *06-ThrLoE30* (LoE), *09-DynPertStr* (dynamics), and *12-PNoise50px* (perception noise). Represents a more extreme scenario, less common in the application of collecting floating waste.

Naive Actuator Policy **(trained without rate limit)**:
[Disturbance on the Actions]: ***03-NoRateLim:*** The policy is trained in simulation without the actuator rate limiter enforced on the MCU, creating a deliberate sim-to-real mismatch.

[Disturbance on the Actions and States]: ***04-LDandNRL:*** Setting combining *02-LocDelay* and *03-NoRateLim* to emulate mis-modeled actuation and mis-tuned localization; as these factors are correctable, we do not include this case in subsequent combined-perturbation experiments.

Results and Discussion

Across 14 experimental conditions, each evaluated on 15 programmatic goals, we conducted 420 trials (simulation + field). Quantitative performance is presented in Figure 7.9 (six metrics) and summarized with mean values, along with sim-to-real gaps in Table 7.5. Qualitative behavior is shown by the actual trajectories in Figure 7.8.

Overall analysis These results show that the RL-based controller achieves high *Success Rate* (Ψ_{SR}) across a wide range of perturbation scenarios. The success rate stays at 1.0 in 23 out of 28 experiments in simulation and field trials (Table 7.5), with the only exceptions being the experiments *03-NoRateLim* and *04-LDandNRL*, with large disturbance in the Actions due to a mismatch of the actuators models from training and deployment worlds, as well as a single missed approach in *07-ThrLoE50* field tests, an experiment also with high disturbance on the Actions.

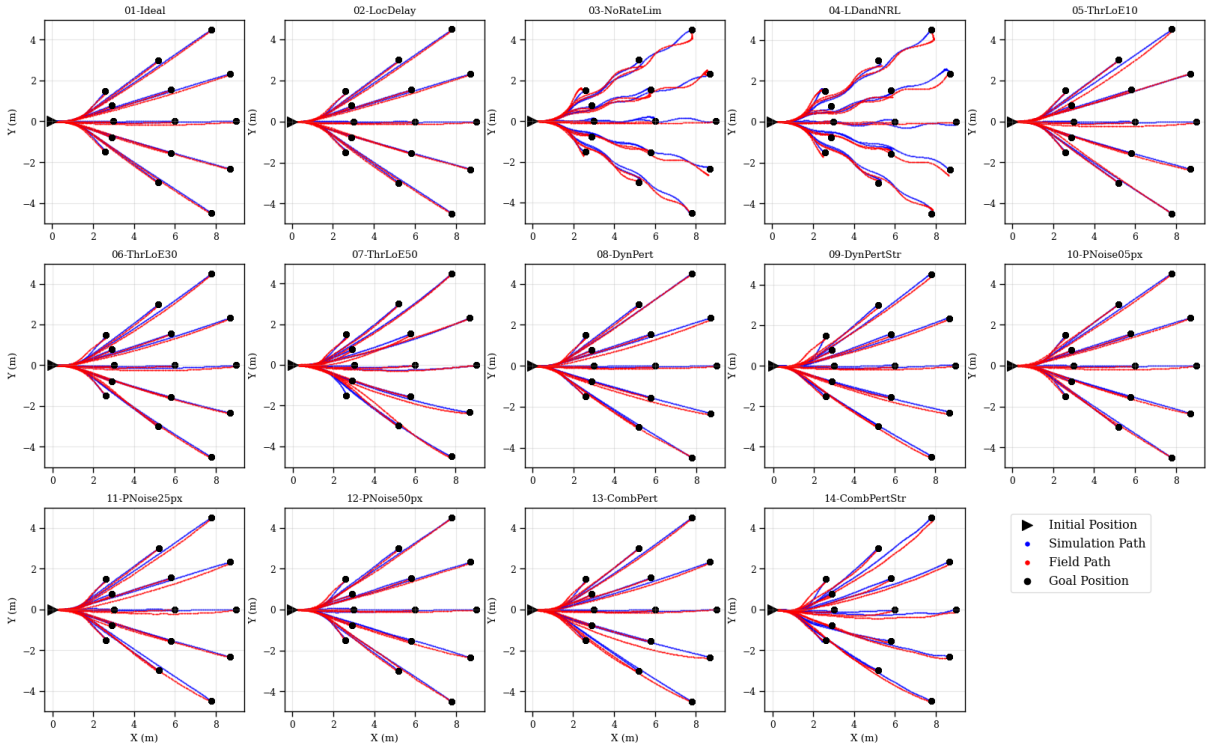


Figure 7.8: Simulated and field-test paths for all experiments. Most cases show no visible degradation; *03-NoRateLim* and *04-LDandNRL* exhibit rotation-induced oscillations, and *14-CombPertStr* shows a slightly larger deviation from the optimal path. Trajectories after the first crossing of the goal's orthogonal line are not displayed. Looking closely at *07-ThrLoE50*, one field trial did not reach the minimal distance to the goal (near position $(2, -2)$).

Degradations with significant effect on robustness A strong degradation is observed in the *03-NoRateLim* condition, which does not include actuator rate limiting during training. In this case, *Excess Rotation* (Ψ_{ER}) rises sharply from nominal values around 0.2 rad (*01-Ideal*) to more than 2.2 rad in simulation and 2.15 rad in field (Table 7.5). This also doubles the *Normalized Energy* (Ψ_{NE}), from ≈ 32 to above 60. The mismatch highlights the sensitivity of the policy to actuator dynamics, particularly rate limits, which can be difficult to model precisely in proprietary designs and can be confounded by the high inertia of aquatic platforms.

Similarly, *04-LDandNRL* (localization delay + missing actuator rate limit modeling) shows pronounced degradation across all metrics, with trajectory-level oscillations visible in Figure 7.8. By contrast, *02-LocDelay* alone remains near nominal. Taken together, these results indicate that (i) only modest localization delay, hence small error on the States, has no significant impact, whereas (ii) missing actuator rate limiting, leading to a large error on the Actions, especially when combined with localization delays, drives the strongest degradation in efficiency, stability, and success.

Degradations with limited effect on robustness Beyond rate-limit effects, the only other failure occurs under *07-ThrLoE50* (50% loss-of-effectiveness on the right thruster, high perturbation on the Actions). It fails to reach the goal on the first pass, so Ψ_{SR} drops to 0.93 (field). Because LoE is applied to the input command signal, the backward force is limited to 1 N (Fig-

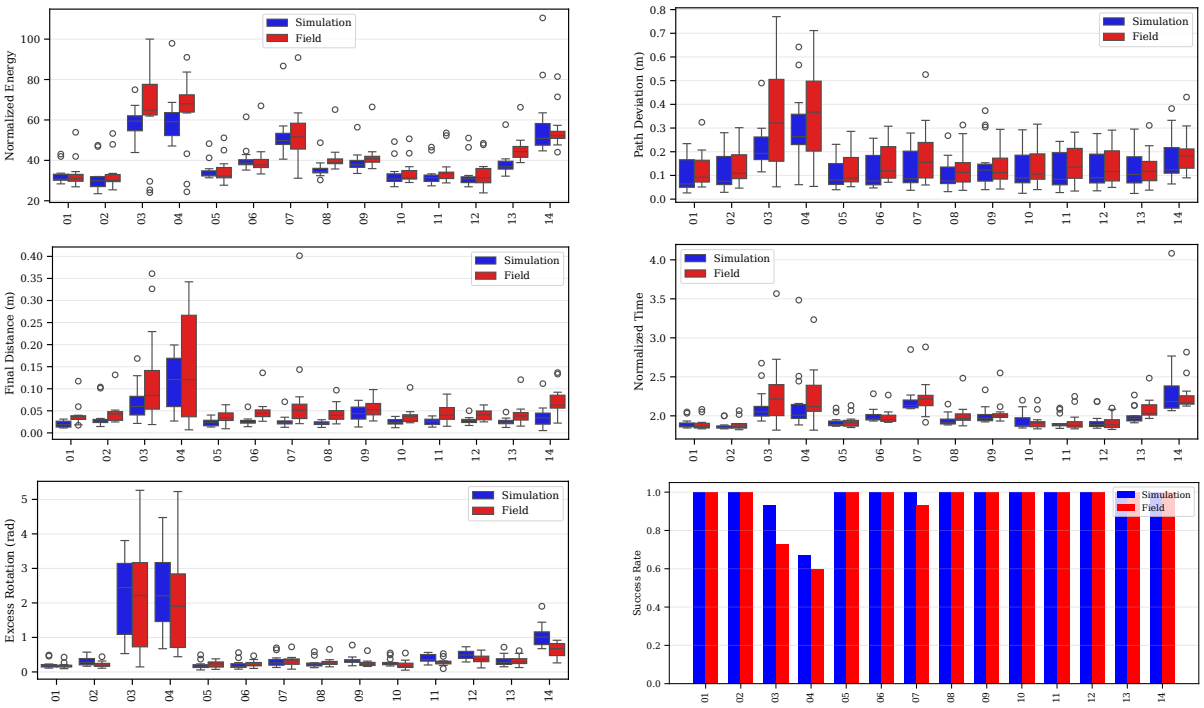


Figure 7.9: Bar plots of the six metrics across all experimental conditions. Each bar plot corresponds to metrics from the 15 goals shown in Figure 7.8. Degradation is pronounced for *03-NoRateLim* and *04-LDandNRL* due to mismodeled thruster dynamics; other disturbances show robust performance with only modest variation. Results are largely consistent between simulation and field tests.

ure 7.4a), reducing differential thrust and preventing a sharp clockwise rotation. Even in these degraded situations, the policy managed to eventually recover and reach the goal, demonstrating that even if not approaching in an optimal distance, recovery was possible. This is presented in Figure 7.10.

For moderate thruster loss-of-effectiveness (*05-ThrLoE10*, *06-ThrLoE30*, smaller perturbations on the Actions), the controller still succeeds in all trials with minimal accuracy loss ($\Psi_{FD} \leq 0.05\,\text{m}$), but efficiency is compromised. Ψ_{NE} increases by 20–25% compared to the ideal case, reflecting the additional actuation needed to compensate for reduced thrust. A similar effect is seen in *08-DynPert* and *09-DynPertStr*, perturbations on the Transition, where field values of Ψ_{NE} increase from 32.9 under ideal conditions to 41.0 and 42.1, respectively, while Ψ_{NT} and Ψ_{FD} remain close to nominal. These results suggest a clear robustness–efficiency trade-off: the agent maintains success and accuracy at the cost of higher energy expenditure.

A noteworthy outcome is the resilience to perception noise (i.e, State perturbation). Even with severe perturbations of 50 pixels (*12-PNoise50px*), which exceed by an order of magnitude the noise levels observed in practice (Section 7.4), the controller achieves $\Psi_{SR} = 1.0$ with negligible loss of accuracy ($\Psi_{FD} \leq 0.04\,\text{m}$). Only a small increase in Ψ_{ER} is observed (field: 0.36 rad, compared to 0.18 rad in the ideal case), indicating strong policy robustness to perception noise.

The combined perturbation experiments further support the robustness of the trained policy. Under *13-CombPert*, performance remains close to nominal ($\Psi_{FD} = 0.04\,\text{m}$, $\Psi_{SR} = 1.0$), with an increase in Ψ_{NE} (45.6 field vs. 32.9 ideal). For the stronger combined disturbances *14-CombPertStr*, the degradation becomes more evident: Ψ_{NT} grows to 2.25 (field), Ψ_{NE} rises close

Table 7.5: Mean performance (Ψ) and sim-to-real gap ($\mathcal{G}$) metrics for each experimental condition. For each metric/column, the values with the highest degradation are highlighted.

Experiment	Simulation						Field						Sim-to-real Gap					
	Ψ_{NT}	Ψ_{NE}	Ψ_{FD}	Ψ_{ER}	Ψ_{PD}	Ψ_{SR}	Ψ_{NT}	Ψ_{NE}	Ψ_{FD}	Ψ_{ER}	Ψ_{PD}	Ψ_{SR}	$\mathcal{G}_{NT}$	$\mathcal{G}_{NE}$	$\mathcal{G}_{FD}$	$\mathcal{G}_{ER}$	$\mathcal{G}_{PD}$	$\mathcal{G}_{SR}$
01-Ideal	1.9	32.52	0.02	0.21	0.11	1	1.89	32.88	0.04	0.18	0.13	1	0.01	0.36	0.02	0.03	0.02	0
02-LocDelay	1.87	31.34	0.04	0.31	0.12	1	1.88	32.92	0.04	0.21	0.14	1	0.01	1.58	0	0.1	0.02	0
03-NoRateLim	**2.12**	**58.51**	**0.07**	**2.22**	**0.22**	**0.93**	**2.27**	**64.35**	**0.12**	**2.15**	**0.35**	**0.73**	**0.15**	**5.84**	**0.05**	0.07	**0.13**	**0.20**
04-LDandNRL	**2.16**	**60.41**	**0.12**	**2.31**	**0.31**	**0.67**	**2.23**	**63.95**	**0.15**	**2.06**	**0.36**	**0.6**	0.07	3.54	0.03	**0.25**	0.05	0.07
05-RThrLoE10	1.92	34.63	0.02	0.19	0.11	1	1.92	34.8	0.04	0.22	0.13	1	0	0.17	0.02	0.03	0.02	0
06-RThrLoE20	2	40.55	0.03	0.21	0.12	1	1.98	40.16	0.05	0.23	0.15	1	0.02	0.39	0.02	0.02	0.03	0
07-RThrLoE50	2.2	52.16	0.03	0.31	0.13	1	2.23	**53.08**	0.08	0.32	0.19	**0.93**	0.03	0.92	0.05	0.01	0.06	0.07
08-DynPert	1.94	35.58	0.02	0.25	0.1	1	2.01	40.97	0.04	0.28	0.14	1	0.07	5.39	0.02	0.03	0.04	0
09-DynPertStr	2	39.14	0.04	0.34	0.15	1	2.04	42.09	0.05	0.25	0.14	1	0.04	2.95	0.01	0.09	0.01	0
10-PNoise05px	1.94	32.77	0.02	0.28	0.13	1	1.92	34.04	0.04	0.2	0.14	1	0.02	1.27	0.02	0.08	0.01	0
11-PNoise25px	1.92	32.74	0.03	0.38	0.12	1	1.93	35.08	0.04	0.29	0.15	1	0.01	2.34	0.01	0.09	0.03	0
12-PNoise50px	1.93	32.54	0.03	0.5	0.13	1	1.92	33.55	0.04	0.36	0.14	1	0.01	1.01	0.01	0.14	0.01	0
13-CombPert	2	38.61	0.03	0.33	0.13	1	2.08	45.59	0.04	0.33	0.13	1	0.08	**6.98**	0.01	0	0	0
14-CombPertStr	**2.35**	**57.07**	0.04	**1.04**	0.17	1	**2.25**	**54.91**	0.07	**0.63**	0.19	1	**0.10**	2.16	0.03	**0.41**	0.02	0

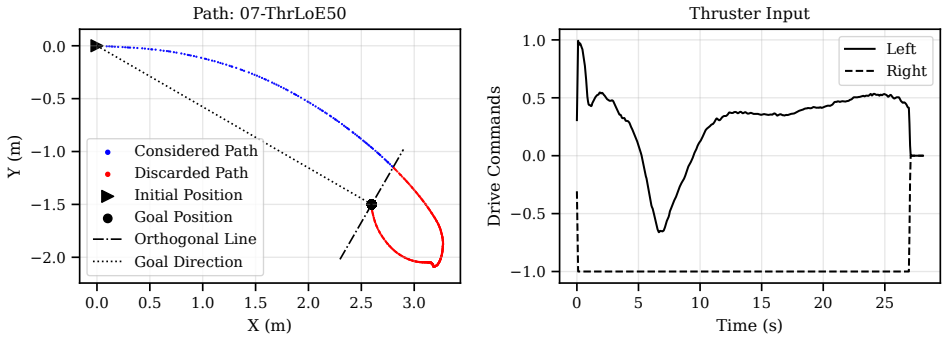


Figure 7.10: Trajectory and actuator commands showing a missed first approach and recovery under *07-ThrLoE50*. Sustained saturation of the right-thruster command at -1 limits clockwise rotation authority, causing the initial miss; the vehicle then reorients and finally reaches the target. For metrics computation, only samples up to the first crossing of the goal-orthogonal line are considered.

to 55, and Ψ_{ER} increases to 0.63. Despite these effects, the success rate remains at 1.0, and the trajectories in Figure 7.8 still converge reliably to the goal.

Sim-to-real gap analysis The sim-to-real gap (Table 7.5) is remarkably small in most conditions, typically ≤ 0.05 for Ψ_{NT} and $\leq 0.02\,\text{m}$ for Ψ_{FD}. The largest discrepancies occur in the dynamic perturbation cases (*08–09*), where Ψ_{NE} increases by more than 5 between simulation and field, reflecting the unmodeled hydrodynamic drag and center-of-mass shifts introduced by physical payloads. Nonetheless, the consistently low mean terminal error ($\approx 0.04\,\text{m}$ across field tests) validates the training setup and the effectiveness of domain randomization for zero-shot sim-to-real transfer.

Main insights Overall, the dominant vulnerability is *actuator dynamics*: omitting the MCU's rate limiting during training induces large heading oscillations, substantially increases effort, and reduces first-pass success. With measured rate limiting and basic domain randomization,

the lightweight PPO policy transfers zero-shot, maintaining a near-perfect success rate and centimeter-level final distance; degradation appears primarily as an efficiency cost (higher energy and longer time), especially under added mass/drag. Observation noise in *goal detection* has a limited effect on accuracy and success, producing at most small increases in excess rotation. The only first-pass miss occurs under severe right-thruster loss-of-effectiveness that pushes the system beyond practical control limits (*07-ThrLoE50*). Sim-to-real gaps remain small for time, distance, and success, with the mild discrepancies in normalized energy under dynamic-perturbation loads. For *point-goal ASV navigation*, simulator fidelity should prioritize accurate actuation modeling and, secondarily, hydrodynamic effects; additional realism in goal-detection noise offers diminishing returns.

7.6 Integration

In this chapter, we present an integrated perception-and-control framework for detecting floating waste and autonomously navigating the ASV to collect it. Our focus is on a system-level evaluation on how the perception and an RL-based controller operate in conjunction. We state the problem, formalize the interfaces, and describe the integration, including a simple goal selection policy that closes the loop between the perception and control modules. The details are provided for clarity and reproducibility, but real applications are encouraged to use more comprehensive approaches. Finally, we present and discuss matched simulation and field-trial results that validate the joint system performance.

7.6.1 Problem Formulation

The goal is to collect all floating bottles within a known bounded planar region using a single ASV. The ASV moves on a planar surface with pose $x_t := (p_t, \theta_t) \in \mathbb{R}^2 \times S^1$. At time t, two finite sets of 2-D points are available, both defined in the global coordinate frame: the remaining search waypoints $\mathcal{W}_t \subseteq \mathbb{R}^2$ and the current perception detections $\mathcal{Z}_t \subseteq \mathbb{R}^2$. The desired behavior is that the ASV alternates between (i) *Target Search*: traverse the waypoint set to cover the area; and (ii) *Point-Goal Navigation*: when detections are available ($\mathcal{Z}_t \neq \emptyset$), temporarily preempt the search to navigate to a selected detection. After visiting a goal position, resume the search.

To achieve this behavior, we need to design an online goal-selection policy μ that, given $(x_t, \mathcal{W}_t, \mathcal{Z}_t)$, produces the next goal $g_t = \mu(x_t, \mathcal{Z}_t, \mathcal{W}_t) \in \mathcal{W}_t \cup \mathcal{Z}_t$.

7.6.2 Methodology

We implement an integrated perception–control pipeline composed of three core modules: (i) *Perception*, which detects floating waste and outputs detections $\mathcal{Z}_t$ (2-D points on the water plane); (ii) *Control*, which tracks a supplied point goal g_t using a learned RL policy; and (iii) *Integration*, which arbitrates between the remaining coverage waypoints $\mathcal{W}_t$ and the current detections $\mathcal{Z}_t$ to produce the next point goal $g_t \in \mathcal{W}_t \cup \mathcal{Z}_t$. The integration layer operates as a lightweight goal publisher that outputs a single selected point goal.

Perception: The perception stack uses either the object detection model with real images or the simulated pipeline implementation described in Section 7.4. The detections on the image plane are projected onto the water plane to yield goal candidates $z \in \mathbb{R}^2$. A short-lived set $\mathcal{Z}_t$ of valid detections is maintained via timeout.

Control: Low-level actuation is handled by the RL-based controller described in Section 7.5. Given the current state observation and a goal $g_t \in \mathbb{R}^2$, the policy controls the ASV by outputting

Algorithm 1 Integration Module for Online Goal Selection

Require: $\mathcal{W} = (w_1, \ldots, w_N)$ (waypoints); r_z (capture distance threshold); r_w (waypoint visit threshold); r_d (detour distance threshold); $\tau_{\max}$ (expiration threshold)

$k \leftarrow 1$ {index of next coverage waypoint}
$\mathcal{Z} \leftarrow \emptyset$ {active detections with timestamps t_z}
$N \leftarrow |\mathcal{W}|$ {number of waypoints}
while true **do**
  $p \leftarrow$ current ASV position
  $t \leftarrow$ current time
  $\mathcal{Z} \leftarrow$ latest detections
  for all $z \in \mathcal{Z}$ **do**
    if $\|p - z\| \leq r_z$ **then**
      remove z from $\mathcal{Z}$
    end if
  end for
  if $\|p - w_k\| \leq r_w$ **then**
    $k \leftarrow (k + 1) \bmod N$
  end if
  $\mathcal{E} \leftarrow \{\, z \in \mathcal{Z} \mid \|z - w_k\| \leq r_d \,\}$
  if $\mathcal{E} \neq \emptyset$ **then**
    $g \leftarrow \arg\min_{z \in \mathcal{E}} \|p - z\|$ {nearest}
  else
    $g \leftarrow w_k$
  end if
  publish g
end while

drive commands for each thruster.

Integration: The integration module arbitrates between coverage waypoints and online detections and, at each control cycle, publishes a single point goal for the controller. At time t, let k_t be the index of the next coverage waypoint and $\mathcal{W}_t = (w_{k_t}, \ldots, w_N)$. Let $\mathcal{Z}_t$ be the set of *active* detections (2-D water-plane points) with timestamps t_z such that $t - t_z \leq \tau_{\max}$. A detection is removed upon collection when $\|p_t - z\| \leq r_z$, and the current waypoint is marked visited (advancing the index) when $\|p_t - w_{k_t}\| \leq r_w$. To keep detours bounded, preemption is restricted to the *eligible* detections.

$$\mathcal{E}_t \triangleq \{\, z \in \mathcal{Z}_t \;:\; \|z - w_{k_t}\| \leq r_d \,\}.$$

The module then publishes one goal per cycle:

$$g_t = \begin{cases} \arg\min\limits_{z \in \mathcal{E}_t} \|p_t - z\|, & \text{if } \mathcal{E}_t \neq \emptyset, \\ w_{k_t}, & \text{otherwise.} \end{cases}$$

In summary, the integration module pursues the nearest eligible detection when one is available; otherwise, continue along the waypoint tour. The integration module is purely reactive and memoryless. Navigation towards the goal is handled by the RL controller that will steer towards the desired position until a new goal is received. Complete logic is described in Algorithm 1.

7.6.3 Experiments

Experiment Description

We evaluated the integrated system with three progressively realistic experiments that first isolate control from perception and later close the loop with real detections. All simulations performed in Gazebo did not include wind and current disturbances. Two obstacle-free rectangular test areas were used in both simulation and field: *Small* ($5 \times 10\,\mathrm{m}$) and *Large* ($15 \times 30\,\mathrm{m}$). Each area was discretized by a lawnmower waypoint pattern with 5 m spacing (yielding 6 and 24 waypoints, respectively). Field trials were conducted on the Kingfisher ASV using the same waypoint sets.

***E1-Sim100* (Large-Scale Simulated Goals).** This experiment stress-tests the arbitration between coverage waypoints and detected targets while abstracting perception. A large set of 100 point targets is sampled randomly within the boundaries of the *Large* test area. The agent must interleave the coverage waypoints with target interception. This experiment is executed solely in simulation to probe the feasibility of the arbitration logic under controlled, disturbance-free dynamics.

***E2-ProgGoals* (Programmed Goals).** This experiment evaluates sim-to-real consistency while still abstracting perception. A fixed set of target goals is executed once in Gazebo and twice in the field on both *Small* and *Large* areas, using identical waypoint sequences and fixed goal locations. The objective is to test coherence under real actuation, localization noise, and platform dynamics.

***E3-Real-Perception* (Bottles-in-the-Loop).** This experiment validates end-to-end performance under real noisy observations. For both *Small* and *Large* areas, up to 5 plastic bottles are simultaneously deployed at diverse locations per run. The full perception pipeline provides target positions used by the RL-based controller, while natural drift caused by wind and water current can generate small target displacements. Bottles are placed manually; after the ASV captures all bottles, a remote-control (RC) takeover is used to approach shore and unload, after which autonomous operation resumes and bottles are eventually placed back in the test area until the experiment is finished.

No artificial disturbances are added in simulation. In the field tests, the ASV is exposed to natural weather conditions. To quantify the wind and water current disturbances, we recorded the RTK IMU measurements while letting the ASV drift without any actuation for over 7 minutes. The ASV reached angular speeds between -0.15 to 0.08 rad/s, surge speeds from -0.14 to 0.29 m/s, and sway speeds in the range of -0.15 to 0.06 m/s. Such disturbances are not negligible considering the ASV's operational velocity of 0.5 m/s. Additionally, the ASV is also exposed to payload variations when bottles are trapped in the net. Missions are started and stopped manually to accommodate bottle placement and unloading. RC takeover is used for unloading or when reaching beyond the predefined region is necessary.

Results and Discussion

In this section, we report experiment data for the three experiments : ***E1-Sim100***, ***E2-ProgGoals***, and ***E3-Real-Perception***, followed by a discussion.

***E1-Sim100* (Large-Scale Simulated Goals).** Figure 7.11 shows the trajectory for 100 uniformly sampled goals placed over the *Large* area. The agent interleaves the coverage of the waypoints with detected goals and captures all targets in 18 minutes. While the alternation policy is naive, these trials demonstrate that the search-and-capture loop is feasible with stable convergence under disturbance-free dynamics. While efficiency and path optimization are left

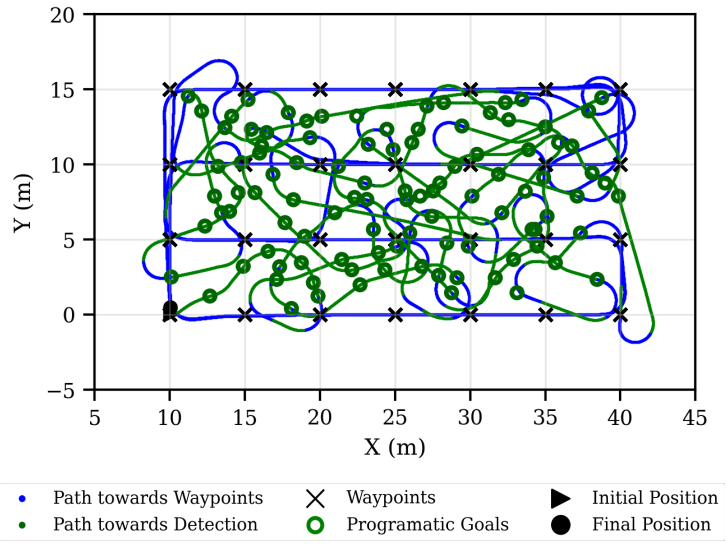


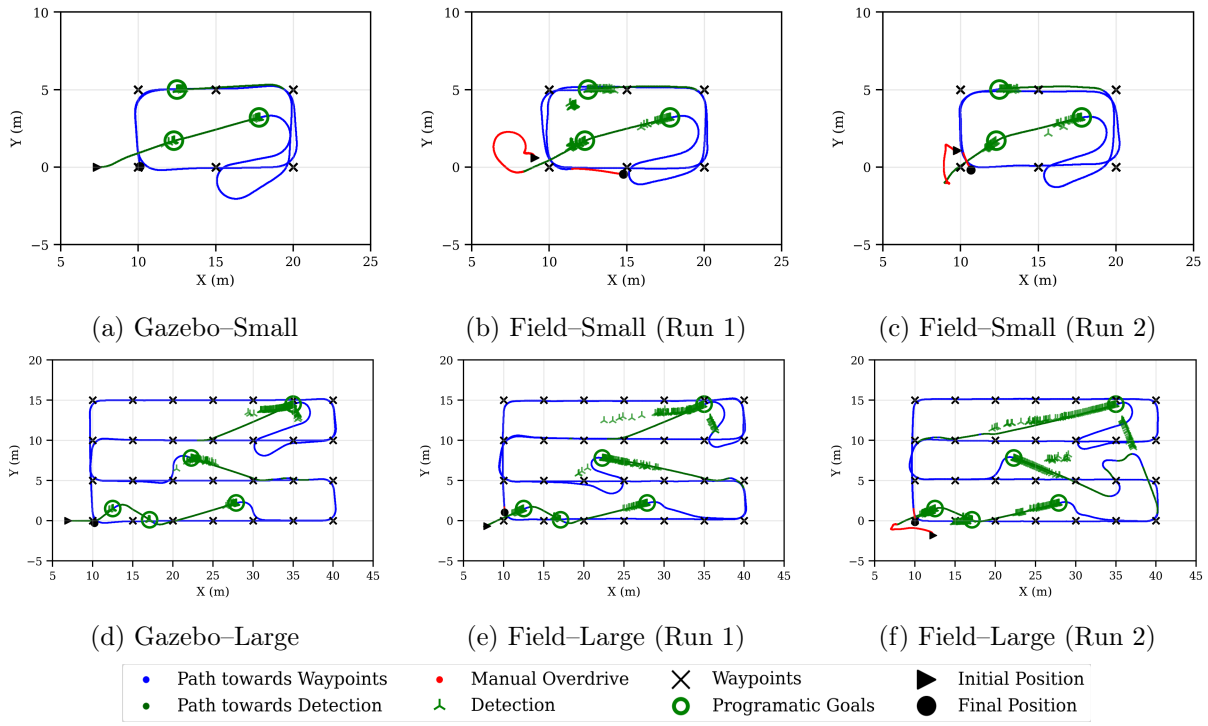
Figure 7.11: *E1-Sim100*: Trajectory on the *Large* area with 100 random goals in simulation. All simulated goals are captured, indicating the feasibility of the proposed goal selection policy.

for improved arbitration in future work this experiment also highlights the value of simulation tests that allows quick large-scale validation that are challenging to repeat in field trials.

***E2-ProgGoals-S2R* (Programmed Goals).** Figure 7.12 compares programmed-goal executions across *Small* and *Large* areas in simulation and in two field repetitions. As a reminder, the programmed-goal experiments use the simulated perception, which projects detections using the calibrated camera parameters (see Section 7.4.2) and is sensitive to pitch-induced projection errors. On the *Small* area, we observe trajectories that are largely similar in all three executions and trajectory deviations arise primarily from initial pose differences. It is possible to observe short detection "trails" near each programmed goal, indicating the projection errors. On the *Large* area, these projection errors become more pronounced. Detections can occasionally enter the acceptance radius, triggering transient goal switches until the integration logic reverts to the waypoints once the distance threshold (r_d) rejects the target. This behavior is particularly visible in the second field repetition for the *Large* area (Figure 7.12), showing that pitch variation leads to alternative goal selections, producing different trajectories. Despite these transients, all programmed goals are eventually reached in both Gazebo and field trials.

***E3-Real-Perception* (Bottles-in-the-Loop).** We conducted 5 field trials (2 *Small*, 3 *Large*), placing a total of 40 bottles. Only two manual interventions were required when wind-driven drift pushed a bottle outside the predefined area. Figure 7.13 illustrates two representative runs. In the *Small* area, the ASV captured five bottles before unloading via manual override. In the *Large* area, it captured 9 bottles autonomously, with a single manual intervention for an out-of-bounds target. The approach to the shore for unloading collected bottles is also visible near the origin coordinate. These results indicate that the perception-to-control loop remains functional using a camera-based perception under real-world conditions.

Perception Diagnostics. Figure 7.14 illustrates typical visual conditions encountered by our onboard detector during autonomous surface collection. The first row (**a**) shows early "search"



(a) Gazebo–Small (b) Field–Small (Run 1) (c) Field–Small (Run 2)

(d) Gazebo–Large (e) Field–Large (Run 1) (f) Field–Large (Run 2)

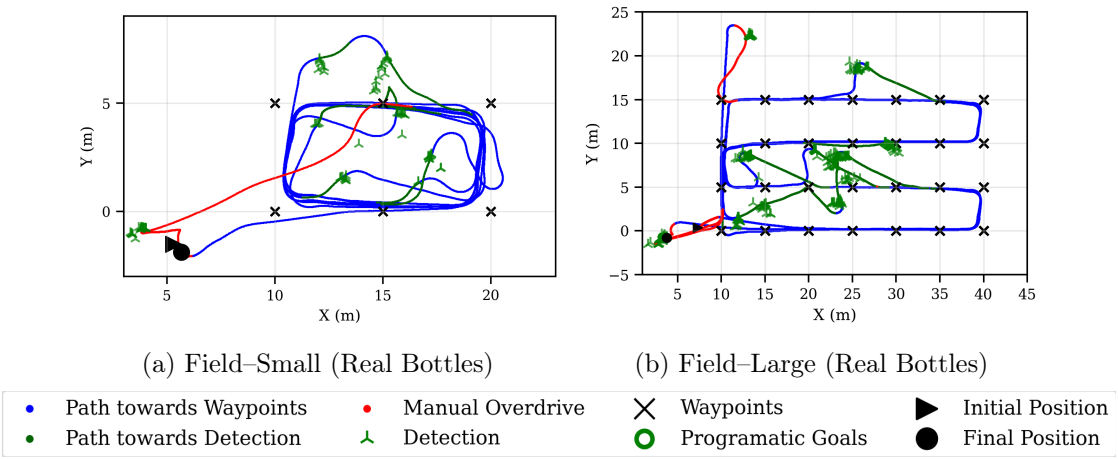
Figure 7.12: *E2-ProgGoals*: Programmed-goal trajectories across simulation and two field runs. In the *Small* area, trajectories align closely; in the *Large* area, increased projection error leads to transient goal switches and different trajectories.

moments in which floating bottles are still distant and occupy only a few pixels. After the vehicle closes in, the second row (**b**) shows the "approach" phase, with targets steered toward the camera's principal axis just prior to capture. The third row (**c**) depicts multi-target scenes, a challenging scenario for the memory-less integration module, where going after one bottle may cause others to exit the field of view. Finally, the fourth row (**d**) presents missed detections most often associated with long range, low contrast, or sensor saturation due to sun glare.

To visualize perception distribution, Figure 7.15 aggregates 805 detections in the ASV coordinate frame. Detections concentrate near the camera's principal axis with a maximum observed range of $\sim 14\,\mathrm{m}$. This spatial distribution is consistent with the expected geometry of the field of view.

Failure Modes and Limitations. The observed limitations primarily stem from the simple, memoryless integration policy. Two recurring challenges are identified: (i) concurrent detections and a policy that prioritizes only the most recent and closest detection can lead to goal oscillations when multiple objects are within the camera's field of view; and (ii) targets detected near the field-of-view boundary may drop out before capture. The memoryless integration makes recovery inefficient: the ASV may loop through remaining waypoints and restart until a missed target re-enters the field of view.

Main insights. Across simulation and field trials, integrating camera-based detections with an RL point-goal controller produced reliable capture behavior, and the perception abstraction enabled identical experiments in simulation and the field, isolating perception integration effects.



(a) Field–Small (Real Bottles) (b) Field–Large (Real Bottles)

Figure 7.13: *E3-Real-Perception*: Representative field trajectories with camera-based perception in the loop. *Small*: five bottles collected, then unloading near the origin. *Large*: nine bottles collected autonomously with one intervention for an out-of-bounds target.

Together, these experiments validate the control-perception integration. The principal limitations stem from the simplified goal selection policy: favoring the newest or closest detection can induce goal oscillations under concurrent targets and near FoV edges, with impacts increasing under larger search areas, higher target density, and restricted FoV. Small calibration biases (e.g., camera pitch) introduce projection offsets that can perturb goal selection and trajectories, yet their effect on the trained controller was negligible within the tested regimes. These findings suggest that future work should prioritize improvements in the planning layer over changes to the controller itself.

7.7 Discussion

7.7.1 Limitations and Future Work

One significant shortcoming of our RL-based control module is the absence of explicit obstacle avoidance within the learned policy. While we already started exploring obstacle avoidance in other preliminary studies, these policies remain untested in field experiments and will be pursued in future work.

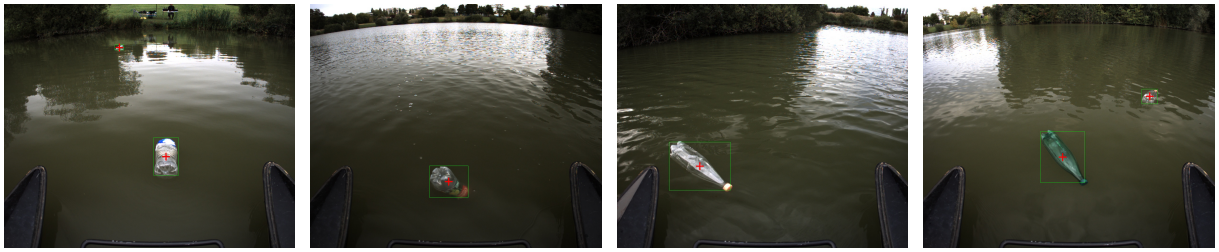
One of the main limitations observed during integration was the simplistic strategy for managing multiple targets. While this design served the purpose of evaluating the control under perception uncertainty, it occasionally led the goal selection to prioritize a single target and then 'forgetting' previously observed targets as they disappeared from the field of view of the camera. Including improved goal management and long-term tracking must be prioritized in future work.

Such target management could also benefit from reduced projection errors. While these projection errors did not significantly impact the control policy, large range variations complicate the association and tracking of detections. A practical next step is to integrate inertial measurements to compensate for pitch variations and to enable online recalibration using the distribution of detections during field tests.

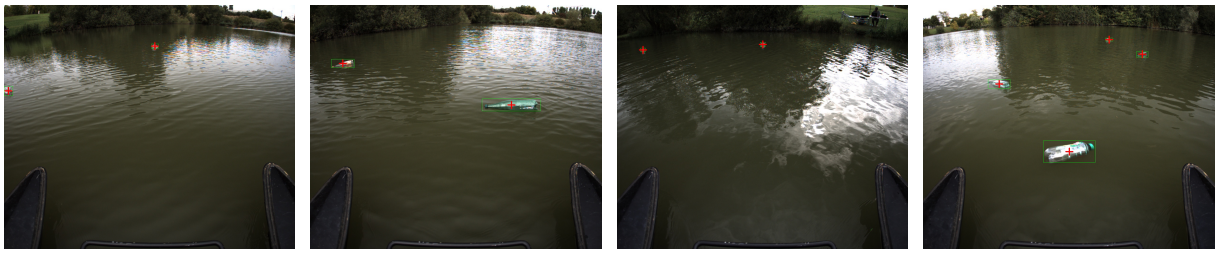
While its impact on our experiments was limited, the perception pipeline would benefit from a stronger detection capability. Fine-tuning the model on newly captured data could provide more generalizable detection models with improved performance. In the scope of this evaluation,



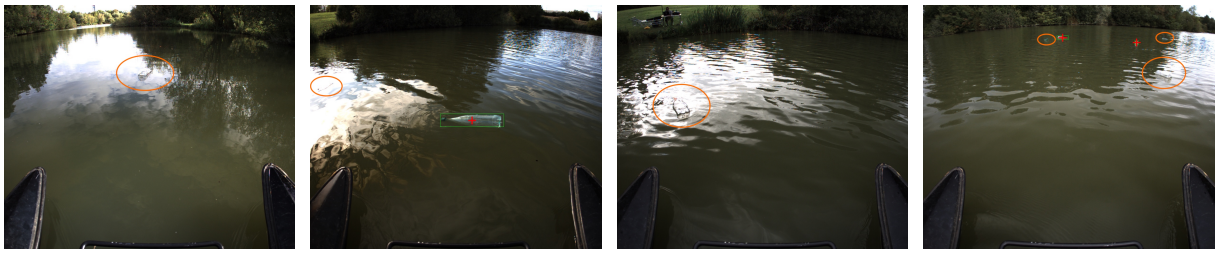
(a) Distant targets at initial detection.



(b) Approach phase showing targets near the image center just before capture.



(c) Multi-target scenes; following one bottle risks losing others.



(d) Missed detections (highlighted in orange) from saturation or low-contrast/long-range targets.

Figure 7.14: Representative detection scenarios for floating bottles during autonomous collection.

we trained models restricted to plastic bottles in the dataset. This facilitated validation of the proposed perception and control integration, but is insufficient for real cleanup tasks, where floating debris types are considerably more diverse.

7.7.2 Lessons Learned

This section details practical insights from our deployments. For each topic, we outline key observations, the actions taken, and our approach to addressing specific challenges.

Sim-to-Real Transferability

Reliable transfer from simulation to the field is nontrivial because dynamics, sensing latency, and environmental disturbances rarely match their simulated counterparts. We obtained better transfer by balancing three aspects: (i) increasing simulation fidelity where the policy is domi-

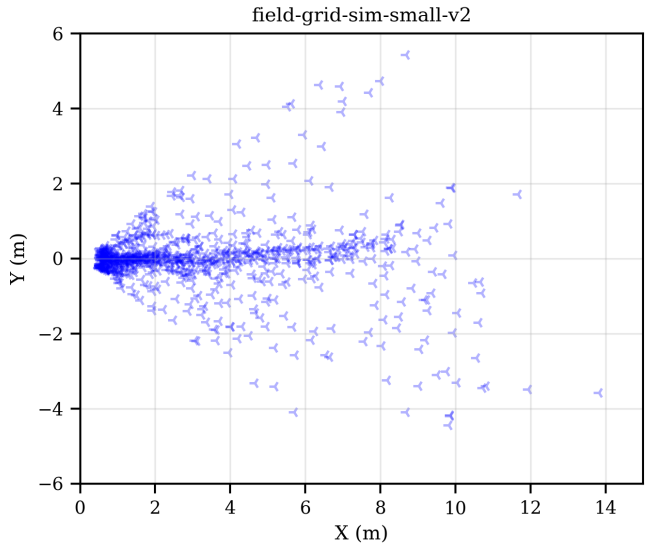


Figure 7.15: Aggregate distribution of bottle detections in the ASV frame. (Total of 805 detections).

nantly sensitive, (ii) applying mild domain randomization on nuisance factors, and (iii) running iterative field trials, isolating components for validation to identify gaps and prioritize based on impact. The fidelity requirement for zero-shot transfer is task dependent. In our case, improving actuation dynamics modeling contributed more to policy robustness than refining observation noise. This prioritization should not be overgeneralized; however, the workflow of isolating and ranking error sources by observed policy sensitivity can be consistently helpful across different domains.

Thruster Modeling

Thruster mismodeling produced policies that were either ineffective or overly conservative. To mitigate this issue, understanding low level actuation signals was critical. In the case of ASVs, separating the effects of thruster dynamics from hull hydrodynamics solely based on field data can be difficult. It makes system identification particularly challenging, especially when having a proprietary microcontroller in the control loop. In our case, the efficient solution was to probe the PPM signal delivered to the motor controller while varying input commands sent to the MCU. This allowed us to estimate the embedded control behavior. Incorporating the observed rate limit into the simulator substantially improved the quality of our actuation model and the performance of the trained policies.

Simulation Environment

We found it invaluable to maintain a secondary simulation environment (Gazebo) with independent physics implementation from the learning environment (Isaac Sim). This cross-validation of trained policies in a second simulation environment helped expose policy transfer limitations early, allowing fast iteration and supporting systems integration checks. It also simplified tasks such as ROS integration and was particularly useful in developing an integration module, which reduced the need for time-consuming field tests and facilitated solving implementation issues

such as frame transforms threshold tuning.

Perception Abstraction

Evaluating a full end-to-end stack complicates error attribution. We introduced a simulated perception abstraction that reuses the calibrated camera model to project water-plane targets. As an intermediary step for modular testing, this enabled fair sim-to-real comparisons and systematic debugging, including during field trials. This was particularly useful to isolate how projection errors influence goal selection before testing with non-reproducible camera detections that lack ground truth target position.

Policy Training

Reward design for reinforcement learning is delicate; inappropriate scales or competing terms can stall learning or create unintended behaviors. Goal progress rewards, such as approaching and facing the target, were scaled to dominate total reward so the agent learned the task quickly. Secondary shaping terms, such as reducing energy usage, were introduced as penalties and kept an order of magnitude smaller to shape behavior without dominating the combined reward. Without this separation, training often failed to converge or converged to undesired behaviors.

Time Synchronization

System implementation as multiple independent modules (perception, localization, control) introduces timing errors that can masquerade as controller faults. Latency must be treated carefully with attention to synchronized clocks, message timestamps, and frame updates. In our system, perception produced goals at a lower rate and with non-negligible latency. Storing detections in the global frame and re-projecting to the local frame at the control rate using the latest pose reduced temporal inconsistency and removed the need for a dedicated target tracker.

Perception in Aquatic Environments

Camera-based perception in outdoor water surface is disrupted by specular reflections, sensor saturation, and rapidly changing illumination, which can create temporarily blind regions in the image plane. We used a polarimetric camera that suppressed reflections where the sensor was not saturated, which can improve detection precision, but can't recover from saturated pixels. Since our controller experiments indicated strong robustness against position error in object detection, we consider that a sensor with higher dynamic range or improved auto exposure should be prioritized.

Field Logistics

Outdoor testing is constrained by weather, battery capacity, limited displays, and access to shore infrastructure, which increases the difficulty of fixing issues in the field. To mitigate those issues, we prioritized automating tests, using configuration files, and implementing modules that could be reused directly in both simulation and field tests. This approach was especially important for the ablation study with repetitive runs, allowing for validating the full pipeline in simulation before, reducing integration surprises, and increasing productivity during field trials.

7.8 Conclusion

This chapter presented a field-validated perception-to-control architecture for autonomous surface vehicles tasked with detecting and collecting floating debris. A polarimetric perception pipeline converts detections into water-plane point goals, and a lightweight policy trained in simulation executes differential-thrust motion control. Field tests across disturbance regimes, including localization delay, added drag or payload, perception noise, and thruster loss of effectiveness, yield a high success rate and terminal accuracy under natural wind and current. Zero-shot transfer and field robustness were enabled by a careful evaluation protocol coupled with component-level abstraction, allowing matching simulation and field experiments that exposed the dominant error sources and concentrated modeling effort where it most affected the transfer. The dominant determinant of transfer performance is actuation fidelity: if the onboard microcontroller's rate limiting is omitted during training, the vessel exhibits heading oscillations, longer completion times, and higher energy usage. In contrast, substantial image-plane detection noise has minimal impact on the observed metrics. Further integrated search-and-capture missions with real detections confirmed end-to-end viability, and remaining failure modes arise from memoryless single-target goal arbitration and limited field-of-view. Future work should include obstacle avoidance, add memory-aware multi-target management, and deploy IMU-assisted online pitch compensation, which should further harden performance for real-world applications.

8

Conclusion

This dissertation investigated how polarimetric perception and reinforcement learning can be integrated to enable autonomy in surface vessels tasked with detecting and retrieving floating debris. This work advanced different layers of the autonomy stack, including sensing, control, and end-to-end deployment. The approach was validated in both simulation and field trials, using a rigorous evaluation protocol to assess the robustness of the trained policy.

8.1 Contributions

8.1.1 Perception

Our experimental results indicate that polarimetric cues, particularly the degree of linear polarization (DoLP) and the angle of linear polarization (AoLP), complement RGB images for detecting and segmenting floating waste in reflective outdoor scenes. In regions with strong polarization, integrating these channels increases mean IoU and reduces contour error by suppressing glare and enhancing target-to-background contrast. To support reproducibility and foster research, we released a polarimetric dataset for this application together with code and baselines that include detection, segmentation, and fusion models. The study also identified limitations. Gains diminish when polarization is weak; division-of-focal-plane sensors produce large superpixels that reduce spatial resolution and introduce artifacts; and existing fusion techniques can produce false positives when background materials exhibit similar polarization signatures. The main takeaway is that polarimetry is particularly valuable when precision matters and in polarization-rich scenes. Current limitations motivate larger datasets and more efficient fusion models.

8.1.2 Reinforcement learning-based control

This thesis contributes to learning frameworks by extending functionality for mobile robots and by incorporating buoyancy and hydrodynamic effects when training policies for surface vessels. Despite the use of simplified dynamics in simulation, we achieve zero-shot sim-to-real transfer of policies learned entirely in simulation. This is achieved by combining system identification with domain randomization to address model uncertainty. While care is required to avoid large actuation mis-modelling, reinforcement learning provides a practical alternative to hand-engineered controllers in dynamic environments. The main advantages include lightweight policies that reliably reach targets while minimizing energy consumption. Future development should prioritize increasing the complexity of the tasks, such as incorporating obstacle avoidance, and integrating

techniques that improve the temporal reasoning (e.g., RNNs/LSTMs) of the trained policy, such as recurrent neural networks or LSTM modules, enabling the policy to learn disturbances and adapt behavior accordingly.

8.1.3 Integration and field testing

We implemented a modular integration pipeline that combines perception and reinforcement learning–based navigation into an end-to-end application. The thesis details an evaluation protocol that combines simulation and field tests across progressive levels of abstraction, serving two purposes: identifying the sim-to-real gap to prioritize improvements in simulation fidelity, and evaluating policy robustness against different types of disturbances. Through this methodology, we achieved high success rates with similar outcomes in simulation and field trials. This enabled zero-shot transfer of policies trained entirely in simulation, achieving high success rates and accurate terminal positioning, including scenarios with localization delay, added drag or payload, perception noise, and partial thruster loss of effectiveness. While the control policy is robust, remaining failures arose from memoryless goal arbitration and a limited field of view.

8.2 Future Work

Task-Aware Demosaicking. Division-of-focal-plane sampling reduces effective spatial resolution and forces demosaicking that introduces aliasing, angle misregistration, and cross-talk, which in turn degrades downstream perception. A promising direction is task-aware demosaicking, where the reconstruction model is trained jointly with the target task, such as semantic segmentation or detection. Coupling the demosaicker with denoise-and-demosaic supervision could further limit artifacts while keeping the parameter count low. For maritime scenarios, the capture pipeline should also prioritize high dynamic range acquisition to prevent saturation from specular glint, ensuring that polarization information remains usable in high-contrast conditions.

Robust, Efficient Polarimetric Fusion. Current fusion pipelines explored for polarimetric images often fail to generalize across scenes and weather conditions, and many rely on heavy backbones that are impractical for embedded platforms. Future work should target robustness and efficiency by designing lightweight fusion modules that respect polarization constraints. Physics-informed heads that operate in Stokes and Mueller representations can enforce valid polarization states and stabilize learning, which may improve out-of-distribution behavior while reducing compute. Progress will also depend on better data. Polarimetrically-correct simulators could be integrated into photorealistic engines such as Isaac Sim, with native support for Stokes and Mueller rendering, so that large, diverse, and well-labeled synthetic datasets can be generated for training and stress-testing. Such simulation capabilities would close current dataset gaps and accelerate research on practical, generalizable fusion for a wide range of robotic applications.

Simulation and Sim-to-Real. Current simulators often miss critical maritime phenomena and realistic sensor characteristics. Research should advance high-fidelity and high-throughput simulation for ASVs. Priority areas include fast physics engines, improved hydrodynamics, and more detailed wave, current, and wind interactions. Reducing and characterizing the sim-to-real gap serves two goals: (i) improve transferability of learned policies to field deployments and (ii) enable large-scale, low-cost evaluation of robustness. When fidelity is sufficient, policies can be

stress-tested in simulation before field trials; future work should consider appropriate sim-to-real gap metrics, calibrate sensor and actuator models against field data, and validate transfer protocols with controlled ablations.

Safety and Energy Efficiency. Safety and energy efficiency remain important field constraints. Recent research has been investigating how to integrate control barrier function (CBF) filters and runtime assurance supervisors that monitor policy actions and assess uncertainty; this is promising and can benefit from on-robot validation over long horizons. In parallel, energy efficiency should continue to be prioritized. While RL can incorporate energy-related rewards and promote efficient operation, future research should continue to pursue lightweight, robust policies that achieve human-like frugality, avoiding complex computation cycles at run time. Future research could explore how to (i) dynamically adjust the energy budget to task requirements (e.g., finish quickly or save energy) and (ii) distill policies for lightweight, on-board deployment.

Multi-Task Policies. While the formulation in this thesis targets a single task, practical deployment requires broader coverage beyond waste interception, including docking, velocity tracking, and exploration. Future research should evaluate multi-task formulations for ASVs by considering (i) context-conditioned policies that infer the task from observations or compact task embeddings, (ii) hierarchical skill libraries with a planner selecting among option policies, and (iii) parameter-sharing strategies that reuse common learned representations for different tasks. The learning framework outlined in Chapter 4 presents an effort to support this trajectory.

Generalist vs. Task-Specific Policies. There has been rapid growth in reinforcement learning, including vision–language–action policies, with promise for generalist and adaptable autonomy, yet many maritime tasks on ASVs can be narrow in scope and safety-critical. Predictable, testable behavior under tight operational constraints is therefore essential. We employ a layered architecture that separates perception from control with an explicit, well-defined interface, enabling sensor interchangeability, systematic unit tests, and interpretability. Nonetheless, future work should closely monitor progress in generalist methods and evaluate potential paradigm shifts grounded in task criticality, data availability, verification requirements, and maintainability.

Operational Complexity. While this thesis focuses on the capture task, it represents only one component of operational autonomy. Real deployments are substantially more complex and demand more elaborate solutions across multiple areas. Here, we outline a few research directions and the problems they target. Addressing underwater obstacles or low-visibility conditions motivates multi-sensor fusion, including forward-looking sonar, radar, and near-field LiDAR. Meeting diverse mission requirements benefits from a layered reasoning framework that orchestrates reusable skills (e.g., docking, interception, and survey operations). Finally, scaling operations might require multi-robot teams that share perception, coordinate tasks, and remain resilient to intermittent communications. Continued research across all of these directions is essential to move from isolated capabilities to robust, scalable autonomy.

8.3 Concluding Remarks

This thesis spanned perception, control, and integration, with deliberate emphasis on field trials. Getting a complete system to work reliably was challenging but consistently rewarding. Despite limited incentives for deployment-focused research, these efforts yielded concrete lessons

on effective test protocols, controller behavior under real disturbances, sensor limitations, time synchronization, and sensor calibration. Integrating perception, control, and integration components in the field exposed failure modes that remained invisible in simulation, and iterative deployments were essential for tracing root causes and implementing durable fixes. The results reinforce that progress in autonomous systems depends as much on careful system design and disciplined integration as on algorithmic novelty. Field-driven evaluation should therefore stand alongside novelty as a primary criterion for academic significance, because it yields generalizable evidence, strengthens causal claims, and improves reproducibility.

Bibliography

[1] A. Kirillov et al. "Segment anything". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2023, pp. 4015–4026.

[2] L. K. Chung et al. "Guaranteed Reach-Avoid for Black-Box Systems through Narrow Gaps via Neural Network Reachability". In: *2025 IEEE International Conference on Robotics and Automation (ICRA)*. 2025, pp. 3574–3581. DOI: 10.1109/ICRA55743.2025.11127579.

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

[4] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye. "Object detection in 20 years: A survey". In: *Proceedings of the IEEE* (2023).

[5] Y. Cheng et al. "Flow: A dataset and benchmark for floating waste detection in inland waters". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 10953–10962.

[6] Y. Qiao, J. Yin, W. Wang, F. Duarte, J. Yang, and C. Ratti. "Survey of deep learning for autonomous surface vehicles in marine environments". In: *IEEE Transactions on Intelligent Transportation Systems* 24.4 (2023), pp. 3678–3701.

[7] Q. Zhang, W. Pan, and V. Reppa. "Model-reference reinforcement learning for collision-free tracking control of autonomous surface vehicles". In: *IEEE Transactions on Intelligent Transportation Systems* 23.7 (2021), pp. 8770–8781.

[8] N. Wang, Y. Gao, H. Zhao, and C. K. Ahn. "Reinforcement learning-based optimal tracking control of an unknown unmanned surface vehicle". In: *IEEE Transactions on Neural Networks and Learning Systems* 32.7 (2020), pp. 3034–3045.

[9] N. Wang, Y. Gao, C. Yang, and X. Zhang. "Reinforcement learning-based finite-time tracking control of an unknown unmanned surface vehicle with input constraints". In: *Neurocomputing* 484 (2022), pp. 26–37.

[10] W. Wang et al. "Roboat III: An autonomous surface vessel for urban transportation". en. In: *Journal of Field Robotics* 40.8 (Dec. 2023), pp. 1996–2009. ISSN: 1556-4959, 1556-4967. DOI: 10.1002/rob.22237. URL: https://onlinelibrary.wiley.com/doi/10.1002/rob.22237 (visited on 07/17/2024).

[11] W. Wang et al. "Deep Reinforcement Learning Based Tracking Control of an Autonomous Surface Vessel in Natural Waters". In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2023, pp. 3109–3115.

[12] L. F. W. Batista, S. Khazem, M. Adibi, S. Hutchinson, and C. Pradalier. "PoTATO: A Dataset for Analyzing Polarimetric Traces of Afloat Trash Objects". In: *Computer Vision – ECCV 2024 Workshops*. Ed. by A. Del Bue, C. Canton, J. Pont-Tuset, and T. Tommasi. Cham: Springer Nature Switzerland, 2025, pp. 190–205. ISBN: 978-3-031-91569-7.

[13] T. van Emmerik and A. Schwarz. "Plastic debris in rivers". In: *Wiley Interdisciplinary Reviews: Water* 7.1 (2020), e1398.

[14] M. MacLeod, H. P. H. Arp, M. B. Tekman, and A. Jahnke. "The global threat from plastic pollution". In: *Science* 373.6550 (2021), pp. 61–65.

[15] A. Chamas et al. "Degradation rates of plastics in the environment". In: *ACS Sustainable Chemistry & Engineering* 8.9 (2020), pp. 3494–3511.

[16] D. V. Politikos, A. Adamopoulou, G. Petasis, and F. Galgani. "Using artificial intelligence to support marine macrolitter research: A content analysis and an online database". In: *Ocean & Coastal Management* 233 (2023), p. 106466.

[17] R. Blin, S. Ainouz, S. Canu, and F. Meriaudeau. "The polarlitis dataset: Road scenes under fog". In: *IEEE Transactions on Intelligent Transportation Systems* 23.8 (2021), pp. 10753–10762.

[18] Y. Wang et al. "Principle and Implementation of Stokes Vector Polarization Imaging Technology". In: *Applied Sciences* 12.13 (2022), p. 6613.

[19] A. G. Andreou and Z. K. Kalayjian. "Polarization imaging: principles and integrated polarimeters". In: *IEEE Sensors journal* 2.6 (2002), pp. 566–576.

[20] J. J. Foster et al. "Polarisation vision: overcoming challenges of working with a property of light we barely see". In: *The Science of Nature* 105 (2018), pp. 1–26.

[21] T. Jia et al. "Deep learning for detecting macroplastic litter in water bodies: a review". In: *Water Research* (2023), p. 119632.

[22] R. Zhang, S. Li, G. Ji, X. Zhao, J. Li, and M. Pan. "Survey on deep learning-based marine object detection". In: *Journal of Advanced Transportation* 2021 (2021), pp. 1–18.

[23] A. Iqbal, M. G. Garcia, L. Chellappan, and N. Gans. "Object detection and classification for small objects in/on water". In: *Journal of Electronic Imaging* 31.3 (2022), pp. 033041–033041.

[24] A. Dave, Y. Zhao, and A. Veeraraghavan. "Pandora: Polarization-aided neural decomposition of radiance". In: *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VII*. Springer. 2022, pp. 538–556.

[25] D. Gao et al. "Polarimetric pose prediction". In: *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IX*. Springer. 2022, pp. 735–752.

[26] C. Lei, X. Huang, M. Zhang, Q. Yan, W. Sun, and Q. Chen. "Polarized reflection removal with perfect alignment in the wild". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 1750–1758.

[27] B. M. Ratliff, D. A. LeMaster, R. T. Mack, P. V. Villeneuve, J. J. Weinheimer, and J. R. Middendorf. "Detection and tracking of RC model aircraft in LWIR microgrid polarimeter data". In: *Polarization Science and Remote Sensing V*. Vol. 8160. SPIE. 2011, pp. 29–41.

[28] R. Blin, S. Ainouz, S. Canu, and F. Meriaudeau. "Adapted learning for polarization-based car detection". In: *Fourteenth International Conference on Quality Control by Artificial Vision*. Vol. 11172. SPIE. 2019, pp. 312–318.

[29] R. Blin, S. Ainouz, S. Canu, and F. Meriaudeau. "Multimodal polarimetric and color fusion for road scene analysis in adverse weather conditions". In: *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2021, pp. 3338–3342.

[30] A. Kalra, V. Taamazyan, S. K. Rao, K. Venkataraman, R. Raskar, and A. Kadambi. "Deep polarization cues for transparent object segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 8602–8611.

[31] R. De Vries, M. Egger, T. Mani, and L. Lebreton. "Quantifying floating plastic debris at sea using vessel-based optical data and artificial intelligence". In: *Remote Sensing* 13.17 (2021), p. 3401.

[32] J. Wang, W. Guo, T. Pan, H. Yu, L. Duan, and W. Yang. "Bottle detection in the wild using low-altitude unmanned aerial vehicles". In: *2018 21st International Conference on Information Fusion (FUSION)*. IEEE. 2018, pp. 439–444.

[33] P. F. Proença and P. Simoes. "Taco: Trash annotations in context for litter detection". In: *arXiv preprint arXiv:2003.06975* (2020).

[34] M. Tkachenko, M. Malyuk, A. Holmanyuk, and N. Liubimov. *Label Studio: Data labeling software*. Open source software available from https://github.com/heartexlabs/label-studio. 2020. URL: `https://github.com/heartexlabs/label-studio`.

[35] D. H. Goldstein. *Polarized light*. CRC press, 2017.

[36] B. M. Ratliff, C. F. LaCasse, and J. S. Tyo. "Interpolation strategies for reducing IFOV artifacts in microgrid polarimeter imagery". In: *Optics express* 17.11 (2009), pp. 9112–9125.

[37] R. Blin, S. Ainouz, S. Canu, and F. Meriaudeau. "A new multimodal RGB and polarimetric image dataset for road scenes analysis". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 2020, pp. 216–217.

[38] T.-Y. Lin et al. *Microsoft COCO: Common Objects in Context*. 2015. arXiv: `1405.0312 [cs.CV]`.

[39] Ultralytics. *ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation*. `https://github.com/ultralytics/yolov5.com`. Accessed: 7th May, 2023. 2022. DOI: `10.5281/zenodo.7347926`. URL: `https://doi.org/10.5281/zenodo.7347926`.

[40] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. *Detectron2*. `https://github.com/facebookresearch/detectron2`. 2019.

[41] K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[42] S. Qiu, Q. Fu, C. Wang, and W. Heidrich. "Linear polarization demosaicking for monochrome and colour polarization focal plane arrays". In: *Computer Graphics Forum*. Vol. 40. Wiley Online Library. 2021, pp. 77–89.

[43] L. F. W. Batista, T. Bourbon, and C. Pradalier. "Evaluation of Polarimetric Fusion for Semantic Segmentation in Aquatic Environments". In: (Sept. 2025). URL: `https://hal.science/hal-05288763`.

[44] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos. "Image Segmentation Using Deep Learning: A Survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.7 (2022), pp. 3523–3542. DOI: `10.1109/TPAMI.2021.3059968`.

[45] M. Nawaz, J. K.-T. Tang, K. Bibi, S. Xiao, H.-P. Ho, and W. Yuan. "Robust Cognitive Capability in Autonomous Driving Using Sensor Fusion Techniques: A Survey". In: *IEEE Transactions on Intelligent Transportation Systems* 25.5 (2024), pp. 3228–3243. DOI: 10.1109/TITS.2023.3327949.

[46] L. Trinh, S. Mercelis, and A. Anwar. "A comprehensive review of datasets and deep learning techniques for vision in unmanned surface vehicles". In: *Ocean Engineering* 334 (2025), p. 121501.

[47] R. Douguet, D. Heller, and J. Laurent. "Multimodal perception for obstacle detection for flying boats - Unmanned Surface Vehicle (USV)". In: *OCEANS 2023 - Limerick*. 2023, pp. 1–8. DOI: 10.1109/OCEANSLimerick52467.2023.10244714.

[48] Y. Cheng, M. Jiang, J. Zhu, and Y. Liu. "Are we ready for unmanned surface vehicles in inland waterways? The usvinland multisensor dataset and benchmark". In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 3964–3970.

[49] X. Li et al. "Polarimetric imaging via deep learning: A review". In: *Remote Sensing* 15.6 (2023), p. 1540.

[50] Y. Liang, R. Wakaki, S. Nobuhara, and K. Nishino. "Multimodal material segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 19800–19808.

[51] J. Zhang, H. Liu, K. Yang, X. Hu, R. Liu, and R. Stiefelhagen. "CMX: Cross-Modal Fusion for RGB-X Semantic Segmentation With Transformers". In: *IEEE Transactions on Intelligent Transportation Systems* 24.12 (2023), pp. 14679–14694. DOI: 10.1109/TITS.2023.3300537.

[52] J. Zhang et al. "Delivering Arbitrary-Modal Semantic Segmentation". In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 1136–1147. DOI: 10.1109/CVPR52729.2023.00116.

[53] M. K. Reza, A. Prater-Bennette, and M. S. Asif. "MMSFormer: Multimodal Transformer for Material and Semantic Segmentation". In: *IEEE Open Journal of Signal Processing* 5 (2024), pp. 599–610. DOI: 10.1109/OJSP.2024.3389812.

[54] B. Li, D. Zhang, Z. Zhao, J. Gao, and X. Li. "Stitchfusion: Weaving any visual modalities to enhance multimodal semantic segmentation". In: *arXiv preprint arXiv:2408.01343* (2024).

[55] O. Ronneberger, P. Fischer, and T. Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer. 2015, pp. 234–241.

[56] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. "Rethinking atrous convolution for semantic image segmentation. arXiv 2017". In: *arXiv preprint arXiv:1706.05587* 2 (2019), p. 1.

[57] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo. "SegFormer: Simple and efficient design for semantic segmentation with transformers". In: *Advances in neural information processing systems* 34 (2021), pp. 12077–12090.

[58] M. El-Hariry et al. "RoboRAN: A Unified Robotics Framework for Reinforcement Learning-Based Autonomous Navigation". In: *Transactions on Machine Learning Research* (2025). ISSN: 2835-8856. URL: https://openreview.net/forum?id=0wDbhLeMj9.

[59] B. Bingham et al. "Toward maritime robotic simulation in gazebo". In: *OCEANS 2019 MTS/IEEE SEATTLE*. IEEE. 2019, pp. 1–10.

[60] M. M. M. Manhães, S. A. Scherer, M. Voss, L. R. Douat, and T. Rauschenbach. "UUV simulator: A gazebo-based package for underwater intervention and multi-robot simulation". In: *OCEANS 2016 MTS/IEEE Monterey*. IEEE. 2016, pp. 1–8.

[61] P. Cieślak. "Stonefish: An advanced open-source simulation tool designed for marine robotics, with a ROS interface". In: *OCEANS 2019-Marseille*. IEEE. 2019, pp. 1–6. DOI: 10.1109/OCEANSE.2019.8867434.

[62] M. El-Hariry, A. Richard, V. Muralidharan, B. C. Yalcin, M. Geist, and M. Olivares-Mendez. "DRIFT: Deep Reinforcement Learning for Intelligent Floating Platforms Trajectories". In: *arXiv preprint arXiv:2310.04266* (2023).

[63] V. Makoviychuk et al. *Isaac gym: High performance gpu-based physics simulation for robot learning*. 2021.

[64] NVIDIA. *Omniverse Isaac Gym Reinforcement Learning Environments for Isaac Sim*. https://github.com/isaac-sim/OmniIsaacGymEnvs. 2024.

[65] M. Mittal et al. "Orbit: A Unified Simulation Framework for Interactive Robot Learning Environments". In: *IEEE Robotics and Automation Letters* 8.6 (2023), pp. 3740–3747. DOI: 10.1109/LRA.2023.3270034.

[66] D. Makoviichuk and V. Makoviychuk. *rl-games: A High-performance Framework for Reinforcement Learning*. https://github.com/Denys88/rl_games. May 2021.

[67] A. Serrano-Muñoz, D. Chrysostomou, S. Bøgh, and N. Arana-Arexolaleiba. "skrl: Modular and Flexible Library for Reinforcement Learning". In: *Journal of Machine Learning Research* 24.254 (2023), pp. 1–9. URL: http://jmlr.org/papers/v24/23-0112.html.

[68] N. Rudin, D. Hoeller, P. Reist, and M. Hutter. "Learning to Walk in Minutes Using Massively Parallel Deep Reinforcement Learning". In: *Proceedings of the 5th Conference on Robot Learning*. Vol. 164. Proceedings of Machine Learning Research. PMLR, 2022, pp. 91–100. URL: https://proceedings.mlr.press/v164/rudin22a.html.

[69] L. F. W. Batista, J. Ro, A. Richard, P. Schroepfer, S. Hutchinson, and C. Pradalier. "A Deep Reinforcement Learning Framework and Methodology for Reducing the Sim-to-Real Gap in ASV Navigation". In: *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2024, pp. 1258–1264. DOI: 10.1109/IROS58592.2024.10802067.

[70] F. Fornai, G. Ferri, A. Manzi, F. Ciuchi, F. Bartaloni, and C. Laschi. "An Autonomous Water Monitoring and Sampling System for Small-Sized ASVs". In: *IEEE Journal of Oceanic Engineering* 42.1 (2017), pp. 5–12. DOI: 10.1109/JOE.2016.2552818.

[71] H.-C. Chang, Y.-L. Hsu, S.-S. Hung, G.-R. Ou, J.-R. Wu, and C. Hsu. "Autonomous water quality monitoring and water surface cleaning for unmanned surface vehicle". In: *Sensors* 21.4 (2021), p. 1102.

[72] X. Zhou, Y. Ge, W. Li, and G. Ye. "Time-constrained multiple unmanned surface vehicles cooperation for sea surface oil pollution cleanup". In: *2021 6th International Conference on Robotics and Automation Engineering (ICRAE)*. IEEE. 2021, pp. 40–45.

[73] M. Lewis, K. Sycara, and P. Walker. "The Role of Trust in Human-Robot Interaction". In: *Foundations of Trusted Autonomy*. Ed. by H. A. Abbass, J. Scholz, and D. J. Reid. Cham: Springer International Publishing, 2018, pp. 135–159. ISBN: 978-3-319-64816-3. DOI: 10.1007/978-3-319-64816-3_8. URL: https://doi.org/10.1007/978-3-319-64816-3_8.

[74] L. Wells and T. Bednarz. "Explainable AI and Reinforcement Learning—A Systematic Review of Current Approaches and Trends". In: *Frontiers in Artificial Intelligence* 4 (2021). ISSN: 2624-8212. DOI: 10.3389/frai.2021.550030. URL: https://www.frontiersin.org/articles/10.3389/frai.2021.550030.

[75] P. Schroepfer and C. Pradalier. "Why There is No Definition of Trust: A Systems Approach With a Metamodel Representation". In: *2023 32nd IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. 2023, pp. 1245–1251. DOI: 10.1109/RO-MAN57019.2023.10309503.

[76] Y. Zhao, X. Qi, Y. Ma, Z. Li, R. Malekian, and M. A. Sotelo. "Path following optimization for an underactuated USV using smoothly-convergent deep reinforcement learning". In: *IEEE Transactions on Intelligent Transportation Systems* 22.10 (2020), pp. 6208–6220.

[77] X. Lin, J. McConnell, and B. Englot. "Robust unmanned surface vehicle navigation with distributional reinforcement learning". In: *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2023, pp. 6185–6191.

[78] Y. Wang, J. Cao, J. Sun, X. Zou, and C. Sun. "Path Following Control for Unmanned Surface Vehicles: A Reinforcement Learning-Based Method With Experimental Validation". In: *IEEE Transactions on Neural Networks and Learning Systems* (2023).

[79] E. I. Sarda, H. Qu, I. R. Bertaska, and K. D. von Ellenrieder. "Station-keeping control of an unmanned surface vehicle exposed to current and wind disturbances". In: *Ocean Engineering* 127 (2016), pp. 305–324.

[80] M. El-Hariry, A. Richard, and M. Olivares-Mendez. "RANS: Highly-Parallelised Simulator for Reinforcement Learning based Autonomous Navigating Spacecrafts". In: *arXiv preprint arXiv:2310.07393* (2023).

[81] A. Richard, S. Aravecchia, T. Schillaci, M. Geist, and C. Pradalier. "How to Train Your HERON". In: *IEEE Robotics and Automation Letters* 6.3 (2021), pp. 5247–5252. DOI: 10.1109/LRA.2021.3065278.

[82] A. Richard, S. Aravecchia, M. Geist, and C. Pradalier. "Learning behaviors through physics-driven latent imagination". In: *Conference on Robot Learning*. PMLR. 2022, pp. 1190–1199.

[83] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. *Proximal Policy Optimization Algorithms*. arXiv:1707.06347 [cs]. Aug. 2017. DOI: 10.48550/arXiv.1707.06347. URL: http://arxiv.org/abs/1707.06347 (visited on 05/12/2025).

[84] L. F. W. Batista, S. Aravecchia, S. Hutchinson, and C. Pradalier. *Evaluating Robustness of Deep Reinforcement Learning for Autonomous Surface Vehicle Control in Field Tests*. 2025.

[85] L. Ljung. "System identification". In: *Signal analysis and prediction*. Springer, 1998, pp. 163–173.

[86] A. Mahé, A. Richard, S. Aravecchia, M. Geist, and C. Pradalier. "Evaluation of Prioritized Deep System Identification on a Path Following Task". en. In: *Journal of Intelligent & Robotic Systems* 101.4 (Apr. 2021), p. 78. ISSN: 0921-0296, 1573-0409. DOI: 10.1007/s10846-021-01341-1. URL: https://link.springer.com/10.1007/s10846-021-01341-1 (visited on 07/03/2024).

[87] M. Bahadorian, B. Savkovic, R. Eaton, and T. Hesketh. "Robust model predictive control for automated trajectory tracking of an unmanned ground vehicle". In: *2012 American Control Conference (ACC)*. IEEE. 2012, pp. 4251–4256.

[88] K. Zhang, Q. Sun, and Y. Shi. "Trajectory tracking control of autonomous ground vehicles using adaptive learning MPC". In: *IEEE Transactions on Neural Networks and Learning Systems* 32.12 (2021), pp. 5554–5564.

[89] W. Wang et al. "Design, Modeling, and Nonlinear Model Predictive Tracking Control of a Novel Autonomous Surface Vehicle". en. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. Brisbane, QLD: IEEE, May 2018, pp. 6189–6196. ISBN: 978-1-5386-3081-5. DOI: 10.1109/ICRA.2018.8460632. URL: https://ieeexplore.ieee.org/document/8460632/ (visited on 07/17/2024).

[90] B. Houska, H. J. Ferreau, and M. Diehl. "ACADO toolkit—An open-source framework for automatic control and dynamic optimization". en. In: *Optimal Control Applications and Methods* 32.3 (May 2011), pp. 298–312. ISSN: 0143-2087, 1099-1514. DOI: 10.1002/oca.939. URL: https://onlinelibrary.wiley.com/doi/10.1002/oca.939 (visited on 07/03/2024).

[91] D. Falanga, P. Foehn, P. Lu, and D. Scaramuzza. "PAMPC: Perception-Aware Model Predictive Control for Quadrotors". In: *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*. 2018.

[92] T. M. C. Sears, M. R. Cooper, S. R. Button, and J. A. Marshall. *OtterROS: Picking and Programming an Uncrewed Surface Vessel for Experimental Field Robotics Research with ROS 2*. 2024. arXiv: 2404.05627 [cs.RO]. URL: https://arxiv.org/abs/2404.05627.

[93] L. F. W. Batista, S. Aravecchia, and C. Pradalier. *Sim-to-Real Transfer and Robustness Evaluation of Reinforcement Learning Control with Integrated Perception on an ASV for Floating Waste Capture*. 2025.

[94] P. Huang, T. Wang, F. Shkurti, and T. D. Barfoot. "Field Testing of a Stochastic Planner for ASV Navigation Using Satellite Images". In: *IEEE Transactions on Field Robotics* 1 (2024), pp. 131–160. DOI: 10.1109/TFR.2024.3450408.

[95] B. Lin, W. Xie, Y. Shi, B. Du, C. Zhang, and W. Zhang. "Robust Target Interception Strategy for a USV With Experimental Validation". In: *IEEE Robotics and Automation Letters* 8.11 (2023), pp. 7042–7049. DOI: 10.1109/LRA.2023.3300235.

[96] Y.-T. Ang, W.-K. Ng, Y.-W. Chong, J. Wan, S.-Y. Chee, and L. B. Firth. "An Autonomous Sailboat for Environment Monitoring". In: *2022 Thirteenth International Conference on Ubiquitous and Future Networks (ICUFN)*. 2022, pp. 242–246. DOI: 10.1109/ICUFN55119.2022.9829573.

[97] P. Dash et al. "Evaluation of water quality data collected using a novel autonomous surface vessel". In: *OCEANS 2021: San Diego–Porto*. IEEE. 2021, pp. 1–10.

[98] G. Ferri, A. Manzi, F. Fornai, F. Ciuchi, and C. Laschi. "The HydroNet ASV, a Small-Sized Autonomous Catamaran for Real-Time Monitoring of Water Quality: From Design to Missions at Sea". In: *IEEE Journal of Oceanic Engineering* 40.3 (2015), pp. 710–726. DOI: 10.1109/JOE.2014.2359361.

[99] G. M. Owusu, J. Kwabena Fosu Okyere, S. K. Armah, and P. Lawerh Kwao. "A Low-Cost Autonomous Surface Vehicle (ASV) for Plastic Waste Collection on Water Bodies". In: *2024 12th International Conference on Control, Mechatronics and Automation (ICCMA)*. 2024, pp. 41–46. DOI: 10.1109/ICCMA63715.2024.10843900.

[100] S. R. Sudha, M. Popović, and E. M. Coates. "An Informative Planning Framework for Target Tracking and Active Mapping in Dynamic Environments with ASVs". In: *arXiv preprint arXiv:2508.14636* (2025).

[101] D. Hoeller, N. Rudin, D. Sako, and M. Hutter. "Anymal parkour: Learning agile navigation for quadrupedal robots". In: *Science Robotics* 9.88 (2024), eadi7566.

[102] I. Radosavovic, T. Xiao, B. Zhang, T. Darrell, J. Malik, and K. Sreenath. "Real-world humanoid locomotion with reinforcement learning". In: *Science Robotics* 9.89 (2024), eadi9579.

[103] E. Kaufmann, L. Bauersfeld, A. Loquercio, M. Müller, V. Koltun, and D. Scaramuzza. "Champion-level drone racing using deep reinforcement learning". In: *Nature* 620.7976 (2023), pp. 982–987.

[104] O. Lockwood and M. Si. "A Review of Uncertainty for Deep Reinforcement Learning". en. In: *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*. Vol. 18. Number: 1. Oct. 2022, pp. 155–162. DOI: 10.1609/aiide.v18i1.21959. URL: https://ojs.aaai.org/index.php/AIIDE/article/view/21959 (visited on 05/12/2025).

[105] H. Zhang et al. "Robust Deep Reinforcement Learning against Adversarial Perturbations on State Observations". In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020, pp. 21024–21037. URL: https://proceedings.neurips.cc/paper_files/paper/2020/hash/f0eb6568ea114ba6e293f903c34d7488-Abstract.html (visited on 05/12/2025).

[106] L. Da, J. Turnau, T. P. Kutralingam, A. Velasquez, P. Shakarian, and H. Wei. *A Survey of Sim-to-Real Methods in RL: Progress, Prospects and Challenges with Foundation Models*. arXiv:2502.13187 [cs]. Mar. 2025. DOI: 10.48550/arXiv.2502.13187. URL: http://arxiv.org/abs/2502.13187 (visited on 05/12/2025).

[107] E. Salvato, G. Fenu, E. Medvet, and F. A. Pellegrino. "Crossing the reality gap: A survey on sim-to-real transferability of robot controllers in reinforcement learning". In: *IEEE Access* 9 (2021), pp. 153171–153187.

[108] X. Hu, S. Li, T. Huang, B. Tang, R. Huai, and L. Chen. "How simulation helps autonomous driving: A survey of sim2real, digital twins, and parallel intelligence". In: *IEEE Transactions on Intelligent Vehicles* 9.1 (2023), pp. 593–612.

[109] M. Quigley et al. "ROS: an open-source Robot Operating System". In: *ICRA workshop on open source software*. Vol. 3. 3.2. Kobe. 2009, p. 5.

[110] R. Szeliski. *Computer vision: algorithms and applications*. Springer Nature, 2022.

[111] M. L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.