Real World Implementation of Deep Reinforcement Learning Policies for Environmental Missions with Autonomous Surface Vehicles

Dame Seck Diop[1*][0009−0004−0752−1041], Alejandro Mendoza Barrionuevo[1][0000−0002−6931−7124], Alejandro Casado-Pérez[1][0009−0007−5572−5913], Samuel Yanes Luis[2][0000−0002−7796−3599], Sergio L., Toral[1][0000−0003−2612−0388], and Daniel Gutiérrez Reina[1][0000−0002−2481−5058]

[1] Electronics Engineering Department, Universidad de Sevilla, Spain
[2] Department of Electronic Technology, Universidad de Sevilla, Spain
[*]Corresponding Author: dseck@us.es

Abstract. This paper presents a pipeline for deploying Deep Reinforcement Learning policies trained in simulation onto real-world Autonomous Surface Vehicles for environmental missions. The proposed framework targets the sim-to-real gap by integrating previously developed Deep Reinforcement Learning approaches for water-quality monitoring and floating-debris cleanup within the unified GuadalPlanner architecture, supporting simulation and real-world Informative Path Planning. Two real-world field experiments were conducted with a custom fleet of two vehicles at Lake Mayor, in Alamillo Park (Seville, Spain). In the water-quality monitoring, the system adapts navigation strategies to heterogeneous sensor quality, resulting in differentiated exploration behaviors based on data fidelity. In the cleanup mission, a reward-greedy policy selection mechanism enables dynamic and asynchronous task allocation, allowing agents to switch between exploration and cleaning based on locally evaluated rewards. Across both tasks, the system demonstrates stable multi-agent coordination and successful execution of simulation-trained policies under real deployment conditions. The results show that the proposed architecture enables effective transfer of learned Deep Reinforcement Learning policies from simulation to real environmental missions while highlighting remaining challenges related to the sim-to-real gap.

Keywords: Autonomous Surface Vehicles · Deep Reinforcement Learning · Sim-to-Real Transfer · Multi-Agent Systems · Environmental Missions

1 Introduction

Water resources, including rivers, seas, and lakes, are vital for sustaining life and diverse ecosystems, yet they are increasingly threatened by pollution, cli-

mate change, and uncontrolled human activity [5]. Poor water quality leads to toxic cyanobacterial blooms [8] and dissolved oxygen depletion [4]. These factors cause human illness, biodiversity loss, and economic damage to sectors like aquaculture [18]. Additionally, plastic waste represents 85% of marine litter [17] and threatens over 800 species through ingestion or entanglement [7]. Large plastics eventually fragment into microplastics which enter the food chain and become nearly impossible to recover [19]. Traditional monitoring and cleanup methods rely on manual sampling or manned vessels. These approaches are too labor intensive and expensive for large scale applications [9].

Autonomous Surface Vehicles (ASVs) have emerged as efficient alternatives to traditional water monitoring, leveraging advances in battery technology and embedded computing [11]. These platforms integrate multiparameter water quality probes and perception systems such as LIDAR and stereo cameras for object detection [13]. ASVs are increasingly deployed for active mitigation tasks, including autonomous collection of floating waste [10]. We developed a custom fleet of 2 ASV platforms for environmental monitoring and cleanup missions [13]. These catamaran-hulled vehicles feature high-performance computing payloads, multiparameter probes for continuous water quality measurement, and stereo cameras for visual perception and debris detection.

The integration of sensor data into advanced decision-making frameworks is crucial for moving beyond toy examples toward real-world autonomous operations [13]. Recent advancements in Artificial Intelligence (AI) have promoted the use of Deep Reinforcement Learning (DRL) [12] to manage the decision-making processes of ASV fleets, allowing them to adapt to the dynamic and uncertain nature of aquatic environments [6]. In previous works, we established the theoretical foundations for two primary environmental mission types using DRL algorithms. For environmental monitoring, we adopted an Informative Path Planning (IPP) approach [2]. This strategy optimizes the trajectories of ASV fleets by autonomously identifying and navigating to areas of the highest informational value to minimize model error. To achieve this, the framework leverages Gaussian Processes (GPs) [20] as surrogate models to capture spatial contamination patterns and account for sensor noise [2]. For environmental cleanup, our methodology tackles the problem by decoupling the mission into two specialized tasks: exploration, which aims to maximize map coverage to localize trash, and cleaning, which focuses on the efficient collection of identified debris [15]. The system utilizes a multi-task learning architecture to learn each task policy and then adaptively switches from one task to another in real-time based on environmental conditions using a reward-greedy selection mechanism [15].

Despite the strong performance of these algorithms in simulated environments, a significant research gap persists regarding the "reality gap" (the transition from simulation to real) world deployment [3]. Real aquatic environments introduce complexities such as sensor noise, communication delays, and unpredictable hydrodynamics that are often simplified during training [14]. Moving from simulation to reality requires a robust hardware and software framework



Fig. 1: Alpha Amber and Beta Blue platforms during an environmental mission in Guadaira river (Sevilla, Spain).

capable of decoupling high-level AI policy execution from low-level vehicle stability and control [1].

The contribution of this work is the demonstration of a pipeline capable of bridging the gap between simulation of aforementioned DRL algorithms addressed in previous works [2, 15] and real world ASVs for different types of environmental missions. To this end, GuadalPlanner [1] is used as it provides an experimental architecture designed for simulation and deployment of IPP algorithms. To the best of our knowledge, it is the only experimental architecture explicitly devised to deploy DRL-based IPP policies on real ASV platforms [1], which motivates its adoption in this work rather than a general-purpose robotics middleware.

2 Autonomous Surface Vehicles: Alpha Amber & Beta Blue

The physical foundation of the ASVs showed in Fig. 1, namely Alpha Amber and Beta Blue, is based on a catamaran-style structure composed of two parallel hulls connected through a modular frame. This configuration was selected due to its inherent stability, wide beam, and increased payload capacity when compared to mono-hull alternatives, which is particularly advantageous for autonomous operations in dynamic aquatic environments [13].

Both ASVs share similar dimensions, approximately 1.3 m in length and 1 m beam, but offer different operational advantages. Alpha Amber is a custom-built, robust platform designed for high-payload missions exceeding 30 kg [13]. In contrast, Beta Blue utilizes the commercial BlueBoat platform, prioritizing portability and rapid deployment due to its modular, foldable design, though it is limited to a 15 kg payload.

To support autonomous operations, both ASVs implement a hierarchical control architecture that decouples high-level decision-making from low-level control tasks. The high-level layer executes AI-driven algorithms, including DRL policies and perception pipelines, powered by an NVIDIA Jetson Xavier NX on Alpha Amber and a more advanced Jetson Orin NX on Beta Blue. These edge computing units enable real-time sensor fusion and mission planning while

serving as the primary communication bridge for remote monitoring and policy updates. The low-level control layer is identical for both platforms, consisting of a Raspberry Pi 4 paired with a BlueRobotics Navigator running ArduPilot within BlueOS. This subsystem handles hardware abstraction and state estimation by fusing IMU, magnetometer, and GPS data through an Extended Kalman Filter (EKF). By managing low-level actuation and orientation, this architecture enables the high-level controller to issue abstract navigation commands, such as waypoints, which the low-level system translates into precise motor actions.

Both ASVs utilize an identical sensing and communication suite to ensure consistent data acquisition. Environmental monitoring is handled by an AML-3 XC probe (measuring pH, conductivity, and dissolved oxygen), while perception tasks are managed by a Stereolabs ZED2i stereo camera for depth-aware object detection. A BlueRobotics Ping2 sonar provides bathymetric data for shallow-water safety. High-precision navigation is achieved through RTK-enabled GPS for centimeter-level accuracy [13], with continuous connectivity maintained via industrial-grade 4G LTE routers. Both ASVs utilize a differential thrust configuration for agile maneuverability and zero-radius turning. The platforms are powered by parallel 4S LiPo batteries, providing four to six hours of autonomy depending on mission intensity [13].

3 Methodology of Environmental Missions

The methodologies described in this section have been previously validated in our earlier research [2, 15]. Firstly, the Deep Reinforcement Learning framework that enables coordinated decision-making across the fleet will be detailed.

3.1 Deep Reinforcement Learning Framework

The decision-making process for the ASV fleet is formalized as a Partially Observable Markov Game (POMG). This mathematical framework generalizes Reinforcement Learning (RL) to scenarios where multiple agents interact within a shared environment but possess limited perception. In our case, environmental data is unknown at the episode's beginning. Agents perceive the environment through observation functions $O(s)$, execute control actions a, and receive rewards r providing feedback on mission progress. They progressively learn, improving movement policies π^* by maximizing expected accumulated rewards through reward optimization. To manage high-dimensional aquatic environments, we utilize Deep Q-Learning (DQL), which employs Deep Neural Networks (DNNs) to learn optimal control policies directly from sensor observations.

For fleet coordination, we adopt Multi-Agent Deep Reinforcement Learning (MADRL) with parameter sharing, allowing all agents to share a single centralized neural network policy. Scalability is supported through egocentric state representations, where observations are relative to each vehicle's perspective, enabling the same policy to execute regardless of fleet size, as demonstrated in our previous works [2, 15].

3.2 Environmental Monitoring Problem

Following previous work [2], the primary objective in the monitoring problem is to optimize vehicle trajectories to minimize the global model error relative to the ground truth. This is achieved by guiding the fleet towards areas of high informational value to efficiently reconstruct the spatial distribution of physicochemical parameters through the integration of GPs as surrogate models. They capture spatial correlation between samples using RBF kernel, typical for smooth variations found in aquatic environments.

The core novelty of this work lies in the introduction of an adaptive decision-making framework that dynamically adjusts the fleet's trajectory based on the sensing qualities of the individual agents, characterized by the standard deviation. High-level decision-making is driven by a Double DQL algorithm implemented through a centralized DNN that coordinates the movements of the agents. The state representation is formalized as a observation function that feeds the neural network with image-like structured matrices, showed in Fig. 2. A key novelty introduced in [2] is the introduction of the knowledge map. This matrix stores the minimum standard deviation achieved at each discretized cell, representing the best confidence level obtained by any agent's sensor to date. When an agent traverses a cell, the knowledge map is updated only if the current sensor's standard deviation is lower than the previously stored value. This way, the network always knows which is the level of accuracy for each part of the model.



Fig. 2: Observation function of DNN for environmental monitoring during a state. From left to right: scenario map, GP model, knowledge map, observing agent position and other agents position.

At each discrete time step, the network evaluates actions corresponding to eight cardinal and diagonal movement directions on a discretized grid representation of the lake (Lake Mayor, the deployment site detailed in Section 4). The learning process is guided by a tailored reward function designed to encourage actions that reduce model error.

3.3 Environmental Cleanup Problem

The cleanup mission decouples into two specialized tasks: exploration to maximize map coverage and localize debris, and cleaning to physically collect identified trash. This is modeled as a POMG where agents make sequential decisions based on partial observations, since trash hotspot locations are unknown

at the start of the mission. The system utilizes a Multi-Task Deep Q-Network (MDQN) with a shared Convolutional Neural Network (CNN) backbone that extracts navigational features and branches into two specialized Q-heads. Each head estimates task-specific Q-values: Q_e for exploration and Q_c for cleaning. A scalar parameter $\nu \in \{0, 1\}$ determines task selection: $\nu = 1$ executes exploration actions and $\nu = 0$ executes cleaning actions.

Task switching is governed by Reward-Greedy Policy Selection (RGPS) [15], which evaluates all 2^N possible task combinations for N ASVs (where each agent is assigned either exploration or cleaning, encoded by ν) at each timestep and greedily selects the configuration maximizing scalarized joint reward. This enables per-agent, per-timestep task allocation, allowing individual vehicles to adaptively switch behaviors based on local conditions. Learning is driven by a vectorized reward signal $R_{t,n} = [R_{e,n}, R_{c,n}]$. The exploration reward incentivizes discovering new cells while penalizing inactivity and redundant visits, while the cleaning reward maximizes trash collection.

Since exploration and cleaning present conflicting goals, the methodology is formalized as a multi-objective optimization problem. By training the MDQN across varying preference weights $w \in [0, 1]$, the framework reconstructs a Pareto front of non-dominated policies, providing mission planners with adaptive strategies ranging from pure coverage to prioritized waste recovery. These preference weights w are the *user preferences* depicted in Fig. 3: at deployment time the operator selects a single weight that fixes the desired trade-off between coverage and waste recovery. Fig. 3 illustrates the diagram of the proposed algorithm.

4 From Simulation to Real-World

The primary testing ground is Lake Mayor in Alamillo Park (Seville, Spain). This environment is discretized into an eight-connected grid graph $G = (V, E)$, where each node $v_{i,j}$ represents a visitable cell of approximately 25 m^2. A binary obstacle mask matrix $M[i, j]$ encodes non-navigable regions, such as shorelines and internal islands, resulting in 351 navigable nodes. This relatively low node count is explained by the coarse cell size and the irregular shape of the lake. All missions are subject to finite energy constraints represented as a maximum distance budget d_{max} or step horizon T_{max}.

GuadalPlanner is a unified experimental architecture that bridges the sim-to-real gap by decoupling high-level decision-making from vehicle-specific control through standardized interfaces, enabling the same algorithm logic to execute, without modification, across three levels: algorithm-level simulation (research), Software-in-the-Loop via ArduPilot (validation), and real-world deployment on physical ASVs. In real-world implementation, the operational pipeline functions as a closed-loop system. The ROS2-based vehicle middleware reads telemetry and sensor data (GPS, orientation, environmental variables) from the MAVLink-compatible autopilot via MAVROS and transmits them to the centralized GuadalPlanner server via MQTT over 4G LTE. GuadalPlanner constructs a multi-channel state representation (stacked image-like matrices encoding the

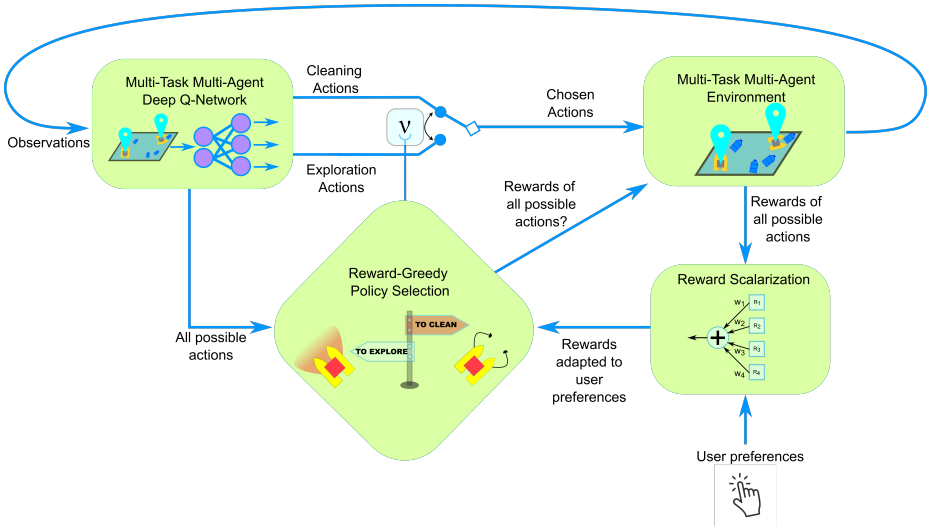


Fig. 3: Diagram of the proposed environmental cleanup algorithm. The *user preferences* input corresponds to the preference weight $w \in [0, 1]$ that selects the operating point on the Pareto front, trading off map coverage against waste recovery.

discretized map, environmental model, and agent positions) from this raw data, as seen in Section 3. The DRL policy selects the next action, which maps to a georeferenced GPS waypoint within the graph. This target is published via MQTT to the vehicle's middleware, forwarded to the autopilot through MAVROS for autonomous navigation. Upon reaching the destination, the vehicle publishes a confirmation message, triggering the next DRL decision cycle based on newly acquired environmental data.

Environmental monitoring missions are executed within the *gpBasedEnv* module of GuadalPlanner. Training maps are generated using Shekel functions [16], which provide smooth spatial distributions with controllable contamination peaks, mimicking real-world parameters like turbidity or pH. Plastic cleanup missions are modeled in the *trashCleanEnv* scenario. In this scenario, floating macroplastics are distributed in clustered patterns, evolve over time through wind-driven and random drift, and are detected by the agents through a limited-range sensing model.

5 Real World Experiments

The goal of these field experiments is to demonstrate the complete deployment pipeline, from simulation-based training to physical execution, rather than to re-validate the underlying learning algorithms, whose quantitative performance (model-reconstruction error, coverage, and Pareto optimality) was established in

their respective works [2, 15]. Accordingly, the experiments serve as a real-world feasibility demonstration of the GuadalPlanner pipeline; we report the metrics that are meaningful under real deployment, namely trajectory execution, area coverage, and task-switching behavior, and we explicitly delimit the simulated mission components. As previously mentioned, the field validation was carried out through two distinct multi-agent missions at the same location, utilizing the Alpha Amber and Beta Blue ASVs.

5.1 Environmental Monitoring

During this experiment, both ASVs were deployed with the objective of obtaining a spatial distribution model of water conductivity at Lake Mayor, covering a maximum trajectory distance of approximately 0.5 km per vehicle.

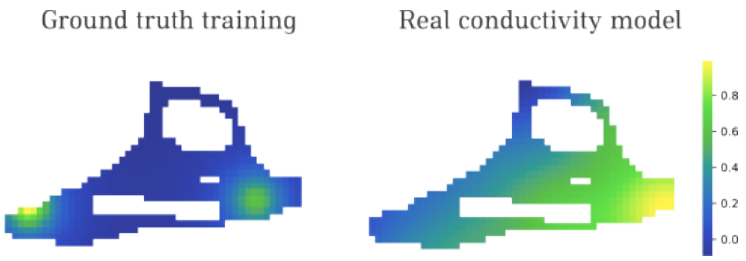


Fig. 4: Comparison between an example of ground truth used during training (left), and the normalized conductivity field obtained in real-world (right).

To evaluate the adaptive capabilities of the DRL policy under heterogeneous sensing conditions, since both vehicles have identical sensing capacities, a discrepancy in sensor quality was intentionally introduced. While both vehicles gathered real-time conductivity data, an artificial Gaussian noise was injected into the measurements of one vehicle. This controlled degradation directly impacted the predictive variance and the overall confidence of the GP surrogate model in the regions traversed by the noisier agent.

This field experiment demonstrates that the centralized DRL policy proposed in [2], originally trained in simulation, successfully transfers its decision-making capabilities to a physical deployment. Instead of executing symmetrical or redundant paths, the policy optimized the individual trajectories based on each vehicle's hardware fidelity. The system continuously processes raw, real-world telemetry and environmental data to generate meaningful navigation strategies in real time. Furthermore, the success of this mission demonstrates that DRL algorithms can be applied to real-world scenarios using the architecture proposed in GuadalPlanner [1].

A significant sim-to-real challenge encountered during the physical deployment involves the discrepancy between the simulated ground-truth assumptions



Fig. 5: Real-world trajectories of the high-precision agent (white), and the agent with the degraded sensor (red).

and the actual environmental conditions of the lake, as seen in Fig. 4. During the training phase, the DRL agents were exposed to highly contrastive contamination scenarios modeled via normalized Shekel functions that simulated localized pollutant spills mapped strictly between 0 (no contamination) and 1 (maximum peak), meaning the vast majority of the simulated map featured values close to 0. In contrast, the real-world deployment took place in a non-polluted, stable aquatic environment where physical conductivity readings lacked severe spikes, exhibiting only minor, smooth spatial variations. To feed these readings into the neural network, the inputs had to be dynamically normalized using min-max scaling to match the expected $[0, 1]$ input range; however, this artificial scaling amplified negligible fluctuations, where a value of 1 merely denoted the highest baseline conductivity found during the mission rather than an actual contamination peak. Despite this significant shift in the data distribution, where the network had to interpret smooth gradients instead of stark pollutant boundaries, the DRL policy provides good coverage of nearly the entire lake, as shown in Fig. 5. However, to improve behaviour in future work, it would be advisable to train the model using more accurate ground truths if the algorithm is to be used in real-world situations where there are very slight variations in the water's physicochemical parameters. Although it would be ideal to report the GP reconstruction error, this is impossible due to the lack of ground truth in real-world data.

5.2 Environmental Cleanup

This experiment evaluates the deployment of the previously developed DRL-based RGPS mechanism [15], described in Subsection 3.3. Using the GuadalPlanner architecture [1], both Alpha Amber and Beta Blue executed the learned policies onboard. The objective was not to re-validate the learning methodology, but to demonstrate the complete pipeline that transfers policies trained in simulation to ASVs operating in the real world.

The ground-truth trash distribution used in the field experiment is shown in Fig. 6. Importantly, no physical contamination of Lake Mayor was introduced. The trash distribution, drift, detection, and collection processes were simulated

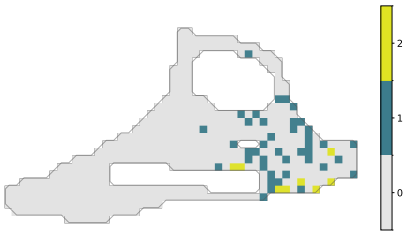


Fig. 6: Simulated ground-truth trash distribution used in the real-world mission.

to preserve the environmental integrity of the lake while maintaining a realistic mission structure. Fig. 7a presents an example of the trajectories generated during one experimental run. Dashed trajectory segments correspond to exploration behavior ($\nu = 1$), while solid segments represent cleaning behavior ($\nu = 0$). Fig. 7b shows the georeferenced GPS trajectories executed by the ASVs on Lake Mayor, illustrating the physical navigation component of the experiment.



(a) Simulated trajectories of the 2 ASVs. (b) Satellite view of Lake Mayor with the georeferenced GPS trajectories of the ASVs.

Fig. 7: ASV trajectories during an environmental cleanup mission: (a) simulated mission-level paths; (b) georeferenced GPS tracks executed by the physical ASVs at Lake Mayor.

The scope of the experiment is therefore hardware-in-the-loop: navigation, fleet communication, GPS logging, and onboard policy execution were performed on real ASVs, whereas the trash field, drift, detection, and collection were simulated. Consequently, no claim is made regarding the physical collection of real debris. The quantitative validation of the underlying RGPS policy, including coverage ratio, collected-trash ratio, and Pareto optimality, is reported in [15].

6 Conclusions and Future Works

This work demonstrates a complete pipeline that bridges the gap between simulation-based development of DRL methods and their deployment on real-world ASVs for environmental missions. Building upon prior works on environmental monitoring [2] and environmental cleanup [15], we show that DRL policies trained in simulation can be successfully transferred to physical platforms without modifying the core decision-making architecture. The integration of these methods is enabled by GuadalPlanner [1], an architecture designed to support both simulation and real-world execution of IPP systems. The real-world experiments show that the proposed approach successfully executes both monitoring and cleanup missions using onboard inference on two ASVs, preserving coordinated multi-agent behavior under real deployment constraints.

Despite these results, a sim-to-real gap remains due to differences between synthetic training environments and real-world environmental distributions. In particular, the normalization of real conductivity data and the absence of true pollutant fields introduce a mismatch in the semantic interpretation of state variables. Additionally, task evaluation under RGPS becomes computationally expensive as the number of agents increases due to the combinatorial nature of the action space.

Future work will focus on further reducing the reality gap between simulation and real-world deployment. This includes improving the fidelity of simulated environments to better match real environmental distribution observed in field conditions.

Acknowledgements

Grant PID2024-158365OB-C21 funded by MICIU/AEI/10.13039/501100011033 and by "ERDF/EU".

References

1. Barrionuevo, A.M., Diop, D.S., Pérez, A.C., Reina, D.G., Marín, S.L., Luis, S.Y.: A unified experimental architecture for informative path planning: from simulation to deployment with guadalplanner. arXiv preprint arXiv:2602.10702 (2026)
2. Barrionuevo, A.M., Luis, S.Y., Reina, D.G., Marín, S.L.T.: Informative deep reinforcement path planning for heterogeneous autonomous surface vehicles in large water resources. IEEE Access **12**, 71835–71852 (2024)
3. Batista, L.F., Aravecchia, S., Hutchinson, S., Pradalier, C.: Evaluating robustness of deep reinforcement learning for autonomous surface vehicle control in field tests. arXiv preprint arXiv:2505.10033 (2025)
4. Chen, C.H., Shang, Y.J., Wu, Y.C., Lin, Y.C.: Development of autonomous electric usv for water quality detection. Sensors **25**(12), 3747 (2025)
5. Dash, P., Moorhead, R.J., Herman, J., Beshah, W., Sankar, M., Moorhead, J., Chesser, G.D., Lowe, W., Simmerman, J., Turnage, G., et al.: Evaluation of water quality data collected using a novel autonomous surface vessel. In: OCEANS 2021: San Diego–Porto. pp. 1–10. IEEE (2021)

6. Diop, D.S., Luis, S.Y., Esteve, M.P., Marín, S.L.T., Reina, D.G.: Decoupling patrolling tasks for water quality monitoring: A multi-agent deep reinforcement learning approach. IEEE Access **12**, 75559–75576 (2024)
7. Harding, S.: Marine debris: understanding, preventing and mitigating the significant adverse impacts on marine and coastal biodiversity. Tech. rep., Secretariat of the Convention on Biological Diversity (2016)
8. Hitz, G., Pomerleau, F., Garneau, M.E., Pradalier, C., Posch, T., Pernthaler, J., Siegwart, R.Y.: Autonomous inland water monitoring: Design and application of a surface vessel. IEEE Robotics & Automation Magazine **19**(1), 62–72 (2012)
9. Huri, A., Mohd Nor, N.S., Saputra, H., Ismail, K., Fazila, M.Z., Keyon, A.: Real-time in situ water quality monitoring: Experimental validation using unmanned surface vehicle (usv). Semarak International Journal of Applied Sciences and Engineering Technology **5**, 1–15 (04 2025). https://doi.org/10.37934/sijaset.5.1.115
10. Kamarudin, N., Mohd Nordin, I.N.A., Misman, D., Khamis, N., Razif, M., Hanim, F.: Development of water surface mobile garbage collector robot. Alinteri Journal of Agriculture Sciences **36**, 534–540 (06 2021). https://doi.org/10.47059/alinteri/V36I1/AJAS21076
11. Katsouras, G., Dimitriou, E., Karavoltsos, S., Samios, S., Sakellari, A., Mentzafou, A., Tsalas, N., Scoullos, M.: Use of unmanned surface vehicles (usvs) in water chemistry studies. Sensors **24**(9), 2809 (2024)
12. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al.: Human-level control through deep reinforcement learning. nature **518**(7540), 529–533 (2015)
13. Pérez, A.C., Barrionuevo, A.M., Luis, S.Y., Diop, D.S., Díaz, L.M., Perales-Esteve, M., Toral, S.L., Reina, D.G.: Autonomous surface vehicle for water monitoring using artificial intelligence methodologies. In: Smart Water Quality Monitoring: Artificial Intelligence, Automation and Analytical Chemistry, pp. 291–322. Springer (2026)
14. Qiao, Y., Yin, J., Wang, W., Duarte, F., Yang, J., Ratti, C.: Survey of deep learning for autonomous surface vehicles in marine environments. IEEE Transactions on Intelligent Transportation Systems **24**(4), 3678–3701 (2023)
15. Seck, D., Yanes-Luis, S., Perales-Esteve, M., Marín, S.T., Gutiérrez-Reina, D.: Adaptive policy switching for multi-agent asvs in multi-objective aquatic cleaning environments. Sensors **26**(2), 427 (2026)
16. Shekel, J.: Test functions for multimodal search techniques. In: Fifth Annual Princeton Conf. on Information Science and Systems, 1971 (1971)
17. United Nations Environment Programme: From pollution to solution: A global assessment of marine litter and plastic pollution. Tech. rep., UNEP, Nairobi, Kenya (2021)
18. Valada, A., Tomaszewski, C., Kannan, B., Velagapudi, P., Kantor, G., Scerri, P.: An intelligent approach to hysteresis compensation while sampling using a fleet of autonomous watercraft. In: International Conference on Intelligent Robotics and Applications. pp. 472–485. Springer (2012)
19. Van Cauwenberghe, L., Claessens, M., Vandegehuchte, M.B., Janssen, C.R.: Microplastics are taken up by mussels (mytilus edulis) and lugworms (arenicola marina) living in natural habitats. Environmental Pollution **199**, 10–17 (2015). https://doi.org/https://doi.org/10.1016/j.envpol.2015.01.008, https://www.sciencedirect.com/science/article/pii/S026974911500010X
20. Williams, C.K., Rasmussen, C.E.: Gaussian processes for machine learning, vol. 2(3). MIT press Cambridge, MA (2006)