2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

A Deep Reinforcement Learning Framework and Methodology for Reducing the Sim-to-Real Gap in ASV Navigation

Luis F. W. Batista,[1,2,*] Junghwan Ro,[1,*] Antoine Richard[3], Pete Schroepfer[1,2,], Seth Hutchinson[1], and Cedric Pradalier[2]

***Abstract*— Despite the increasing adoption of Deep Reinforcement Learning (DRL) for Autonomous Surface Vehicles (ASVs), there still remain challenges limiting real-world deployment. In this paper, 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 by 13.1% while reducing task completion time by 7.4%. These findings, supported by sharing our open-source implementation, hold the potential to impact the efficiency and versatility of ASVs, contributing to environmental conservation efforts.**

I. Introduction

Water plastic pollution poses a significant threat to global sustainability, impacting marine life, ecosystems, and human health [1]. To restore ecological balance, it is necessary to not only 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 [2], [3]. The task of cleaning water surfaces autonomously has gathered attention not only from the academic sphere, [3], [4] 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 [5]. Recent work has even shown that the DRL-based approach is capable of outperforming traditional control algorithms in real-world trials [6]. Yet, field applications of DRL for ASVs are scarce, with most studies not extending beyond simulations, especially in the context of energy-efficient agents.

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

Fig. 1: Simulated and Real ASV Platforms Used Through our Experiments

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 [7] 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 [8], [9], [10].

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. Our work's main contributions include:

- Open-source, highly parallelized hydrodynamics and buoyancy implementation suitable for reinforcement learning framework[1].
- 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.

This research was supported by the French Agence Nationale de la Recherche, under grant ANR-23-CE23-0030 (project R3AMA).

[1] are with Georgia Institute of Technology, Atlanta, USA
[2] is with GeorgiaTech Europe - IRL2958 GT-CNRS, Metz, France
[3] is with University of Luxembourg, Luxembourg
[*] These authors contributed equally to this work.

[1] https://github.com/luisfelipewb/RL4WasteCapture/

II. Related Work

There is a growing adoption of Deep Learning (DL) on ASVs in the areas of perception, control, and multi-robot collaboration [5]. 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 [5].

To address the complex dynamics, DRL-based controllers have been explored as a potential solution. In [11], a DRL-based controller was used for path following. Here, [11] demonstrated that in simulation, a DRL-based controller was capable of controlling ASVs in extremely complex systems. Similarly, [12] 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 [13], [14], [15]. 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 [6]. An RL-based strategy to integrate guidance and heading control, improving tracking accuracy and robustness of the controller is proposed in [16]. 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 [7]. They extend previous initiatives to simulate buoyancy and hydrodynamics [17] and provide a solid starting point. Cieślak introduced Stonefish [18], 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, [19] 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.

III. Problem Formulation

A. 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 realistic field of view of 90 degrees and capping the operational range at 10 meters.

B. Dynamic Model

The dynamic model of the ASV in the simulation is based on Fossen's six degrees of freedom model [20].

$$\mathbf{M}\dot{\nu} + \mathbf{D}(\nu)\nu + g(\eta) = \tau_{\text{thruster}} + \tau_{\text{disturbance}} \tag{1}$$

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.

$$\mathbf{D}(\nu) = \mathbf{D}_l + \mathbf{D}_q(\nu) \tag{2}$$

$\mathbf{D}$ 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 ($< 2\ m/s$).

IV. Methodology

Our proposed methodology shown in Fig 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.

A. 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.

$$\mathbf{D}_l = \begin{bmatrix} X_u & 0 & 0 \\ 0 & Y_v & 0 \\ 0 & 0 & N_r \end{bmatrix} \tag{3}$$

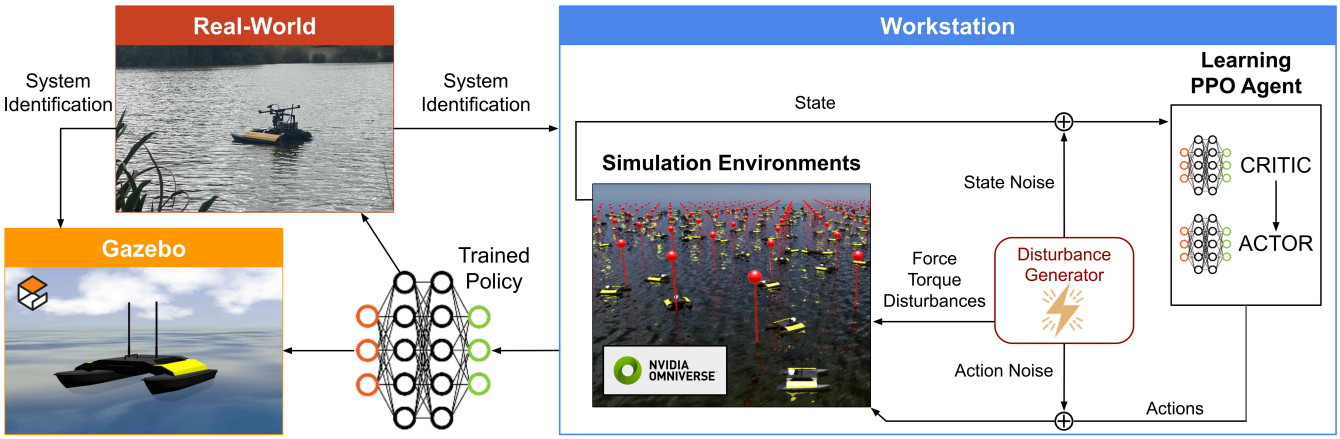

Fig. 2: Integrated pipeline for DRL-Based ASV Control consisting of four main steps. 1. Real-world experiments for system identification; 2. Train policy in simulation; 3. Tests using an independent simulation environment; 4. Real-world tests.

$$\mathbf{D}_q(\nu) = \begin{bmatrix} X_{u|u|}|u| & 0 & 0 \\ 0 & Y_{v|v|}|v| & 0 \\ 0 & 0 & N_{r|r|}|r| \end{bmatrix} \tag{4}$$

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 [21].

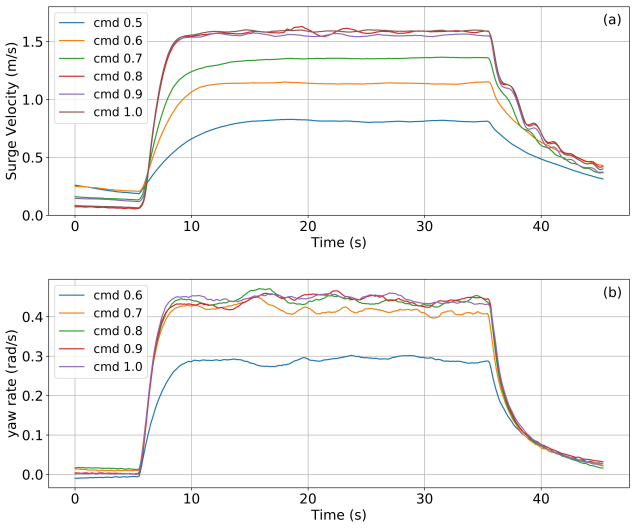

Fig. 3: System Identification Experiments. (a) shows surge velocity of the ASV during the acceleration test, and (b) shows the yaw rate of the ASV during the rotation test

Fig. 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. Fig. 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 Tab. I.

TABLE I: Hydrodynamics 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

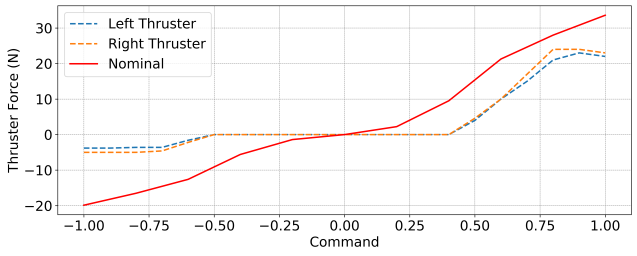

Fig. 4: Comparison of Nominal and Measured Thruster Forces

Fig. 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.

B. RL Framework for ASV

Expanding upon the established methodologies from [22] and [23], [24], 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 IsaacSim to simulate realistic marine conditions. In particular, we performed a GPU accelerated port of the UUV gazebo plugin [17] to Isaac Sim. This port allows, for instance, the ability to compute the hydrodynamic forces of thousands of systems seamlessly on GPU. Similarly to [22], this simulator is built on top of Nvidia's Isaac Sim and OmniIsaacGym [25], 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 models such as PPO [26] 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.

C. 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.

$$\mathbf{o}_t = [u_t \quad v_t \quad r_t \quad \cos(\delta_t^{\text{head}}) \quad \sin(\delta_t^{\text{head}}) \quad d_t]^T \tag{5}$$

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 (Fig. 5). This approach of focusing only on the local frame enhances the model's generalizability.

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.

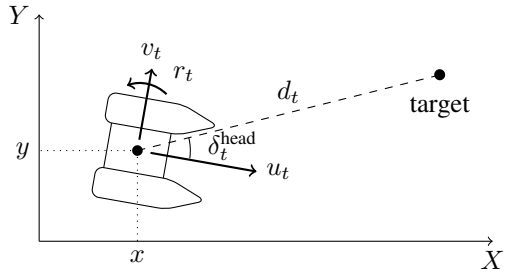

Fig. 5: Diagram of the ASV's observation space components

$$r_t = r_t^{\text{dist}} + r_t^{\text{head}} + r_t^{\text{energy}} + r_t^{\text{alpha}} + r_t^{\text{time}} + r_t^{\text{goal}} \tag{6}$$

Each reward term is detailed as follows:

$$r_t^{\text{dist}} = \lambda_1 \cdot (d_{t-1} - d_t) \tag{7}$$

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

$$r_t^{\text{head}} = \lambda_2 \cdot (e^{k_1(\delta_t^{\text{head}})^4} + e^{k_2(\delta_t^{\text{head}})^2}) \tag{8}$$

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

$$r_t^{\text{energy}} = \lambda_3 \cdot (e^{k_3 E} - 1) \tag{9}$$

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

$$r_t^{\text{alpha}} = \lambda_4 \cdot (e^{k_4|r_{t-1} - r_t|} - 1) \tag{10}$$

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

$$r_t^{\text{time}} = \lambda_5 \tag{11}$$

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

$$r_t^{\text{goal}} = \begin{cases} \lambda_6, & \text{if } d_t < d_{threshold} \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

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 II: 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 Tab. II, have been carefully calibrated to optimize the agent 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.

Tab. III 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.

TABLE III: 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

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: $\pm 3\ cm$, orientation: $\pm 0.025\ rad$) and perturbing action commands to reflect control imprecision. $\pm 10\%$ 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.

D. Simulation Environment

The initial evaluation of the trained models was conducted in the Gazebo simulation environment, using the UUV [17] plugin. This choice was motivated by the chance to validate the models' performance independence from Isaac Sim training environment and to facilitate ROS integration for field tests. Gazebo's parameters, calibrated with system identification parameters identified in Sec. IV-A, provided an environment for assessing model behavior.

E. 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 weight of $35.96\ Kg$. Its propulsion system is based on a thruster mounted on each hull, as shown in Fig. 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.

V. 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 Tab. I. Thruster force domain randomization (DR) was applied in some cases, resulting in four different tests (Tab. IV).

TABLE IV: Agents training configurations for experiments

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

A. 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 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.

B. 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 Fig. 7 where SID-DR provides smoother trajectories while reaching the target faster.

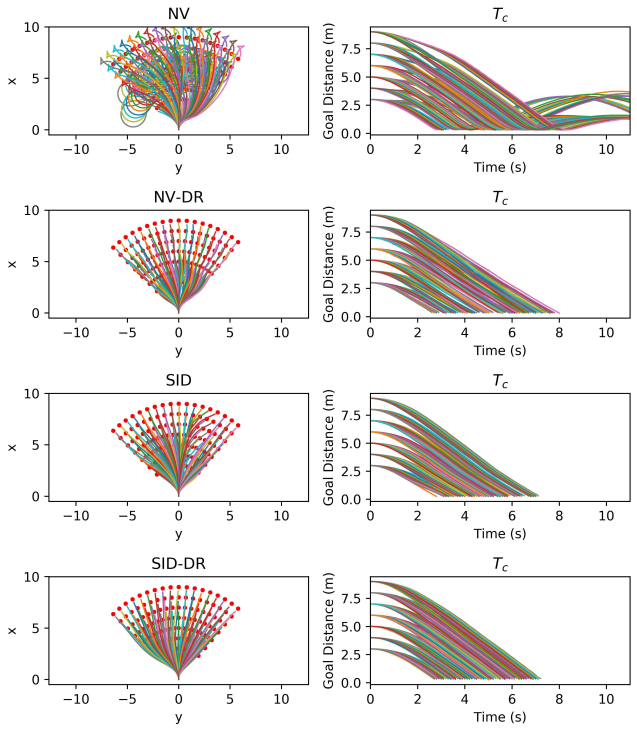

Fig. 6: Simulation tests showing trajectories (left) and goal distance over time (right). NV exhibits difficulty in reaching the target (red dot), while SID-DR completes the task faster.

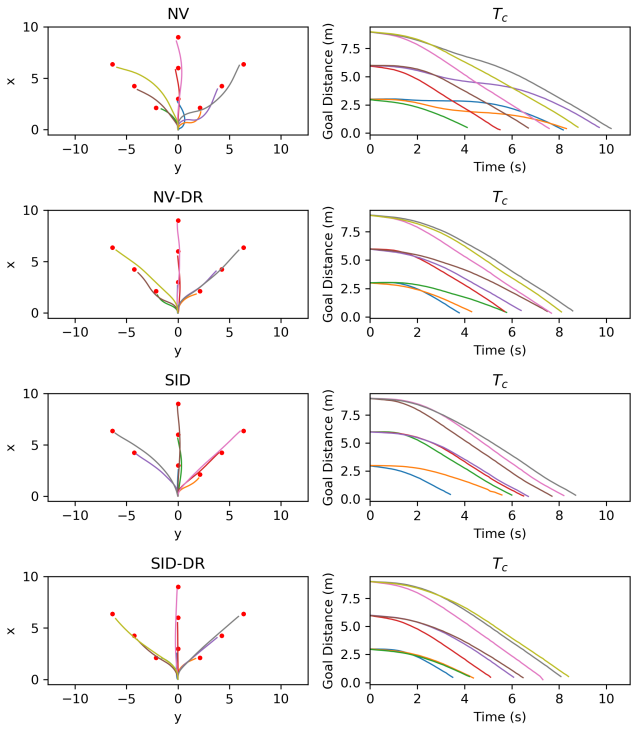

Fig. 7: Field tests demonstrate that all models are capable of reaching the target while SID-DR completes the task faster with smoother trajectories.

C. 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 Eq. 13, where N is the number of the data points and a_l^i and a_r^i are the action commands on thruster left and right.

$$E_{\text{acc}} \propto \sum_{i=1}^{N} (|a_l^i| + |a_r^i|) \tag{13}$$

D. Results

TABLE V: 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

Our field experiments demonstrated a 13.1% reduction in energy consumption and a 7.4% improvement in task completion efficiency when comparing the performances

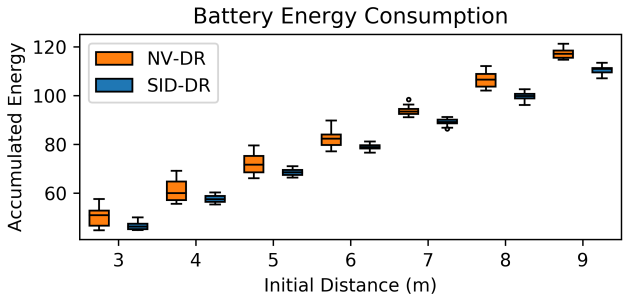

Fig. 8: Large-scale Simulation Analysis Shows that SID-DR Outperforms NV-DR in Battery Efficiency.

of NV-DR and SID-DR approaches. These numbers are extracted from the average across the different initial distances presented on Tab. V 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 Tab. V.

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 Fig. 8 corroborate the trends observed in real-world tests, specifically when comparing NV-DR and SID-DR across various initial distances.

VI. 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.

VII. Conclusion

In this study, we contribute to the field of marine robotics by enhancing 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. We are optimistic that the dissemination of our findings and open implementation can empower the research community, contributing to the development of more efficient and versatile autonomous systems in aquatic environments.

References

[1] T. van Emmerik and A. Schwarz, "Plastic debris in rivers," *Wiley Interdisciplinary Reviews: Water*, vol. 7, no. 1, p. e1398, 2020.

[2] F. Fornai, G. Ferri, A. Manzi, F. Ciuchi, F. Bartaloni, and C. Laschi, "An autonomous water monitoring and sampling system for small-sized asvs," *IEEE Journal of Oceanic Engineering*, vol. 42, no. 1, pp. 5–12, 2017.

[3] 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," *Sensors*, vol. 21, no. 4, p. 1102, 2021.

[4] 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.

[5] Y. Qiao, J. Yin, W. Wang, F. Duarte, J. Yang, and C. Ratti, "Survey of deep learning for autonomous surface vehicles in marine environments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 4, pp. 3678–3701, 2023.

[6] W. Wang, X. Cao, A. Gonzalez-Garcia, L. Yin, N. Hagemann, Y. Qiao, C. Ratti, and D. Rus, "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.

[7] B. Bingham, C. Agüero, M. McCarrin, J. Klamo, J. Malia, K. Allen, T. Lum, M. Rawson, and R. Waqar, "Toward maritime robotic simulation in gazebo," in *OCEANS 2019 MTS/IEEE SEATTLE*. IEEE, 2019, pp. 1–10.

[8] M. Lewis, K. Sycara, and P. Walker, *The Role of Trust in Human-Robot Interaction*. Cham: Springer International Publishing, 2018, pp. 135–159. [Online]. Available: https://doi.org/10.1007/978-3-319-64816-3_8

[9] L. Wells and T. Bednarz, "Explainable ai and reinforcement learning—a systematic review of current approaches and trends," *Frontiers in Artificial Intelligence*, vol. 4, 2021. [Online]. Available: https://www.frontiersin.org/articles/10.3389/frai.2021.550030

[10] 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.

[11] 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," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 10, pp. 6208–6220, 2020.

[12] 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.

[13] Q. Zhang, W. Pan, and V. Reppa, "Model-reference reinforcement learning for collision-free tracking control of autonomous surface vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 8770–8781, 2021.

[14] N. Wang, Y. Gao, H. Zhao, and C. K. Ahn, "Reinforcement learning-based optimal tracking control of an unknown unmanned surface vehicle," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 7, pp. 3034–3045, 2020.

[15] 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," *Neurocomputing*, vol. 484, pp. 26–37, 2022.

[16] 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," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[17] 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.

[18] 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.

[19] 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," *arXiv preprint arXiv:2310.04266*, 2023.

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

[21] 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," *Ocean Engineering*, vol. 127, pp. 305–324, 2016.

[22] M. El-Hariry, A. Richard, and M. Olivares-Mendez, "Rans: Highly-parallelised simulator for reinforcement learning based autonomous navigating spacecrafts," *arXiv preprint arXiv:2310.07393*, 2023.

[23] A. Richard, S. Aravecchia, T. Schillaci, M. Geist, and C. Pradalier, "How to train your heron," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5247–5252, 2021.

[24] 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.

[25] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," 2021.

[26] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017.