Benchmarking and Training Simulation Frameworks for USV Navigation in Congested and Disturbance-filled Environments

Xi Lin and Brendan Englot

***Abstract*— Safe maritime navigation is a challenging problem for vehicles subjected to wind and wave disturbances in congested environments. We have developed a benchmarking simulation framework for evaluating autonomous navigation systems under different levels of congestion, wind and waves. We have also developed a training framework for creating Deep Reinforcement Learning (DRL) based navigation policies that can learn to operate robustly under environmental disturbances. A total of eight classical and DRL-based navigation systems have been implemented for use with both frameworks.**

I. INTRODUCTION

Navigation safety in high-traffic maritime environments is of paramount importance. According to [1], 25% of vessel losses reported around the world happened in the maritime region around South China, Indochina, Indonesia, and the Philippines from 2014 to 2023, which is related to the huge volume of imports and exports flowing through the region and the resulting high levels of shipping traffic. In addition, the South China Sea is an area of significant typhoon activities, where approximately 9.2 tropical cyclones occurred annually during 1981–2015 [2], and the resulting strong waves and winds may significantly affect vehicle stability and threaten navigation safety.

In recent years, an increasing number of Unmanned Surface Vehicles (USVs) have been deployed in maritime missions to reduce human errors in operation and enhance navigation safety. As we consider scenarios where strong environmental disturbances can significantly impact vehicle motion, there may be high uncertainty in a vehicle's interpretation of its perceived surroundings, and in the outcomes of its control commands. Additionally, the Convention on the International Regulations for Preventing Collisions at Sea (COLREGs) [3] can be vague in addressing multi-vehicle encountering situations, hence there can be uncertainty in inferring the intent of other vehicles in congested scenarios.

In this paper, we develop a benchmarking framework which can be used to evaluate the performance of USV navigation systems in scenarios with different traffic volumes, wind and wave strengths, and we also implement navigation systems based on state-of-the-art methods. The benchmarking framework is based on the Virtual RobotX (VRX) simulator [4], which can simulate realistic sensor measurements, along with the dynamics of USV interactions with wind and waves.

The authors are with the Department of Mechanical Engineering, Stevens Institute of Technology, Hoboken, NJ, USA {xlin26,benglot}@stevens.edu

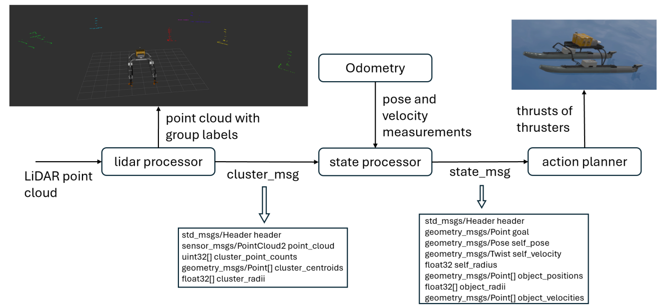

Fig. 1: Our benchmarking framework's USV navigation system.

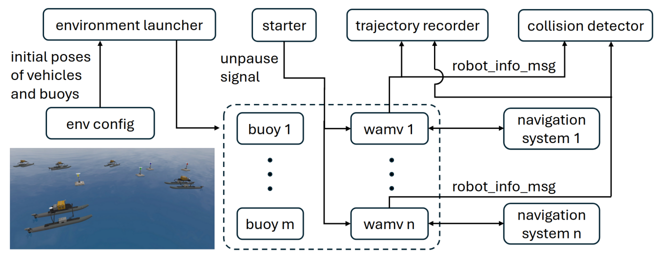

Fig. 2: Our benchmarking framework's experiment manager.

In addition, as Deep Reinforcement Learning (DRL) approaches have shown advantages in fostering robust robot navigation policies in recent years [5], we also develop another framework for training decision-making and control policies that learn to be robust to environmental disturbances with state-of-the-art DRL algorithms. The trained policies can be used for autonomous navigation without the need for further training.

Our benchmarking and training frameworks, as well as implementations of a total of eight classical and DRL based navigation systems, are freely available[1]. The user can also develop and integrate new sensor modalities and decision-making solutions into the existing frameworks conveniently.

II. BENCHMARKING FRAMEWORK - NAVIGATION SYSTEM

The structure of the navigation system is shown in Figure 1. We developed new ROS 2 packages to realize the framework, including three new node types, `lidar processor`, `state processor` and `action planner`, and two new message types, `cluster_msg` and `state_msg`. The `lidar processor` subscribes to the

[1] https://github.com/RobustFieldAutonomyLab/Distributional_RL_Decision_and_Control

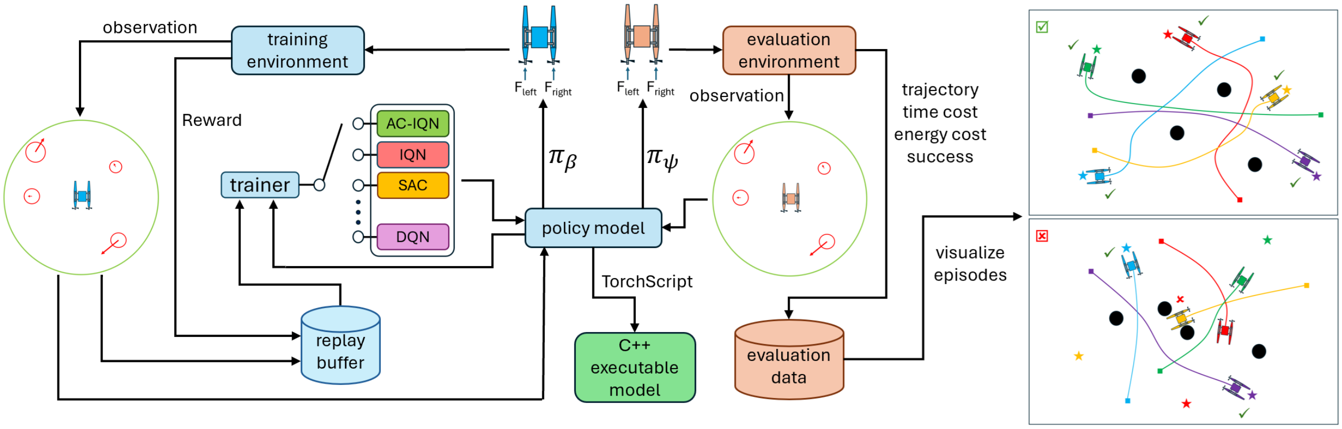

Fig. 3: **Training manager.** π_β is the behavior policy, for which we use $\epsilon - greedy$ to balance exploration and exploitation, and π_ψ is the greedy execution policy.

raw LiDAR point cloud topic, and performs point cloud segmentation with the method proposed in [6]. It returns point clouds with segmented cluster labels, and a `cluster_msg` that summarizes the estimated centroid and radius of every cluster. The `state processor` receives `cluster_msg` and odometry information, including ego pose and velocity measurements, and estimates the cluster velocities using current and previous measurements. Based on the received `state_msg`, the `action planner` generates thrust commands to control vehicle motions. We implemented Artificial Potential Field (APF)-based [7], Model Predictive Control (MPC)-based [8] and DRL-based planners. The DRL-based planners are introduced in Section IV.

III. Benchmarking Framework - Experiment Manager

The workflow of the experiment manager is shown in Figure 2. The `environment launcher` spawns buoys and vehicles in accordance with specified environment settings for a given episode of the simulation. Each vehicle exchanges information with an individual navigation system introduced in Section II. To ensure that all vehicles start at the same time, `action planner` in the navigation system will not generate thrust commands until receiving `unpause signal`, which will be sent to all vehicles simultaneously after the environment launch is completed. During the experiment, each vehicle sends `robot_info_msg`, including its pose, velocity and travel time data to `trajectory recorder`, which ends the episode if the goal is reached or the travel time exceeds the maximum time limit, and also to `collision detector`, which ends the episode after a collision happens.

IV. Training Manager

Training DRL-based decision-making policies requires extensive interactions between the learning agent and the environment, which is computationally expensive in the VRX simulator, which simulates realistic sensor measurements and complex interactions among USVs and waves. Thus, we perform training in a simplified 2D environment, and the training framework is shown in Figure 3. The policy model being trained matches the input and output of `action planner` in the navigation system in Figure 1, which receives observations that include position, radius and velocities of detected surrounding objects, and generates thrusts as control commands. The effects of wave and wind disturbances on robot motion and perception are not simulated in this training environment. Instead, we use a technique similar to [9], which models disturbance effects as random perception errors from Gaussian distributions, under which the policy model can gradually learn to be robust to corrupted observations. Every interaction between the agent and the training environment is stored in the replay buffer as a tuple of observation, action and reward. Periodically, a batch of experience tuples are sampled from the replay buffer and used to update parameters of the policy model with the corresponding DRL algorithm. We implemented DRL training algorithms and policy models based on Actor-Critic Implicit Quantile Networks (AC-IQN) [10], standard IQN [11], Soft Actor-Critic (SAC) [12], Deep Deterministic Policy Gradient (DDPG) [13], Rainbow [14] and Deep Q-Networks (DQN) [15]. To evaluate the training performance, after a specified number of iterations of training, the policy model is used to interact with an evaluation environment, and the resulting data is stored and can be visualized to understand policy behaviors. In Figure 3, all vehicles in the upper episode reach their respective goals without colliding with any other vehicles or obstacles, and the episode is successful. The lower episode is ended and deemed failed after a collision happens. After the training process finishes, the parameters of the trained PyTorch model are saved in a C++ executable model with TorchScript, which is loaded into the corresponding DRL-based `action planner` for thrusts inferences.

Acknowledgment

This research was supported by the Office of Naval Research, grants N00014-20-1-2570 and N00014-24-1-2522.

REFERENCES

[1] Allianz Commercial. Shipping and safety review 2024. [Online]. Available: https://commercial.allianz.com/news-and-insights/reports/shipping-safety.html

[2] S. Li, J. Li, J. Yang, Q. Bao, Y. Liu, and Z. Shen, "Monthly prediction of tropical cyclone activity over the South China Sea using the FGOALS-f2 ensemble prediction system," *Atmospheric and Oceanic Science Letters*, vol. 15, no. 2, p. 100116, 2022.

[3] International Maritime Organization, "Convention on the international regulations for preventing collisions at sea, 1972 (COLREGs)," 1972.

[4] B. Bingham, C. Aguero, M. McCarrin, J. Klamo, J. Malia, K. Allen, T. Lum, M. Rawson, and R. Waqar, "Toward maritime robotic simulation in Gazebo," in *Proceedings of MTS/IEEE OCEANS Conference*, Seattle, WA, October 2019.

[5] K. Zhu and T. Zhang, "Deep reinforcement learning based mobile robot navigation: A review," *Tsinghua Science and Technology*, vol. 26, no. 5, pp. 674–691, 2021.

[6] I. Bogoslavskyi and C. Stachniss, "Fast range image-based segmentation of sparse 3D laser scans for online operation," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 163–169.

[7] W. Naeem, S. C. Henrique, and L. Hu, "A reactive COLREGs-compliant navigation strategy for autonomous maritime navigation," *IFAC-PapersOnLine*, vol. 49, no. 23, pp. 207–213, 2016.

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

[9] H. Hu, K. Zhang, A. H. Tan, M. Ruan, C. Agia, and G. Nejat, "A sim-to-real pipeline for deep reinforcement learning for autonomous robot navigation in cluttered rough terrain," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6569–6576, 2021.

[10] X. Lin, P. Szenher, Y. Huang, and B. Englot, "Distributional reinforcement learning based integrated decision making and control for autonomous surface vehicles," *IEEE Robotics and Automation Letters*, vol. 10, no. 2, pp. 1194–1201, 2025.

[11] W. Dabney, G. Ostrovski, D. Silver, and R. Munos, "Implicit quantile networks for distributional reinforcement learning," in *International Conference on Machine Learning (ICML)*. PMLR, 2018, pp. 1096–1105.

[12] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International Conference on Machine Learning*. PMLR, 2018, pp. 1861–1870.

[13] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[14] M. Hessel, J. Modayil, H. Van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, and D. Silver, "Rainbow: Combining improvements in deep reinforcement learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[15] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.