Collision avoidance in maritime traffic under COLREGs constraints: a reinforcement learning approach

Antoine Wanctin, Thomas Chaffre, Matthew Stephenson, Paulo Santos, Karl Sammut, Benoit Clement

Collision avoidance in maritime traffic under COLREGs constraints: a reinforcement learning approach

Antoine Wanctin with Thomas Chaffre, Matthew Stephenson, Paulo Santos, Karl Sammut and Benoit Clement

September 28, 2024

Abstract

Space exploration is often too hazardous for humans to perform, relying on autonomous rovers and probes to safely operate. Sea navigation presents a similar challenge within a non-modelled open environment, with autonomous surface vehicles (ASVs) being developed to operate without direct human control. Before being integrated into actual traffic, ASVs must follow the International Regulations for the Prevention of Collisions at Sea (COLREGs) However, automating these rules is challenging since they have multiple interpretations depending on the situation. A COLREG-compliant framework is proposed where the ASV predicts the actions of nearby vessels. Most of the time, it will abide by the rules, except in extreme situations. Then, the ship must find a safe strategy to avoid collisions and should be able to depart from COLREGs if necessary. This article shows how Deep Reinforcement Learning can be used to achieve this task, specifically the Proximal Policy Optimization. This method has been chosen because it is efficient at creating a policy solving the dual problem of path planning and collision avoidance while following rules. This framework has potential applications beyond maritime applications, such as spacecraft collision avoidance in congested orbits, highlighting the broader implications of this work in scenarios requiring protocol adherence in hazardous environments.

1 Introduction

Autonomous vehicles are increasingly present in all kinds of environments to perform tasks that are too dangerous or tedious for humans, with the global market for autonomous systems expected to grow significantly in the coming years [1]. These vehicles are often brought in to work alongside humans, necessitating the adoption of common rules and frameworks to facilitate collaboration [2]. This is even more important in space exploration, where the extreme environment amplifies the necessity for autonomous robots.

A similar challenge could be found in the maritime domain where Autonomous Surface Vehicles (ASVs) are now being used in a variety of applications such as oceanographic data collection, search and rescue, marine transport, and surveillance. The maritime industry, responsible for transporting over 80% of the world's goods, is experiencing continuous growth [3], which increases the risks associated with vessel collisions [4]. Maritime accidents not only pose significant threats to human lives and ecosystems but also contribute to economic losses and increased greenhouse gas emissions. The same study [3] estimates that shipping contributes to nearly 3% of global CO2 emissions and more importantly emissions could reach 130% of their 2008 levels by 2050, making efficient navigation systems critical to reducing environmental impact. ASV integration into the current maritime traffic is a challenge to be taken on. This is particularly concerning in high-traffic areas such as narrow channels, where collisions are most likely to occur.

The International Regulations for Preventing Collisions at Sea (COLREGs) are globally adopted standards to address the collision issue [5]. However, these regulations were designed for human-operated vessels, creating a gap in how they apply to autonomous systems. While several algorithms have been developed to incorporate COLREGs into autonomous vessels, many are limited in their ability to handle non-COLREGs-compliant behaviours, multiple encounters, or dynamic

and unpredictable conditions.

To achieve fully autonomous vessel operations, new approaches are needed. Existing methods, spanning from Rule-based [6] to the more recent Learning-based [7], have made strides toward integrating COLREGs into autonomous navigation systems, but they fall short in handling complex, real-world situations such as multi-encounter or adversarial scenarios. These limitations highlight the need for innovative solutions that not only comply with COLREGs but also adapt to scenarios where other vessels or environmental factors do not.

This thesis proposes the development of a novel algorithm that addresses these gaps by combining collision avoidance with efficient path planning in dynamic, multi-vessel environments. Our approach is designed to improve the adaptability and robustness of ASVs, ultimately facilitating their safe integration into existing maritime operations.

This paper is organized as follows: Section 2 presents the related work on collision avoidance and path planning for ASVs. Section 3 presents the methodology used to develop the COLREGs-compliant algorithm. Section 4 presents the results of the algorithm on a simulation environment. Finally, Section 5 concludes the paper and presents the future work.

2 Related Works

To be merged with the actual traffic, an ASV has to solve a certain number of challenges, from understanding the COLREGs to predicting the behaviour of the other vessels in its vicinity. The COLREGs were written in 1972 [5] to have a clear code of conduct when entering a possible collision scenario. They were written by sailors for sailors and are very vague in their content. They do not use measurements or numerical values for the description of the situations. They usually rely on the experience and knowledge of the Officer on Watch to make the correct decision while in a difficult or dangerous situation. Finding the correct answer to these situations while staying COLREG-compliant is a challenge for autonomous boats, as the rules are unclear. [8, 9] Another issue lies within the fact that the COLREGs only take into consideration scenarios with just two ships. The rules have been designed to work in this specific case only. However, in narrow channels or dense traffic areas, it is often necessary to consider more than two possible actors or obstacles to apply the COLREGs.

Several approaches have attempted to develop COLREG's compliant ASVs in recent years. The most common one is the creation of a 'ship zone' that divides the space around the ship into different sectors. Depending on which sector another vessel is in, a corresponding COLREG rule is applied. The shape of the zone around the boat can be a simple circle [10] or more complex depending on different factors such as the speed [11] or the surroundings [12]. An extensive review of ship zones and ship areas can be found in [13].

After defining how to implement COLREGs, the next step is to go to the objective while staying COLREG-compliant. The method used must solve the dual problem of path planning and collision avoidance, while also maintaining a certain degree of efficiency that is required for commercial ship applications. Several methods have been developed to achieve this goal. A common one is the use of a Dynamic Window Approach [14] that will compute the best velocity and steering angle to reach the objective while avoiding collisions. This method has several advantages, such as being able to compute the best trajectory in real-time, but also has some drawbacks, such as the fact that it is not able accurately predict the future position of the other vessels. Since the Dynamic Window Approach is a local path planning algorithm, it may not be optimal when the size of the trajectory becomes too important. Several classic algorithms of path planning and collision avoidance have been developed over the years, such as the class of Rapidly-exploring random trees algorithm [15] or the Velocity Obstacle [16]. These methods have shown good results in simple environments, but have some drawbacks in complex environments e.g. when there are multiple encounters.

Another possible solution that has been developed is the use of Deep Reinforcement Learning (DRL) [17, 18] to solve the path planning and collision avoidance problem. This method has been

used in several papers [19] and has shown good results in complex environments. Several kinds of DRL have been used in the literature, from the classical Q-Learning [20] to the Proximal Policy Optimization (PPO) algorithm [21]. The PPO algorithm has shown good results in complex environments over several studies [22, 23, 24, 25]. However, some drawbacks have been identified. Multiple encounters are often not taken into account. Vessels that are not following properly the COLREGs are not considered either. In [22] when multiple encounters are present, the algorithm is not able to take into account the possible interactions between the different vessels. No trajectory prediction is made.

In this paper, we propose a COLREGs-compliant algorithm that uses the PPO algorithm to solve the path planning and collision avoidance problem. The main innovation of this work is the development of a collision identification method that allows the ASV to choose a safe path by considering the possible trajectories of the other vessels. This method is based on the creation of a spatio-temporal representation of the different paths that can be taken by the other vessels. This representation will allow the algorithm to extract zones where avoidance maneuvers are to be applied. The algorithm will then assess the COLREGs-compliance degree of the possible trajectory and select the optimal action to take. In the case where the COLREGs are not applicable, the algorithm will be able to make a decision that will ensure the safety of the ship.

The final framework will have three merged components. First, a Path Planning algorithm will choose a path for the vessel in compliance with the COLREGs. At the same time, the situation assessment algorithm will identify the level of risk presented by the other vessels and predict their path. Finally, the decision-making algorithm will select the best action to take to ensure the safety of the ship. In the situation where following the COLREGs leads to a collision, the Reinforcement Learning algorithm will be able to make the best decision to ensure the safety of the ship, leaving the COLREGs aside to ensure the safety of the ship.

3 Methodology

3.1 COLREG-supported ASV simulator

To train the ASV to avoid collisions, a simulator must be used. It is indeed necessary to run a large number of experiments in an environment where the obstacles and the starting position of the vessel are randomised. Several simulators exist that could be used for our purpose but they are not tailored for it. The Fossen Simulator [26] for example recreates the behaviour of ASV in a 3D environment with an accurate physics engine. However, it does not take into account the collision with other vehicles or the COLREGs. Due to its complexity, it is also too slow to be used for training a reinforcement learning algorithm. Another possibility would have been the UTSeaSim Simulator [27] which is used for underwater robotics. A collision avoidance method is implemented into this simulator and it is possible to use it for multi-agent simulations. However, it is not tailored for the use of reinforcement learning algorithms and does not take into account the COLREGs.

To tackle these issues, a lightweight and quick simulator has been developed by the combined team at ENSTA-Bretagne and Flinders University. COLSim [28] is a 2D simulator for ASVs developed in Python. It has been designed to be fast, simple and easy to use. For each simulation, several actors with their position and speed must be specified. Each actor has its own goal to reach. The simulator will then calculate the new position of each actor for a defined number of steps. Using Artificial Potential Fields [29], the simulator will avoid collision between actors and obstacles while following some of the COLREGs.

Indeed, based on which sector of the other actor the agent is, it will take different actions. For example, if the agent is in the starboard sector of the other actor, it will turn to the right.

The simplicity of the simulator allows modifications to be made easily. For example, real AIS data have been gathered by another member of the team. After some post-processing, this data can be integrated into the simulator to recreate real situations where ships are using COLREGs. This data will be used to train the agent to avoid collisions in a more realistic environment using

trajectories from real ships.

The simulator has also been modified to be used as an environment compatible with Reinforcement Learning. Several changes have been made, to achieve this. Firstly, all of the visualisation tools have been removed to make the simulation faster and avoid wasting computation time. Then, a functionality has been added to the algorithm to be able to launch several simulations in a row. This will allow the agent to learn from different situations and improve its performance. To automatise the process, the position, speed, goal, heading and type of the actors are randomly generated at the beginning of each simulation. The agent will then have to learn to avoid collisions and reach its goal in the correct amount of time.

3.2 Deep Reinforcement Learning

Reinforcement Learning (RL) can be framed as a Markov Decision Process (MDP), where an agent interacts with an environment in discrete time steps. At each time step, the agent observes the current state of the environment, takes an action, and receives a reward, transitioning to the next state according to a transition probability distribution. The objective of RL is to learn a policy that maximizes the expected cumulative reward over time. In this context, the agent's decisions depend only on the current state, satisfying the Markov property.

Deep Reinforcement Learning (DRL) is a subfield of machine learning that combines reinforcement learning and deep learning. It trains agents to interact with an environment and learn a policy that maximizes a reward function. It has been used in a wide range of applications, from playing video games [30] to controlling complex robots [31].

Several different algorithms exist in the field of DRL, each with its strengths and weaknesses. Proximal Policy Optimisation (PPO) [21] is a policy gradient method directly optimizing the policy rather than a value function. The goal is to improve the policy over time by interacting with the environment and receiving feedback in the form of rewards. One of its major features is the use of a clipped surrogate objective function. PPO is clipping the probability ratio between the new and old policy, ensuring that the policy does not change too much between updates. This helps to stabilize the training process and prevent large policy updates that could lead to catastrophic forgetting. Here, PPO has been chosen as the algorithm to train the agent for several reasons. First, it is known for its stability and ease of use as it is a simple algorithm that is easy to implement and tune. Second, it has been shown to perform well on a wide range of tasks [32], making it a good choice for this project. Finally, the algorithm has been implemented in the stable-baselines library [33], which provides a simple and easy-to-use interface for training and evaluating DRL agents. The training time of the PPO model could be a drawback as it could be quite long in complex environments, but it is not a major issue as the training can be done offline.

3.3 Experimental setup

Here, the PPO is used as a reinforcement learning algorithm to train an agent to avoid collisions in a 2D environment. The agent must reach a goal while avoiding obstacles in a limited amount of time.

Task: The task is to train an agent to avoid collisions with obstacles while reaching a goal in a 2D environment. The agent must learn to navigate the environment and reach the goal in the shortest amount of time possible while avoiding collisions and staying within the bounds of the environment.

Environment: The environment shown in Figure 1 is a 2D grid where the agent and the obstacles are represented as circles. The obstacles are static and their number stays the same during the training. The environment is bounded and the agent cannot go outside of the bounds.

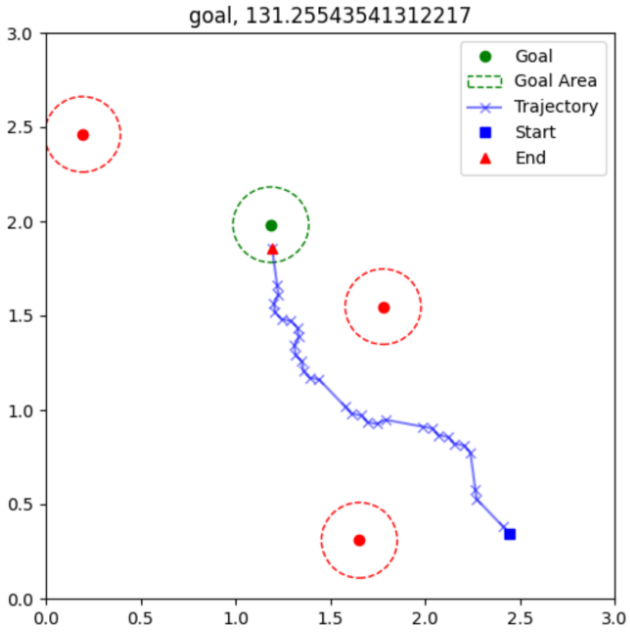


Figure 1: An episode in the Reinforcement Learning environment.

Action space: As the purpose of this algorithm is solely to avoid collisions, the action space is kept simple on purpose to speed up the learning and have a solid basis. The agent has 3 actions at its disposal.

- Move forward
- Turn left by θ degrees while advancing a certain distance
- Turn right by θ degrees while advancing a certain distance

State space:

The state space is composed of the following elements:

- The position of the agent
- The heading of the agent
- The position of the goal
- The distance between the agent and the goal
- The distance between the agent and the obstacles
- The last action taken by the agent

Its dimension is $5 + n\_obstacles$ with $n\_obstacles$ the number of obstacles present in the environment. The agent has 500 steps to reach the goal, as in the worst-case scenario it would take 200 steps to reach the goal. The size of the environment is 3x3 to reproduce the conditions of a busy narrow channel.

Reward function:

Each episode ends if the agent reaches one of the following conditions:

- The agent reaches the goal
- The agent collides with an obstacle

- The agent reaches the maximum number of steps
- The agent goes outside of the bounds of the environment

The termination criteria of the episode for reaching the goal or colliding with an obstacle are the same. The agent must be under a certain distance threshold to these elements to be considered as reached.

The reward is calculated using different components. We have:

$$R_t = R + R_{passing\_time} + R_{distance} \tag{1}$$

With R_t the total reward, R the reward for the termination of the episode, $R_{passing\_time}$ the reward for the time spent in the simulation and $R_{distance}$ the reward for the euclidean distance to the goal.

$$R = \begin{cases} 100 & \text{if distance\_to\_goal} \leq \text{threshold} \\ -20 & \text{if outside\_bonds} \\ -200 & \text{if collision} \end{cases} \tag{2}$$

$$R_{\text{passing\_time}} = -0.06 \cdot \text{current\_step} \tag{3}$$

$$R_{\text{distance}} = \exp(-\text{distance\_to\_goal}) \cdot 4 \tag{4}$$

The numerical rewards have been chosen empirically to encourage the agent to reach the goal as fast as possible while avoiding collisions. *Evaluation metrics:*

The agent is evaluated using the success rate per episode (using a moving window) and the average reward per episode. The success rate is the percentage of episodes where the agent reaches the goal. The average reward is obtained by the agent over all the evaluation episodes.

Hyper-parameters

A few hyper-parameters have been chosen to tune the PPO algorithm in Table 1. These parameters can be found below:

Parameter	**Value**
Learning rate	0.0003
Number of steps per simulation	500
Batch size	256
Number of epochs	5
Discount factor	0.99
GAE lambda	0.95
Clip range	0.2
Entropy coefficient	0.0
Value function coefficient	0.5
Maximum gradient norm	0.5

Table 1: Model Parameters

These hyper-parameters have been chosen empirically for most of them. However, the number of steps per simulation for example is directly linked to the size of the environment as the agent must be able to reach the goal in every possible situation. The batch size and the number of epochs have been chosen to be able to train the agent in a reasonable amount of time. The learning rate has been chosen to be small to prevent the agent from diverging.

4 Results

4.1 Simulator

For now, the agent inside the simulator obeys the following equations to move.

With x and y the position of the agent, v its speed and θ its heading the following state equations are used to move the agent:

$$\begin{cases} \dot{x} = v \cdot \cos(\theta) \\ \dot{y} = v \cdot \sin(\theta) \\ \dot{v} = u_1 \\ \dot{\theta} = u_2 \end{cases} \tag{5}$$

With u_1 and u_2 the control inputs of the agent. Each actor in the simulation has a goal to reach and must avoid collisions with the others. A path-planning algorithm must be specified and chosen over a collection of several algorithms. However, for now, the actors will only follow COLREGs if they use path planning based on artificial potential fields.

To achieve this result, the artificial potential fields around the obstacles are circled-shaped. Depending on the situation, this circle turns either clockwise or counterclockwise. This simulator is used to test the path planning algorithms and will be used to test the combination of path planning and reinforcement learning algorithms.

However, the environment for the Reinforcement Learning training is a bit different. To speed up the training of the collision avoidance procedure, the simulator is even more simplified as the agent still has the same state equations but is only confronted with static obstacles. All the other actors are not taken into account nor are the Artificial Potential Fields. It allows a significant reduction of the time taken to perform a simulation, and thus to train the agent. Ten million simulation steps can be performed in less than 3 hours on a single Intel Core i5 CPU.

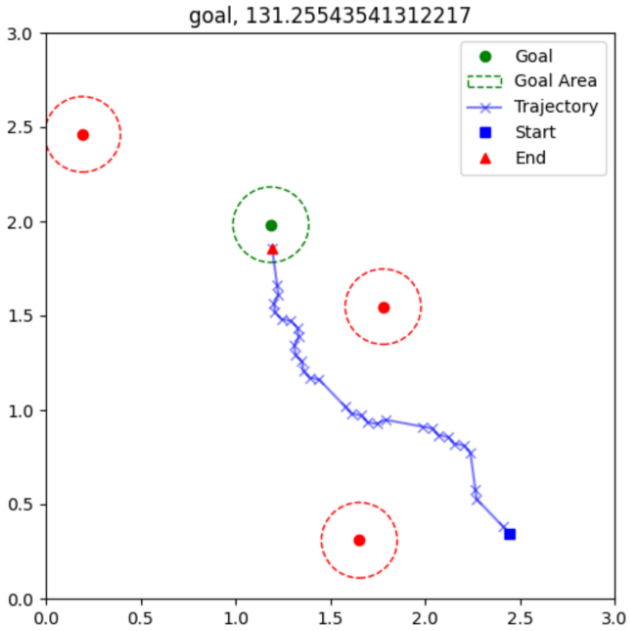
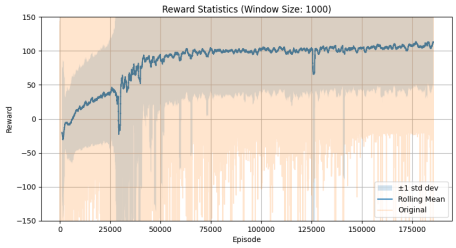


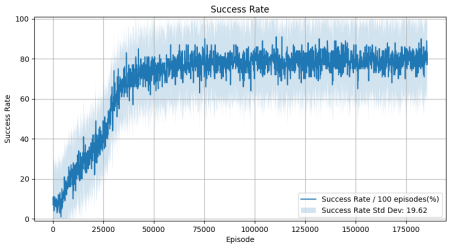
Figure 2: An episode in the Reinforcement Learning environment.

Here, in Figure 2, the agent's starting position is represented by a blue square and the ending one by a red triangle. The agent reaches the goal area when it is inside the green circle. It must

avoid the obstacles represented by the red circles.



(a) Reward per episode during training



(b) Success rate per episode during training

Figure 3: Training metrics for a 2 obstacles environment.

A clear improvement of the agent performances can be seen in Figure 3. The reward per episode and the success rate over 100 episodes are increasing and stabilizing.

In addition, an ablation study was performed to assess the robustness of the method against variations in the number of obstacles. To that end, we performed multiple training sessions for different numbers of obstacles. One agent is trained at a time from scratch and for the same number of episodes but with a different number of obstacles, ranging from 0 to 9. The resulting performances are provided in Table 2 below.

Number of obstacles	Mean Reward	Standard deviation
0	142.06	42.89
1	102.72	353.97
2	109.06	141.38
3	100.77	90.08
4	94.48	93.91
5	73.58	208.35
6	57.19	331.38
7	62.23	173.59
8	52.13	229.35
9	31.19	380.04

Table 2: Influence of the number of obstacles.

As we can see in Table 2, the performance of the agent is linearly correlated with the number of obstacles. The mean reward consistently decreases with the increase in the number of obstacles. On the other hand, the interpretation of the resulting reward standard deviation is less straightforward. In fact, for some number of obstacles, the resulting reward standard deviation is similar to cases with a lower number of obstacles, where the mean reward is higher. This could suggest that the impact of the number of obstacles is also correlated with another parameter which is not captured here. A potential candidate is the global geometry of the map as, for example, depending if the map is rectangular-shaped or circular-shaped, the addition of obstacles will lead to highly different ASV optimal trajectories.

5 Conclusion

This paper investigated a novel framework for integrating COLREGs into autonomous navigation systems for Autonomous Surface Vehicles. By leveraging Proximal Policy Optimization and developing a unique collision identification and prediction method, the proposed approach allows ASVs to plan safe paths in dynamic, multi-vessel environments while ensuring compliance with COLREGs. The framework includes a path-planning module, a situation assessment algorithm, and a decision-making process that balances between rules compliance and safety requirements.

The main focus of this study was the collision avoidance component of the overall framework. While all the individual components are functioning independently, they are not yet fully integrated. The collision avoidance module, in particular, requires further development, and improving its robustness will be a focus of future work. Additionally, further reward engineering will be done to ensure that reward maximization is only obtained by executing the desired behaviour. Future efforts will focus on these aspects, ensuring the complete framework operates cohesively to tackle more complex and realistic maritime scenarios.

References

[1] P. A. Hancock, I. Nourbakhsh, and J. Stewart, "On the future of transportation in an era of automated and autonomous vehicles," *Proceedings of the National Academy of Sciences*, vol. 116, pp. 7684–7691, Apr. 2019. Publisher: Proceedings of the National Academy of Sciences.

[2] S. Panagou, W. P. Neumann, and F. Fruggiero, "A scoping review of human robot interaction research towards Industry 5.0 human-centric workplaces," *International Journal of Production Research*, vol. 62, pp. 974–990, Feb. 2024. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/00207543.2023.2172473.

[3] *Towards a green and just transition*. No. 2023 in Review of maritime transport / United Nations Conference on Trade and Development, Geneva, Geneva: United Nations, 2023.

[4] EMSA, "Annual Overview of Marine Casualties and Incidents 2023." https://www.emsa.europa.eu/publications/item/5052-annual-overview-of-marine-casualties-and-incidents.html, June 2023.

[5] I. M. Organization, *COLREG: Convention on the International Regulations for Preventing Collisions at Sea, 1972*. International Maritime Organization, 2002. Accessed: 2024-06-27.

[6] S. Campbell and W. Naeem, "A Rule-based Heuristic Method for COLREGS-compliant Collision Avoidance for an Unmanned Surface Vehicle," *IFAC Proceedings Volumes*, vol. 45, pp. 386–391, Jan. 2012.

[7] L. Hu, H. Hu, W. Naeem, and Z. Wang, "A review on COLREGs-compliant navigation of autonomous surface vehicles: From traditional to learning-based approaches," *Journal of Automation and Intelligence*, vol. 1, p. 100003, Dec. 2022.

[8] C. Wilkinson, L. Grosser, M. Oppert, S. Banks, and B. Clement, "Automation at Sea and Human Factors," *15th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles*, 2024.

[9] B. Clement, M. Dubromel, P. E. Santos, K. Sammut, M. Oppert, and F. Dayoub, "Hybrid Navigation Acceptability and Safety," *Proceedings of the AAAI Symposium Series*, vol. 2, no. 1, pp. 11–17, 2023. Number: 1.

[10] X. Zhang, C. Wang, Y. Liu, and X. Chen, "Decision-Making for the Autonomous Navigation of Maritime Autonomous Surface Ships Based on Scene Division and Deep Reinforcement Learning," *Sensors*, vol. 19, p. 4055, Sept. 2019.

[11] Z. Pietrzykowski and M. Wielgosz, "Effective ship domain - Impact of ship size and speed," *Ocean Engineering*, vol. 219, p. 108423, Jan. 2021.

[12] R. Szlapczynski and J. Szlapczynska, "A ship domain-based model of collision risk for near-miss detection and Collision Alert Systems," *Reliability Engineering & System Safety*, vol. 214, p. 107766, Oct. 2021.

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

[14] X. Yuan, C. Tong, G. He, and H. Wang, "Unmanned Vessel Collision Avoidance Algorithm by Dynamic Window Approach Based on COLREGs Considering the Effects of the Wind and Wave," *Journal of Marine Science and Engineering*, vol. 11, p. 1831, Sept. 2023. Number: 9 Publisher: Multidisciplinary Digital Publishing Institute.

[15] H.-T. L. Chiang and L. Tapia, "COLREG-RRT: An RRT-Based COLREGS-Compliant Motion Planner for Surface Vehicle Navigation," *IEEE Robotics and Automation Letters*, vol. 3, pp. 2024–2031, July 2018. Conference Name: IEEE Robotics and Automation Letters.

[16] E. H. Thyri and M. Breivik, "Partly COLREGs-compliant collision avoidance for ASVs using encounter-specific velocity obstacles," *IFAC-PapersOnLine*, vol. 55, pp. 37–43, Jan. 2022.

[17] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction, 2nd ed.* Adaptive computation and machine learning., Cambridge, MA, US: The MIT Press, 2018.

[18] K. Kasaura, S. Miura, T. Kozuno, R. Yonetani, K. Hoshino, and Y. Hosoe, "Benchmarking Actor-Critic Deep Reinforcement Learning Algorithms for Robotics Control with Action Constraints," *IEEE Robotics and Automation Letters*, vol. 8, pp. 4449–4456, Aug. 2023. arXiv:2304.08743 [cs].

[19] A. Heiberg, T. N. Larsen, E. Meyer, A. Rasheed, O. San, and D. Varagnolo, "Risk-based implementation of COLREGs for autonomous surface vehicles using deep reinforcement learning," *Neural Networks*, vol. 152, pp. 17–33, Aug. 2022.

[20] H. Shen, H. Hashimoto, A. Matsuda, Y. Taniguchi, D. Terada, and C. Guo, "Automatic collision avoidance of multiple ships based on deep Q-learning," *Applied Ocean Research*, vol. 86, pp. 268–288, May 2019.

[21] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," Aug. 2017. arXiv:1707.06347 [cs].

[22] Z. Rongcai, X. Hongwei, and Y. Kexin, "Autonomous collision avoidance system in a multi-ship environment based on proximal policy optimization method," *Ocean Engineering*, vol. 272, p. 113779, Mar. 2023.

[23] E. Meyer, "COLREG-Compliant Collision Avoidance for Unmanned Surface Vehicle Using Deep Reinforcement Learning."

[24] W. Xie, L. Gang, M. Zhang, T. Liu, and Z. Lan, "Optimizing Multi-Vessel Collision Avoidance Decision Making for Autonomous Surface Vessels: A COLREGs-Compliant Deep Reinforcement Learning Approach," *Journal of Marine Science and Engineering*, vol. 12, p. 372, Mar. 2024. Number: 3 Publisher: Multidisciplinary Digital Publishing Institute.

[25] Z. Luman and M.-I. Roh, "COLREGs-compliant multiship collision avoidance based on deep reinforcement learning," *Ocean Engineering*, vol. 191, Oct. 2019.

[26] "Python Vehicle Simulator."

[27] "UTSeaSim - UT Austin's Naval Surface Navigation Simulator."

[28] B. Clement, P. Sarhadi, M. Dubromel, and T. Chaffre, "Colsim, a simulator for hybrid navigation acceptability and safety," in *2024 IFAC Conference on Control Applications in Marine Systems, Robotics and Vehicles (CAMS)*, Sept. 2024.

[29] K. Messaoudi, O. S. Oubbati, A. Rachedi, A. Lakas, T. Bendouma, and N. Chaib, "A survey of UAV-based data collection: Challenges, solutions and future perspectives," *Journal of Network and Computer Applications*, vol. 216, p. 103670, July 2023.

[30] K. A. ElDahshan, H. Farouk, and E. Mofreh, "Deep Reinforcement Learning based Video Games: A Review," in *2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*, pp. 302–309, May 2022.

[31] M. Rybczak, N. Popowniak, and A. Lazarowska, "A survey of machine learning approaches for mobile robot control," *Robotics*, vol. 13, no. 1, 2024.

[32] A. Raffin, J. Kober, and F. Stulp, "Smooth Exploration for Robotic Reinforcement Learning," June 2021. arXiv:2005.05719 [cs, stat].

[33] A. Hill, A. Raffin, M. Ernestus, A. Gleave, A. Kanervisto, R. Traore, P. Dhariwal, C. Hesse, O. Klimov, A. Nichol, M. Plappert, A. Radford, J. Schulman, S. Sidor, and Y. Wu, "Stable baselines." https://github.com/hill-a/stable-baselines, 2018.