Journal of *Marine Science and Engineering*

MDPI

Article

Robust Decision-Making for the Reactive Collision Avoidance of Autonomous Ships against Various Perception Sensor Noise Levels

Paul Lee *, Gerasimos Theotokatos * and Evangelos Boulougouris

Maritime Safety Research Centre (MSRC), Department of Naval Architecture, Ocean & Marine Engineering, University of Strathclyde, Glasgow G4 0LZ, UK; evangelos.boulougouris@strath.ac.uk
* Correspondence: p.lee@strath.ac.uk (P.L.); gerasimos.theotokatos@strath.ac.uk (G.T.)

Abstract: Autonomous ships are expected to extensively rely on perception sensors for situation awareness and safety during challenging operations, such as reactive collision avoidance. However, sensor noise is inevitable and its impact on end-to-end decision-making has not been addressed yet. This study aims to develop a methodology to enhance the robustness of decision-making for the reactive collision avoidance of autonomous ships against various perception sensor noise levels. A Gaussian-based noisy perception sensor is employed, where its noisy measurements and noise variance are incorporated into the decision-making as observations. A deep reinforcement learning agent is employed, which is trained in different noise variances. Robustness metrics that quantify the robustness of the agent's decision-making are defined. A case study of a container ship using a LIDAR in a single static obstacle environment is investigated. Simulation results indicate sophisticated decision-making of the trained agent prioritising safety over efficiency when the noise variance is higher by conducting larger evasive manoeuvres. Sensitivity analysis indicates the criticality of the noise variance observation on the agent's decision-making. Robustness is verified against noise variance up to 132% from its maximum trained value. Robustness is verified only up to 76% when the agent is trained without the noise variance observation with lack of its prior sophisticated decision-making. This study contributes towards the development of autonomous systems that can make safe and robust decisions under uncertainty.

Keywords: maritime autonomous surface ship; reactive collision avoidance; decision-making; deep reinforcement learning; deep deterministic policy gradient; robustness; safety; perception sensor; sensor noise; LIDAR

check for updates

Citation: Lee, P.; Theotokatos, G.; Boulougouris, E. Robust Decision-Making for the Reactive Collision Avoidance of Autonomous Ships against Various Perception Sensor Noise Levels. *J. Mar. Sci. Eng.* **2024**, *12*, 557. https://doi.org/10.3390/jmse12040557

Academic Editors: David Moreno-Salinas and Shiyuan Zheng

Received: 8 January 2024
Revised: 12 February 2024
Accepted: 15 February 2024
Published: 27 March 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

1.1. Background

The maritime sector is currently experiencing a paradigm shift towards the new "shipping 4.0" era, characterised by its increased connectivity, digitalisation, and autonomy [1]. The main driving factor of this transformation is the vision for a more sustainable future [2] propelled by the emergence of key enabling technologies, including the internet of things [3], cloud computing [4], big data analytics [5], blockchain [6], cyber-physical systems [7], digital twins [8], additive manufacturing [9], and augmented and virtual reality [10], as well as artificial intelligence (AI) [11]. Owning to such technologies, conventional ships have been gradually adopting various degrees of autonomy, forming a new class of maritime autonomous surface ships (MASSs) [12] with the premise of unlocking new levels of sustainability via reducing fuel consumption, optimising route planning, eliminating accidents related to erroneous human decision-making, and more [13].

One of the preconditions to achieve higher degrees of autonomy MASSs that do not require human intervention is the advancement of sensor technologies used as their primary source of perceiving the surrounding environment [14], known as perception

sensors [15]. Perception sensors are exteroceptive that measure values originated outside of the controlled system to perceive the surrounding environment, while localisation sensors are proprioceptive that measure values within the system to localise it with respect to the perceived environment [16]. In safety-critical operations, like collision avoidance, the role of perception sensors is pivotal not only for sensing the surrounding obstacles, but also due to their chain effect on the quality of situation awareness, safety of decision-making, and effectiveness of actuators control [17], as presented in Figure 1. Among the various global to local collision avoidance approaches [18], reactive collision avoidance is the most local and perception sensor-dependent one, as it requires real-time decisions made at the point of sensing and at the lowest level of actuators control to achieve the highest possible degree of autonomy for real-world applications, where information is often limited, dynamic, and uncertain [19].

However, despite the current advancements of sensor technologies and unique features of each perception sensor [20],degradation of measurements due to sensor noise is inevitable [21]. Sensor noise refers to the random variation inherent in the measurements that can be attributed to a wide range of factors from intrinsic limitations, such as a low resolution or design imperfections, to extrinsic influences, such as atmospheric conditions or electromagnetic interference [22]. The adverse marine environment adds another layer of complexity considering the extreme temperatures and humidity levels, corrosive nature of sea water, and visibility impairing rain, fog, or glare [23]. This further exacerbates the uncertainty of the measurements, which can manifest as false detection, such as false positive or false negative [24], or even as complete detection failure of a collision-inducing obstacle [25]. Addressing sensor noise during reactive collision avoidance, where the temporal and spatial margins for error are minimal, is a pressing challenge that needs to be investigated prior to the full-scale implementation of MASSs.

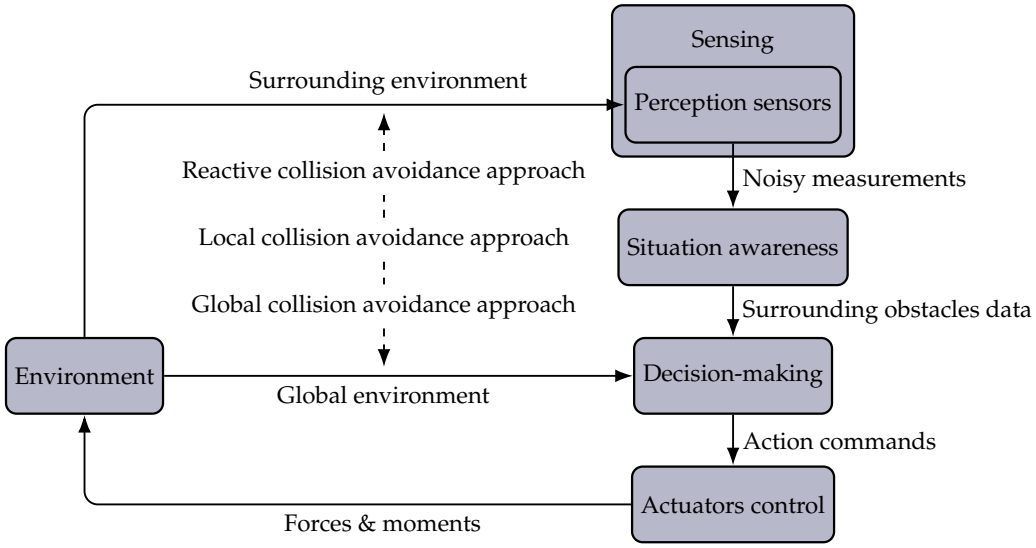


Figure 1. Block diagram of the high-level interaction between the main modules in global, local, and reactive collision avoidance approaches.

1.2. Literature Review

Previous studies employed several algorithms to address the reactive collision avoidance of MASSs in different scenarios. Xu et al. [26] employed modified artificial potential field and velocity obstacle in scenarios with a single static obstacle to multiple static obstacles. Blindheim et al. [27] used model predictive control in scenarios with multiple static obstacles, considering dynamic risk pertaining to emergency situations, such as impaired thrusters, total blackout, or strong winds. Gao et al. [28] employed particle swarm optimisation and dynamic window (DW) in scenarios with a single dynamic obstacle, while complying to the International Regulations for Preventing Collisions at Sea (COLREGs)

under head-on, crossing, and overtaking situations. Serigstad et al. [29] employed hybrid DW and a rapidly exploring random tree in scenarios with static and dynamic obstacles, while tracking a global path. Wang et al. [30] used a deep Q network (DQN) in scenarios with a single static obstacle or single to multiple dynamic obstacles with COLREGs compliance and good seamanship. Cheng et al. [31] used DQN in scenarios with multiple static obstacles and unknown environmental disturbances. However, a common limitation across these studies was the assumption of complete information of the navigating area and the obstacles without accounting for perception sensors.

Some studies addressed the above limitation by incorporating various perception sensors for obstacles detection. Zhou et al. [32], Meyer et al. [33], and Heiberg et al. [34] used general range finder sensors with multiple beams to measure the distances and relative angles to obstacles. Kim et al. [35], Gonzalez et al. [36], and Villa et al. [37] used light detection and rangings (LIDARs) to generate point clouds or clusters of the surrounding environment and measure the positions, distances, and relative angles to obstacles. Wang et al. [38] used radio detection and ranging (RADAR) to calculate the distance to the closest point of approach and the time to the closest point of approach. Song et al. [19] used automatic identification system for dynamic obstacles data. Peng et al. [39] used LIDAR and sound navigation and ranging to detect surface and underwater obstacles, respectively. However, these studies assumed ideal sensors without accounting for more realistic noisy measurements.

Some studies addressed the influence of noise on perception sensors using various noise filtering techniques. Han et al. [40] used an extended Kalman filter (EKF) for electro-optical and infrared camera, RADAR, and LIDAR measurements to predict the motion of obstacles. Han et al. [41] used EKF on RADAR measurements to predict the motion of obstacles. Stanislas et al. and Kim et al. [42] filtered RADAR and LIDAR measurements by experimentally tuning the intensity of basic RADAR settings and filtering out LIDAR measurements that exceeded a predefined threshold to estimate the state of obstacles. However, the primary focus of these studies was limited to filtering out sensor noise to enhance the situation awareness for the state estimation of the obstacles.

1.3. Aim & Contributions

The review of pertinent literature highlights a gap of investigating the direct impact of unfiltered noisy measurements of a perception sensor on the decision-making during safety-critical operations. Hence, this study aims to bridge this gap by proposing a methodology that enhances the robustness of decision-making for the reactive collision avoidance of a MASS against various perception sensor noise levels. The key contributions of this study are as follows.

- A novel approach that incorporates sensor noise for end-to-end decision-making pertaining to a deep reinforcement learning (DRL) agent is proposed, providing a way forwards towards a more effective integration of signal processing and decision-making techniques.
- An effective training framework for a DRL agent that enhances the robustness and sophistication of the agent's decision-making against various sensor noise levels is proposed.
- A systematic way to analyse and interpret the decision-making of a trained DRL agent is presented.
- A systematic way to analyse the sensitivity of a trained DRL agent to inputs pertaining to sensor noise is presented.

1.4. Outline

The remainder of this study is structured as follows. Section 2 presents the proposed methodology and the subsequent phases involved. Section 3 presents the rationale and characteristics of the investigated case study. Section 4 presents the results and discussion. Finally, Section 5 presents the main findings, limitations, and outlook for future studies.

2. Methodology

The proposed methodology is developed into five subsequent phases, as presented in Figure 2, with each phase serving a specific objective as outlined below.

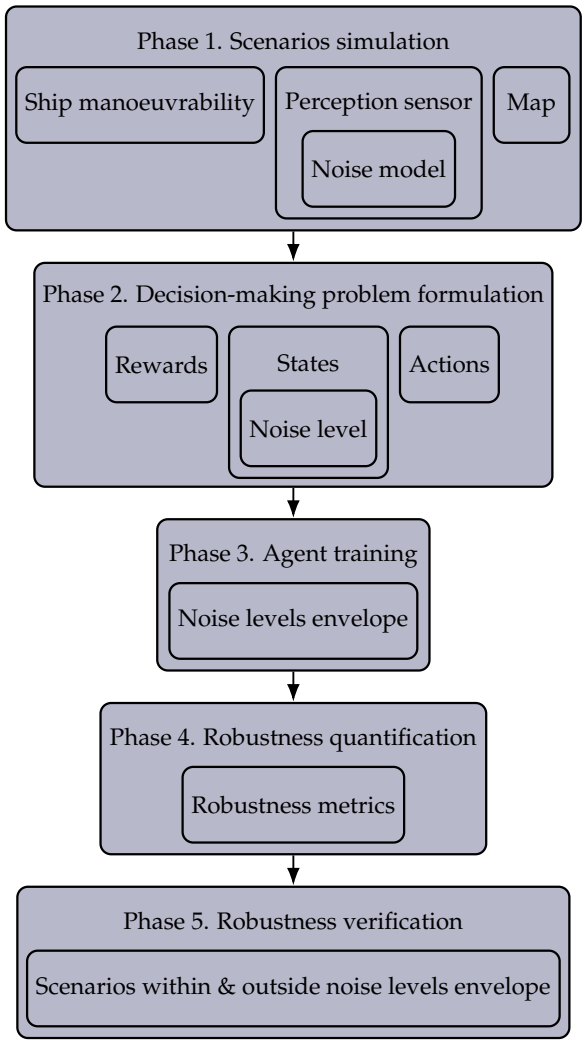


Figure 2. Flowchart of the proposed methodology consisting of the key steps in each phase.

- Phase 1. Scenarios simulation: The objective is to simulate scenarios pertaining to the reactive collision avoidance of a MASS using a noisy perception sensor, by employing the main components as digital twins. The employed digital twins are the ship manoeuvrability, perception sensor and its noise model, and map.
- Phase 2. Decision-making problem formulation: The objective is to formulate the decision-making problem pertaining to the investigated scenarios as a Markov decision process, by identifying the rewards, states, and actions. The identified rewards are associated with path following, nominal navigation, actuator control, and collision avoidance objectives. The identified states are associated with variables related to the rewards and noise level of the perception sensor. The identified actions are associated with the actuator control.
- Phase 3. Agent training: The objective is to train a DRL agent that can make decisions over the formulated problem, by developing a training framework. The developed training framework considers the noise levels envelope of the perception sensor investigated during the agent's training.
- Phase 4. Robustness quantification: The objective is to quantify the robustness of the trained agent's decision-making against various perception sensor noise levels, by

defining robustness metrics. The defined robustness metrics are associated with path following and collision avoidance.

- Phase 5. Robustness verification: The objective is to verify the robustness of the trained agent's decision-making against various perception sensor noise levels, by investigating various scenarios. The investigated scenarios are within and outside the trained noise levels envelope of the perception sensor.

2.1. Scenarios Simulation

The objective of this phase is to simulate scenarios pertaining to the reactive collision avoidance of a MASS using a noisy perception sensor, by employing the main components as digital twins. Digital twins stand for high-fidelity models that enable real-time virtual simulations, while accurately mirroring the full-scale reality of their physical twins [43]. Digital twins are particularly important for the investigation of safety-critical operations in extreme scenarios [44], where risk on the property's integrity, environmental protection, and human life is involved [45]. The digital twins of the main components, as delineated in this Section, are employed in MATLAB/Simulink 2023b, due to the customisation capabilities of a wide range of digital twin libraries, including the Navigation Toolbox [46], LIDAR Toolbox [47], and Robotics System Toolbox [48].

2.1.1. Ship Manoeuvrability

The digital twin of the own ship (OS) manoeuvrability is employed, which refers to the ability of a ship to maintain or change its motion states under the operation of actuators, such as propellers, rudders, and thrusters [49]. For digital twin applications, numerical simulations of ship manoeuvring can be employed using either system-based methods that use mathematical models to express the manoeuvring motion as an ordinary differential equation according the Newton's second law [49] or computational fluid dynamics (CFD) methods that resolve the complex fluid-structure interactions by incorporating both viscous and rotational effects in the flow problem [50]. However, despite CFD simulations being considered complementary to the experimental methods due to their high-fidelity and growing computational power [51], their time-consuming aspect renders them impractical [52], especially when real-time or a significant number of simulations need to be conducted [53].

On the other hand, system-based methods are typically limited to three or four degrees of freedom (3DOF, 4DOF) [54] with their main advantage being the rapid computational time, despite their compromise on the accuracy for long term manoeuvring simulations [53]. For instance, a system-based ship manoeuvring simulation of 150 s can be executed in 50 s on a typical Intel Core i7 @3.2 GHz processor [53], whereas a CFD-based simulation of 19 s can take 225 h on an Intel Xeon E5-2680 v2 @2.8 GHz processor [55]. The two most widely used system-based models are the Abkowitz model [56], also known as the whole ship model, which expresses the hydrodynamic forces and moments acting on a ship as Taylor series, and the manoeuvring modelling group (MMG) model [57], also known as the modular model, which decomposes the forces and moments acting on different components, typically the ship hull, propeller, and rudder [58]. Compared to the Abkowitz model that considers the ship as a whole, the MMG model is based on first principles that carry explicit and clear physical meaning of each component including their interactions [59]. In addition, it has the capability to accommodate different actuators configurations, such as single-screw [60], twin-screw [61], and twin azimuth thrusters [62], integrate external disturbances, such as currents, waves, and wind, and be combined with other numerical, experimental, or empirical methods [59].

To describe the manoeuvring motions, two coordinate systems are typically used, which are the space-fixed coordinate system o_0–$x_0y_0z_0$ and the body-fixed coordinate system o_B–$x_By_Bz_B$ [63], as presented in Figure 3. The origin o_B is assigned at the midship instead at the centre of gravity G, when a fixed position is needed regardless of the load condition. The x_0y_0-plane and x_By_B-plane coincide with the still-water surface plane, whereas the x_B-axis is

directed towards the bow and the z_0-axis and z_B-axis are directed downwards. u and v denote the surge and sway velocities in the x_B-direction and y_B-direction, respectively. $\dot{\psi}$ denotes the yaw rate around z_B-axis, where ψ denotes the heading angle from x_0-axis to x_B-axis. $\dot{\phi}$ denotes the roll rate, where ϕ denotes the roll angle. U denotes the resultant velocity as $U = \sqrt{u^2 + v^2}$, and β denotes the drift angle as $\beta = \tan^{-1}(-v/u)$.

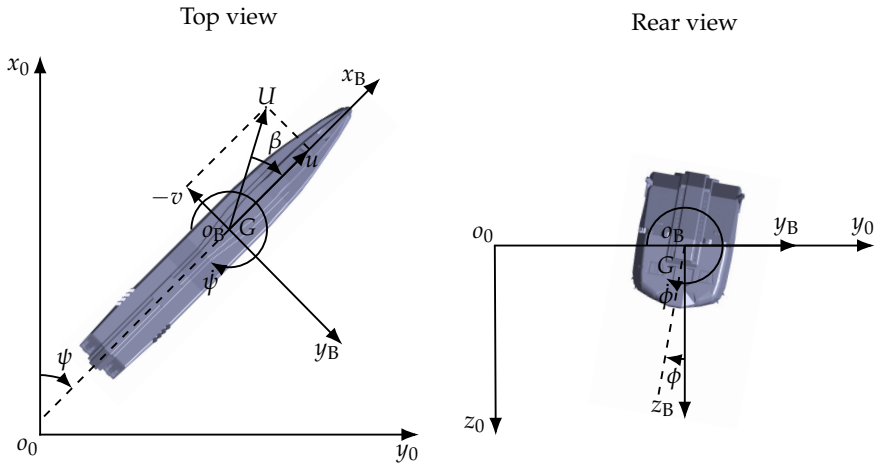


Figure 3. Top and rear views of the space-fixed coordinate system and body-fixed coordinate systems o_0–$x_0y_0z_0$ and o_B–$x_By_Bz_B$, respectively.

The motion equations of a 4DOF MMG that couples the surge, sway, yaw, and roll motions, assuming that the heave and pitch motions are negligible, is expressed as [64]:

$$\begin{cases} (m + m_{x_B})\dot{u} - (m + m_{y_B})v\dot{\psi} = X \\ (m + m_{y_B})\dot{v} + (m + m_{x_B})u\dot{\psi} + m_{y_B}a_{y_B}\ddot{\psi} - m_{y_B}l_{y_B}\ddot{\phi} = Y \\ (I_{z_B} + J_{z_B})\ddot{\psi} + m_{y_B}a_{y_B}\dot{v} = N - x_{B,G}Y \\ (I_{x_B} + J_{x_B})\ddot{\phi} - m_{y_B}l_{y_B}\dot{v} - m_{x_B}l_{x_B}u\dot{\psi} = K - W\overline{GM}\phi \end{cases} \quad (1)$$

where m, m_{x_B}, and m_{y_B} denote the ship's mass and the added mass in the x_B-direction and y_B-direction, respectively. I_{x_B}, I_{z_B}, J_{x_B}, and J_{z_B} denote the moment of inertia about the x_B-axis and z_B-axis, and the added moment of inertia in their corresponding axes, respectively. a_{y_B} denotes the x_B-coordinate of centre of m_{y_B}, while l_{x_B} and l_{y_B} denote the z_B-coordinates of the centres of m_{x_B} and m_{y_B}, respectively. $x_{B,G}$ denotes the x_B-coordinate of centre of gravity, W denotes the weight of water displaced by the ship hull, and $\overline{GM}$ denotes the metacentric height. X, Y, N, and K denote the forces in the x_B-axis and y_B-axis, yaw moment about the midship, and roll moment about the centre of gravity, respectively.

2.1.2. Perception Sensor

The digital twin of the perception sensor is employed considering its fundamental measurement principle and its noise model. During reactive collision avoidance, where the temporal and spatial margins for error are limited, perception sensors that provide distance measurements from collision-inducing obstacles become essential [65], such as stereo-cameras, RADARs, and LIDARs. Stereo-cameras passively capture electromagnetic waves using two pairs of monocular-cameras and provide the cheapest option when it comes to extracting distance measurements [14]. However, they require complex and heavy image processing techniques to extract such information [66] and exhibit the lowest accuracy depending on the illumination conditions [67].

RADARs and LIDARs actively emit electromagnetic waves in the microwave and infrared spectrum, respectively, and use the time-of-flight principle of the pulses reflected from the obstacles to directly measure the distances [68]. RADARs are suitable for long distance measurements of up to 10 NM and operation in all maritime environments [69].

However, the scanning frequency is low, approximately every 2–3 s, the measurements are limited only to 2 dimensions (2D), and the resolution and accuracy are low even at close distances, ranging from several to hundreds of m, due to the relatively wide beams [70,71].

In contrast, LIDARs exhibit the highest accuracy of distance and angular measurements due to their very narrow laser beams [15]. Scanning is conducted multiple times per s with each spin consisted of a few million measurements, allowing accurate distance, speed, position, and size estimation of the obstacles, as well as generation of a 3D map of the surrounding environment [70]. Typical LIDARs measure distances of up to a few hundreds m with some reaching up to a few km [69,72].

However, LIDARs are particularly susceptible to noise coming from the maritime environment, including sea water, rain, fog, and snow [70]. Water droplets in the atmosphere cause severe attenuation of the laser beams, absorb their energy, reduce the surface reflectivity of the obstacles, or cause pseudo-points from beams backscattered from the droplets that can be falsely detected as obstacles [73]. In addition, random noise can be produced by the sensor itself due to its intrinsic design limitations, discharges in the circuits, or during the conversion from analogue to digital signal [74]. Due to the many contributing factors that are hard to be known apriori [74], a common approach is to assume that the noise follows a Gaussian distribution [75], as presented in Figure 4. Hence, the time-of-flight equation of a LIDAR sensor with an added Gaussian noise model is expressed as [76]:

$$d_i = \frac{ct_i}{2} + \epsilon \tag{2}$$

where d_i denotes the distance of the ith LIDAR measurement, c denotes the speed of light, t_i denotes the time of flight, and ϵ denotes the added noise that follows a Gaussian distribution with 0 mean and variance σ^2. The total LIDAR measurements at each timestep t constitutes to the array $d_t = [d_{1,t}, d_{2,t}, \ldots]$.

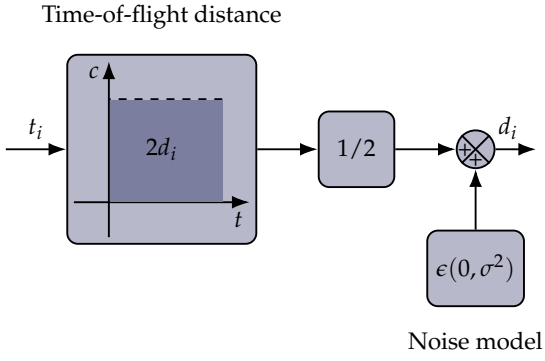


Figure 4. Block diagram of a LIDAR time-of-flight measurement principle and its added noise model.

2.1.3. Map

The digital twin of the map is employed, which stands for the spatial representation of the environment [77], such as the navigational area of the OS. Typical map representation methods are divided into topological maps, feature maps, and occupancy grid maps [78]. Topological maps use graphs, such as nodes and arcs, to represent the environment in the most compact way, but exhibit challenges in storing proximity information or modelling complex environments due to their simplicity [79]. Feature maps use parametric features, such as points and lines, to represent distinctive parts of the environment that are identifiable by the perception sensor, but they are not suitable for highly unstructured environments with no distinct geometries and when detailed navigation is needed due to its limited resolution [78].

Occupancy grid maps use grid of fixed resolution to represent the environment, where each cell stores a probability of being free or obstacle-occupied [80]. Their main advantages are their rich representation of the environment in both 2D and 3D, flexibility in balancing resolution with accuracy depending on the application requirements, and

compatibility with noisy perception sensors, such as a LIDAR [81]. Thus, occupancy grid maps are commonly used to represent the navigational areas for path planning and collision avoidance applications [82], as presented in Figure 5.

(**a**) (**b**)

Figure 5. (**a**) Satellite image of a navigational area captured using Google Maps. (**b**) 2D occupancy grid map of the navigational area.

2.2. Decision-Making Problem Formulation

The objective of this phase is to formulate the decision-making problem pertaining to the investigated scenarios. Specifically, this falls into the category of complex sequential decision-making problems under uncertainty, which are typically formulated using a mathematical framework known as the Markov decision process (MDP) [83]. In fact, many real-world decision-making problems can be modelled as an MDP due to its abstract and flexible framework [84]. The fundamental idea of this framework is to achieve a goal by learning from interactions, in which the learning decision-maker is known as the agent and everything else that interacts with it as the environment $\mathcal{E}$.

The interaction at each timestep t is boiled down into the agent receiving a state of the environment S_t and selecting an action A_t based on the received state that leads to the reception of a reward R_{t+1} and new state S_{t+1} [85], as presented in Figure 6. The sum of all rewards that the agent accumulates from timestep t and onwards through this interaction is known as the return G_t, which is expressed as:

$$G_t = R_t + R_{t+1} + R_{t+2} + \dots \tag{3}$$

However, considering the uncertainties of the future, the return is formulated as the expected discounted return, which is expressed as:

$$G_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \dots = \sum_{i=t}^{\mathcal{T}} \gamma^{i-t} R_i \tag{4}$$

where $\gamma = [0, 1]$ denotes the discount rate, i denotes the timestep $i = [0, \mathcal{T}]$, and $\mathcal{T}$ denotes the terminal timestep.

The overall goal of the agent is to learn a policy π that maximises the expected return from its initial timestep G_0 by mapping from states to actions. However, considering that the policy can be stochastic, the return G_0 is formulated as an expected return J, which is expressed as:

$$J = \mathbb{E}_{r_i, s_i \sim \mathcal{E}, a_i \sim \pi}[G_0] \tag{5}$$

where, s_i, a_i, and r_i denote the values of the state, action, and reward, respectively.

To learn such a policy, the expected return at each timestep t and onwards after taking the action a_t in state s_t following the policy π needs to be learned, known as the action-value function $Q^\pi(s_t, a_t)$, which is expressed as:

$$Q^\pi(s_t, a_t) = \mathbb{E}_{r_{i\geq t}, s_{i>t} \sim \mathcal{E}, a_{i>t} \sim \pi}[G_t \mid s_t, a_t] \tag{6}$$

The action-value function can be formulated in its recursive form using the Bellman equation, which is expressed as:

$$Q^\pi(s_t, a_t) = \mathbb{E}_{r_t, s_{t+1} \sim \mathcal{E}}\left[r(s_t, a_t) + \gamma \mathbb{E}_{a_{t+1} \sim \pi}[Q^\pi(s_{t+1}, a_{t+1})]\right] \tag{7}$$

Finally, the action-value function can be further simplified when a deterministic policy μ is used, which is expressed as:

$$Q^\mu(s_t, a_t) = \mathbb{E}_{r_t, s_{t+1} \sim \mathcal{E}}[r(s_t, a_t) + \gamma Q^\mu(s_{t+1}, \mu(s_{t+1}))] \tag{8}$$

Thus, the goal of the agent is to learn an optimal policy by accurately estimating the optimal action-value function at each timestep, thereby maximising the expected return.

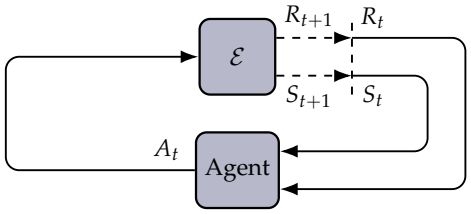


Figure 6. The agent–environment interaction in MDP.

2.2.1. Rewards

The rewards that the agent learns to maximise are defined as reward functions, which are identified according to path following, nominal navigation, actuator control, and collision avoidance objectives. The path following reward is expressed as:

$$R_{1,t} = \begin{cases} k_1 + \dfrac{k_2}{e_{\mathrm{XT},t}^2 + k_3} + \dfrac{k_4}{e_{\mathrm{H},t}^2 + k_5 + k_6|e_{\mathrm{XT},t}|}, & \text{if } |e_{\mathrm{XT},t}| \leq \mathrm{XTL} \\ \dfrac{k_2}{e_{\mathrm{XT},t}^2 + k_3} + \dfrac{k_4}{e_{\mathrm{H},t}^2 + k_5 + k_6|e_{\mathrm{XT},t}|}, & \text{if } |e_{\mathrm{XT},t}| > \mathrm{XTL} \end{cases} \tag{9}$$

where $k_1, k_2, k_3, k_4, k_5, k_6 > 0$ denote the coefficients, e_H denotes the heading error of the OS from the path as $e_\mathrm{H} = \psi_\mathrm{WP} - \psi$ given the heading of the path $\psi_\mathrm{WP} = \arctan(y_2 - y_1, x_2 - x_1)$ from two waypoints (x_1, y_1) and (x_2, y_2), e_XT denotes the cross-track error of the OS from the path as $e_\mathrm{XT} = -(x - x_1)\sin(\psi_\mathrm{WP}) + (y - y_1)\cos(\psi_\mathrm{WP})$ [86], and XTL denotes the threshold for the path following criterion often refer to as the cross-track limit [87]. It can be inferred that the reward is maximised when the cross-track and heading errors of the OS from the path are minimised concurrently. Also, an additional constant reward k_1 is given when the OS follows the path within the given threshold.

The nominal navigation reward is expressed as:

$$R_{2,t} = k_7 v_t^2 + k_8 \dot{\psi}_t^2 + k_9 \dot{v}_t^2 + k_{10} \ddot{\psi}_t^2 \tag{10}$$

where $k_7, k_8, k_9, k_{10} < 0$ denote the coefficients. It can be inferred that the reward is maximised when the sway and yaw velocities and accelerations of the OS are minimised.

The actuator control reward is expressed as:

$$R_{3,t} = k_{11}\delta_{\mathrm{C},t}^2 + k_{12}\dot{\delta}_{\mathrm{C},t}^2 \tag{11}$$

where $k_{11}, k_{12} < 0$ denote the coefficients, δ_C and $\dot{\delta}_C$ denote the magnitude and rate of the action command executed by the agent. It can be inferred that the reward is maximised when the magnitude and the rate of the actuator are minimised.

The collision avoidance reward is expressed as:

$$R_{4,t} = k_{13} d_t + k_{14}, \text{ if } d_t > d_{\min} \tag{12}$$

where $k_{13} > 0$ and $k_{14} < 0$ denote the coefficients, and $d_{\min}$ denotes the threshold for the collision criterion. It can be inferred that the reward is maximised when the distances of the OS from the obstacles are maximised.

The final reward function is expressed as:

$$R_t = R_{1,t} + R_{2,t} + R_{3,t} + R_{4,t} \tag{13}$$

2.2.2. States & Actions

The states that the agent receives from the environment are identified as the set of variables related to the rewards, known as observations, which constitutes the state space [85]. A low-dimensional state space that consists of a limited number of observations can lead to an incomplete understanding of the environment and sub-optimal decision-making, whereas a high-dimensional state space can exponentially increase the computational complexity and training time of the agent [88]. Thus, the state space is expressed as:

$$S_t = [\psi_t, u_t, \dot{u}_t, v_t, \dot{v}_t, r_t, \dot{r}_t, U_t, \dot{U}_t, e_{\mathrm{XT},t}, \dot{e}_{\mathrm{XT},t}, \ddot{e}_{\mathrm{XT},t}, e_{\mathrm{H},t}, \dot{e}_{\mathrm{H},t}, \ddot{e}_{\mathrm{H},t}, d_t, \mathcal{N}] \tag{14}$$

It is worth noting that despite noise being added on the perception sensor measurements observation d, an additional noise variance observation $\mathcal{N}$ is added to represent the value of the noise variance σ^2 of the perception sensor, as presented prior in Equation (2).

The actions that the agent executes to affect the state of the environment and maximise the rewards are identified as the set of all possible actions that the actuator can take, which constitutes the action space [85]. Thus, the action space is expressed as:

$$A_t = \delta_{C,t} \tag{15}$$

2.3. Agent Training

The objective of this phase is to train an agent that can make decisions over the formulated problem, by developing a training framework. Recently, DRL methods have gained significant attention due to the major breakthroughs achieved in decision-making, by integrating the trial-and-error techniques of reinforcement learning and feature extraction capabilities of deep learning using artificial neural networks [89]. Specifically, DRL methods have surpassed previous limitations of addressing complex decision-making problems with high-dimensional state and action spaces [90] with significant end-to-end learning [91] at a human level [92,93]. The DRL training framework, as delineated in this Section, is developed in MATLAB/Simulink 2023b, due to the customisation capabilities of a wide range of machine learning libraries, including the Deep Learning Toolbox [94] and Reinforcement Learning Toolbox [95].

2.3.1. Deep Reinforcement Learning Agent

A DRL agent is employed, where some of the most commonly used algorithms are the trust region policy optimisation [96], deep deterministic policy gradient (DDPG) [97], proximal policy optimisation [98], soft actor-critic [99], and twin delayed DDPG (TD3) [100]. While each algorithm has its own advantages and disadvantages, choosing the most suitable one for each application is not a trivial effort. However, a comparative study evaluating the aforementioned algorithms for process control applications provides a good indication of the superior control performances of DDPG and TD3 algorithms [101]. In addition, DDPG algorithm is suitable for continuous and high-dimensional state and action spaces and challenging physical control problems with inertia and fine control of actions [102].

The DDPG algorithm is based on an actor–critic architecture, where the actor and the critic represent the policy $\mu(s \mid \theta^{\mu})$ and action-value function $Q(s, a \mid \theta^{Q})$, respectively, using deep neural networks (DNNs) as function approximators parameterised by their respective weights θ^{μ} and θ^{Q}. The actor and critic are referred to as the main actor network and main critic network, respectively, that constitute the main network. The actor and critic learn off-policy by utilising a replay buffer $\mathcal{D}$ of finite size that stores the tuples (s_t, a_t, r_t, s_{t+1}) generated from the environment at each timestep t. Particularly, the actor and critic are updated by uniformly sampling a minibatch $\mathcal{B}$ of tuples (s_i, a_i, r_i, s_{i+1}) from the replay buffer to minimise the correlations between the samples, where $i = [1, b]$ and b denotes the size of the minibatch.

Based on the prior Equation (7), the targets for the learning of the critic is expressed as:

$$y_i = r_i + \gamma Q(s_{i+1}, \mu(s_{i+1} \mid \theta^{\mu}) \mid \theta^{Q}) \tag{16}$$

However, since the action-value function is updated while in the target, it leads to divergence in learning. To tackle this, DDPG algorithm employs a target network, which is a copy of the main network. Specifically, the target actor network $\mu'(s \mid \theta^{\mu'})$ and target critic network $Q'(s, a \mid \theta^{Q'})$ with their respective weights $\theta^{Q'}$ and $\theta^{\mu'}$ are used to give stable targets, which is expressed as:

$$y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1} \mid \theta^{\mu'}) \mid \theta^{Q'}) \tag{17}$$

Thus, the main critic network learns by minimising the critic loss, which is expressed as:

$$L(\theta^{Q}) = \frac{1}{N} \sum_{i=1}^{N} (y_i - Q(s_i, a_i \mid \theta^{Q}))^2 \tag{18}$$

and by updating its weights in the direction of gradient descent that minimises the loss, which is expressed as:

$$\theta^{Q} \leftarrow \theta^{Q} - a_Q \nabla_{\theta^{Q}} L \tag{19}$$

where a_Q denotes the learning rate of the main critic network.

The main actor network learns by minimising the actor loss, which is expressed as:

$$J(\theta^{\mu}) = -\frac{1}{N} \sum_{i=1}^{N} Q(s_i, \mu(s_i \mid \theta^{\mu}) \mid \theta^{Q}) \tag{20}$$

and by updating its weights in the direction of gradient ascent that maximises the expected return, which is expressed as:

$$\theta^{\mu} \leftarrow \theta^{\mu} + a_{\mu} \nabla_{\theta^{\mu}} J \tag{21}$$

where a_{μ} denotes the learning rate of the main actor network.

The target actor network and target critic network learn by updating their weights using soft target updates, which is expressed as:

$$\theta^{Q'} \leftarrow \tau \theta^{Q} + (1 - \tau) \theta^{Q'} \tag{22}$$

$$\theta^{\mu'} \leftarrow \tau \theta^{\mu} + (1 - \tau) \theta^{\mu'} \tag{23}$$

where $\tau \ll 1$ denotes the target smooth factor.

It is worth noting that every tuple of the replay buffer are normalised for effective learning, known as batch normalisation. Also, noise sampled from a noise process $\mathcal{W}$, such as an Ornstein-Uhlenbeck process, is added on the main actor network's output to encourage exploration. A pseudo-code and schematic diagram of the algorithm are presented in Algorithm 1 and Figure 7.

Algorithm 1: Deep deterministic policy gradient

Initialise $Q(s, a \mid \theta^Q)$ with random θ^Q and $Q'(s, a \mid \theta^{Q'})$ with $\theta^{Q'} \leftarrow \theta^Q$;
Initialise $\mu(s \mid \theta^\mu)$ with random θ^μ and $\mu'(s \mid \theta^{\mu'})$ with $\theta^{\mu'} \leftarrow \theta^\mu$;
Initialize $\mathcal{D}$;
for *Episode* $= 1, \ldots$ **do**
  Initialize $\mathcal{W}$;
  Receive s_1;
  for $t = 1, \mathcal{T}$ **do**
    Set $a_t = \mu(s_t \mid \theta^\mu) + \mathcal{W}_t$;
    Execute a_t, receive r_t and s_{t+1};
    Store (s_t, a_t, r_t, s_{t+1}) in $\mathcal{D}$;
    Sample $\mathcal{B}$ of (s_i, a_i, r_i, s_{i+1}), $i = 1, \ldots, b$ from $\mathcal{D}$;
    Set $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1} \mid \theta^{\mu'}) \mid \theta^{Q'})$;
    Calculate $\nabla_{\theta^Q} L = \frac{1}{b} \sum_{i=1}^{b} \nabla_{\theta^Q} (y_i - Q(s_i, a_i \mid \theta^Q))^2$;
    Calculate $\nabla_{\theta^\mu} J = \frac{1}{b} \sum_{i=1}^{b} \nabla_a Q(s_i, a \mid \theta^Q) \mid_{a=\mu(s_i \mid \theta^\mu)} \nabla_{\theta^\mu} \mu(s_i \mid \theta^\mu)$;
    Update $\theta^Q \leftarrow \theta^Q - a_Q \nabla_{\theta^Q} L$;
    Update $\theta^\mu \leftarrow \theta^\mu + a_\mu \nabla_{\theta^\mu} J$;
    Update $\theta^{Q'} \leftarrow \tau\theta^Q + (1 - \tau)\theta^{Q'}$;
    Update $\theta^{\mu'} \leftarrow \tau\theta^\mu + (1 - \tau)\theta^{\mu'}$;
  end
end

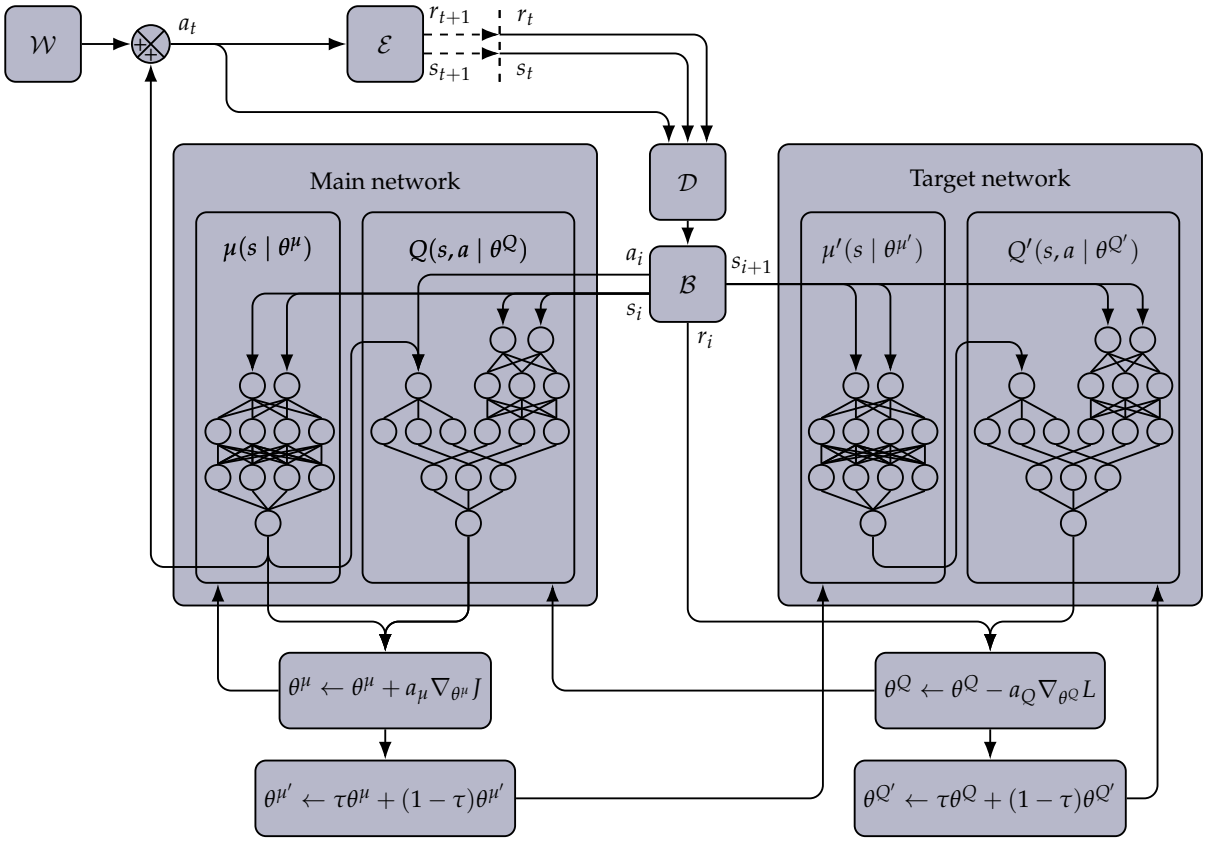


Figure 7. Block diagram of the DDPG algorithm.

2.3.2. Training Framework

A training framework is developed considering the noise levels envelope of the perception sensor investigated during the agent's training. Specifically, the noise variance

σ^2 of the perception sensor are generated from uniformly distributed random numbers between a predefined minimum and maximum value, the range of which constitutes the noise levels envelope. A single noise variance is used during each episode until the termination of that episode, where an episodes stands for the sequence of timesteps of an episodic task from the beginning to the end [85]. It is worth noting that the reward functions R_t and thresholds for the path following and collision criteria XTL and $d_{\min}$, respectively, remain the same, regardless of whether the noise variance is low or high.

Each episode is automatically terminated when the OS surpasses the boundaries of the map or when the timestep reaches the predefined terminal timestep $\mathcal{T}$. In addition, the episode is terminated when the collision criterion is triggered. The training performance of the agent is evaluated based on different training performance metrics, including the return per episode and the loss per step, where step refers to the update iteration. Particularly, the overall trends of these metrics towards a local maximum or local minimum and reduction of their variances are considered as good indicators of convergence and stability of training. Finally, the training is conducted in parallel using 30 cores of an Intel Xeon Platinum 8260 CPU processor for the episodes simulation and a NVIDIA Quadro RTX 8000 GPU processor for the update of the agent's learnable parameters.

2.4. Robustness Quantification & Verification

The objective of the robustness quantification phase is to quantify the robustness of the trained agent's decision-making against various perception sensor noise levels, by defining robustness metrics. Robustness stands for the magnitude of tolerability of a system against disturbances until a set requirement or threshold is violated [103,104]. A robustness metric provides a binary assessment of system's robustness state based on the compliance or violation of the threshold. In addition, it quantifies the degree of robustness by measuring the available margin towards violating the threshold.

Two robustness metrics are defined pertaining to path following and collision avoidance $RM_\times$ and RM_*, respectively, which are expressed as:

$$RM_\times = |e_{\mathrm{XT},\times}| < \mathrm{Threshold}_\times \tag{24}$$

$$RM_* = d_* > \mathrm{Threshold}_* \tag{25}$$

where $\times$ and $*$ denote the timesteps of the first and minimum obstacle detection, respectively, $\mathrm{Threshold}_\times$ denotes the robustness threshold for path following, and $\mathrm{Threshold}_*$ denotes the robustness threshold for collision avoidance. It can be inferred that the robustness for path following requires cross-track error within a margin prior to the first obstacle detection, whereas the robustness for collision avoidance requires sufficient distance from the obstacles at the minimum obstacle detection. It is worth noting that the robustness thresholds are not required to be equal to the thresholds for path following and collision criteria XTL and $d_{\min}$, respectively, depending on the robustness requirements.

Finally, the objective of the robustness verification phase is to verify the robustness of trained agent's decision-making, by investigating various scenarios. Specifically, scenarios within and outside the trained noise levels envelope are investigated, until either of the robustness thresholds for path following or collision avoidance are violated.

3. Case Study

The case study considers a single-screw high-speed container ship as the OS, known as the S-175 [64]. The S-175 is selected, as it is one of the benchmark hull forms used in manoeuvring studies, due to its well-defined hydrodynamic coefficients [105]. A 4DOF MMG model is used, as released in the Marine Systems Simulator Toolbox [106], that considers the none-negligible roll-coupling effect in high-speed container ships with low metacentric height [107], such as the S-175. In addition, the model considers no environmental disturbances and it is assumed to provide sufficient fidelity, as 3DOF is often sufficient in most manoeuvring studies [108].

A first-order linear differential equation model is used to simulate the rudder dynamics, where the deflection rate of the rudder $\dot{\delta}$ is defined as $\dot{\delta} = (\delta_C - \delta)/T_\delta$ given the commanded rudder angle δ_C, rudder angle δ, and time constant T_δ that expresses the time delay due to the main servo in linear manners [109,110]. Two slew rates are used to saturate the maximum commanded rudder angle as $|\delta_C| \leq \delta_{C,max}$ and the maximum deflection rate of the rudder as $|\dot{\delta}| \leq \dot{\delta}_{max}$ [111]. In addition, a constant nominal propeller revolution n_{nom} is considered to operate at 85% load of the candidate engine's specified maximum continuous rating (SMCR). The main particulars of the OS are presented in Table 1.

Table 1. Main particulars of the OS.

Particulars	Symbol	Value
Length	L	175 m
Beam	B	25.4 m
Draft	T	8.5 m
Depth	D	11.0 m
Displaced volume	∇	21,222 m^3
Block coefficient	c_B	0.559
Maximum commanded rudder angle [1]	$\delta_{C,max}$	10 deg
Maximum rudder deflection rate [2]	$\dot{\delta}_{max}$	5 deg/s
Time constant [3]	T_δ	1 s
Nominal propeller revolution at 85% SMCR [4]	n_{nom}	99.5 rpm
Nominal speed	U_{nom}	10.41 m/s

[1] It should not exceed 35 deg according to the International Convention for the Safety of Life at Sea (SOLAS), Regulation 29 [112]. A relatively small maximum commanded rudder angle is selected to reduce the roll effect [64]. [2] It should exceed 2.32 deg/s according to the SOLAS, Regulation 29 [112]. [3] Typical values range between 1–3 s [113]. [4] For a candidate marine two-stroke engine, such as the 7S60ME-C10, that satisfies the power requirements of SMCR between 11,200–15,000 kW for a container ships with deadweight tonnage between 26,500–33,000 mt [114].

A 2D binary occupancy map is considered, assuming that in most surface-based applications the most critical information to be perceived is on the horizontal plane [115]. The size of the map is 5 × 5 km with resolution of 1 × 1 m, where each cell stores a binary value, 0 or 1, to represent a free or obstacle-occupied area, respectively. It is worth noting that collision with static obstacles remains a challenging issue as global databases of maritime accidents suggest that 21% of 513 investigated collision accidents originated from strikes with other ships that were not underway using their engines, such as being engaged in fishing, moored, or at anchor [116]. For this reason, the case study considers a single static obstacle on the map that the OS encounters during the following of a straight global path generated by two waypoints. The static obstacle is modelled as a circle of 100 m radius to arbitrarily approximate the shape and size of any mid-sized ship, similar to the OS size.

A 2D LIDAR is considered, which is set to scan at the midship with a maximum detecting range of 1341 m. The area that exceeds this range constitutes the sensor's blind-spot zone, where no information can be derived from [117]. It is worth noting that the selected maximum detecting range is not arbitrary but it is comparative to the manoeuvrability of the OS. Specifically, assuming that the turning circle at the maximum rudder angle is a good indicator of the OS manoeuvrability [118], the ratio of the maximum detecting range to its maximum advance is 1.4. In addition, a forward scanning with 225 deg field of view is assumed to be sufficient among static obstacles, where an angular resolution of 5.63 deg constitutes to a total of 41 LIDAR measurements that guarantees the detection of any circular obstacle with radius greater than 65.8 m at its maximum detecting range.

A representation of investigated case study is presented in Figure 8. In each episode, the OS is always initiated at the first waypoint and at nominal navigation conditions heading towards the next waypoint. The obstacle is allocated on the path with its vertical position randomly generated between $y_{obs} = [2000, 3500]$ m. In addition, the noise variance

of the LIDAR is randomly generated between $\sigma^2 = [0, 25]$ m^2. The thresholds for the path following and collision criteria are set as XTL $= B/2$ and $d_{\min} = B/2$, respectively. The robustness thresholds for path following and collision avoidance as set as Threshold$_\times$ = XTL and Threshold$_*$ $= d_{\min}$, respectively. A DDPG agent is employed, whose main hyperparameters pertaining to the network's architecture are two hidden layers with 600 neurons each for the actor network and two hidden layers with 500 neurons each for the critic network constituting to a total of 396,001 and 281,001 learnable parameters including their weights, respectively. Finally, the simulation timestep and update iteration step of the agent are set to 1 s to be compatible with real-time applications.

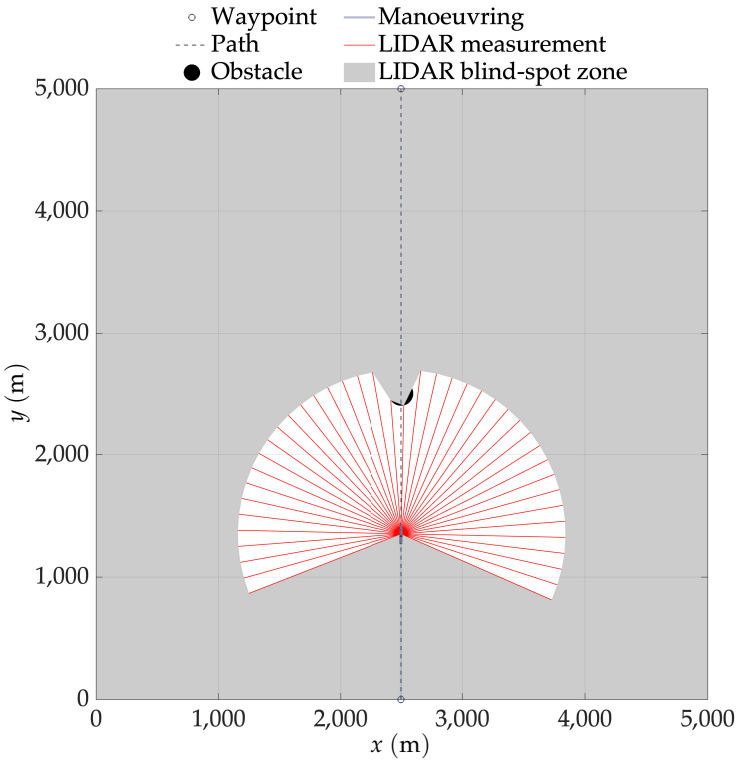


Figure 8. Representation of the investigated case study.

4. Results & Discussion

4.1. Agent Training

The training performance of the agent$_A$ in terms of the return per episode, actor loss per step, and critic loss per step is presented in Figure 9. It is noted that the return convergences to a local maximum approximately from episode 4800, as indicated by its moving median, with further reduction of its variance from episode 5500, which denotes that the agent consistently maximises the return in each episode. The actor loss converges to a local minimum from step 2.25×10^6, as indicated by its moving median, which denotes that the actor converges to a stable policy. In contrast, the critic loss diverges from a local minimum from step 1.75×10^6, as indicated by its moving median, with a general increase of its variance due to exploding values, which suggests that the critic does not converge to a stable estimation of the expected return. However, despite this instability, the critic loss exhibits a downwards trend potentially towards a local minimum from step 2.3×10^6, as indicated by its moving median, which is in alignment to the overall convergence trends of the other training performance metrics.

The training is terminated at episode 5820, which constitutes to a total of 2,529,998 steps and approximately 44 h of training. The minimum and maximum values of the training performance metrics derived during the training and their respective moving medians at the end of the training are presented in Table 2. The noise variances σ^2 and vertical positions

y_{obs} generated randomly during the training in terms of their distributions are presented in Figure 10. The mean values of their distributions $\overline{\sigma^2} = 12.40$ m^2 and $\overline{y}_{obs} = 2745.70$ m denote that the agent is trained effectively from diverse scenarios of uniformly distributed random values.

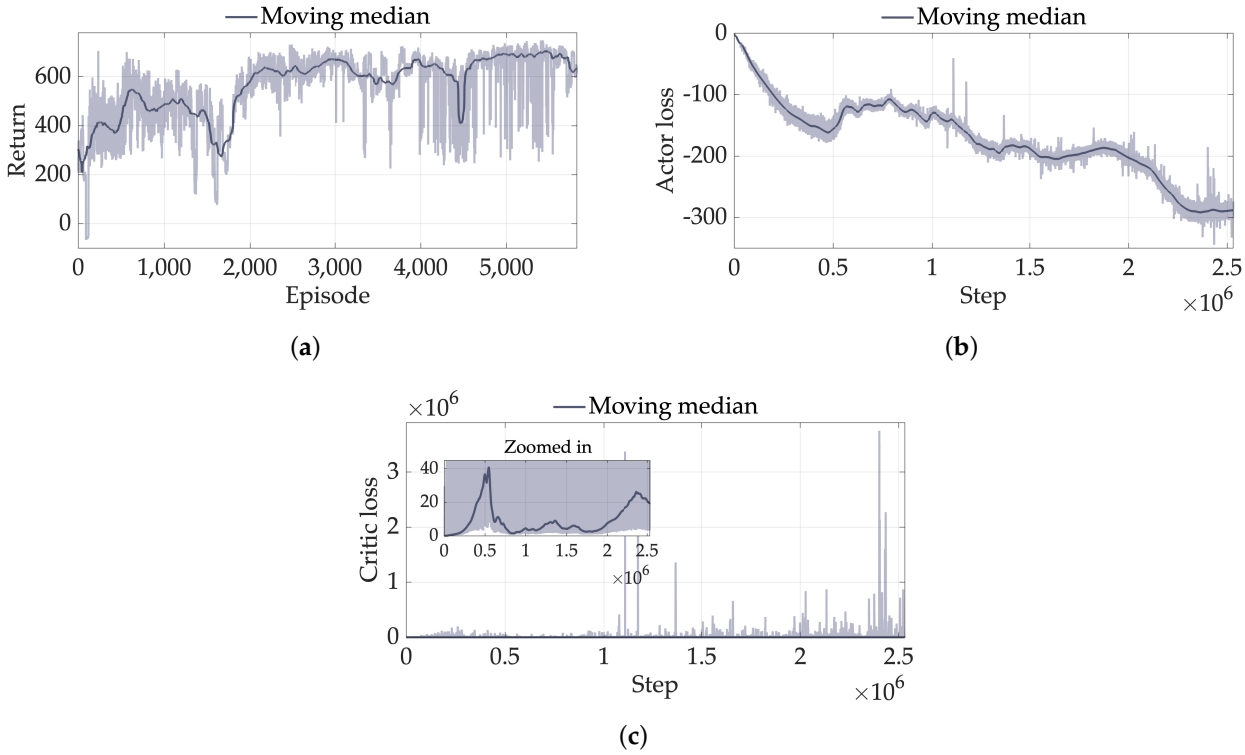


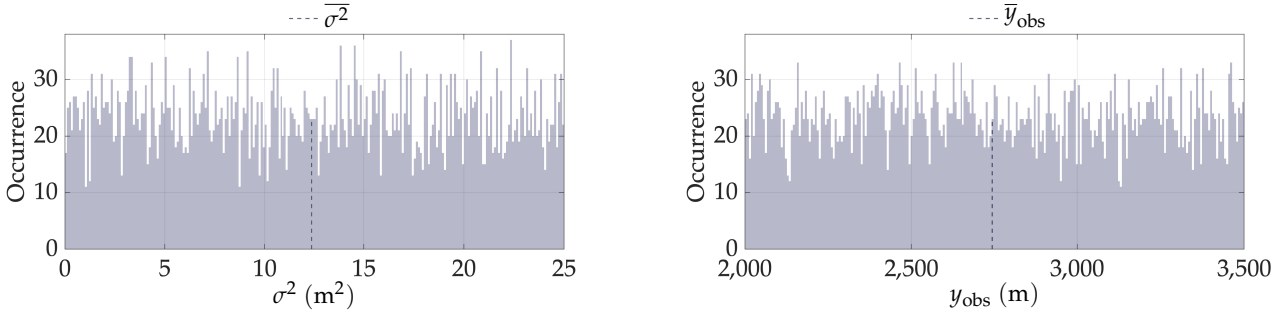
Figure 9. (**a**) Return per episode, (**b**) actor loss per step, and (**c**) critic loss per step during the training of agent$_A$. The sliding windows of their respective moving medians are 150 episodes and 25,000 steps.

Table 2. Training performance metrics during the training of agent$_A$.

Return$_{A,min}$	**Return$_{A,max}$**	**Moving Median$_A$** [1]
−67.42	749.72	633.84
Actor loss$_{A,min}$	**Actor loss$_{A,max}$**	**Moving Median$_A$** [2]
−344.62	−0.45	−288.36
Critic loss$_{A,min}$	**Critic loss$_{A,max}$**	**Moving median$_A$** [2]
2.10×10^{-3}	3,750,911	19.39

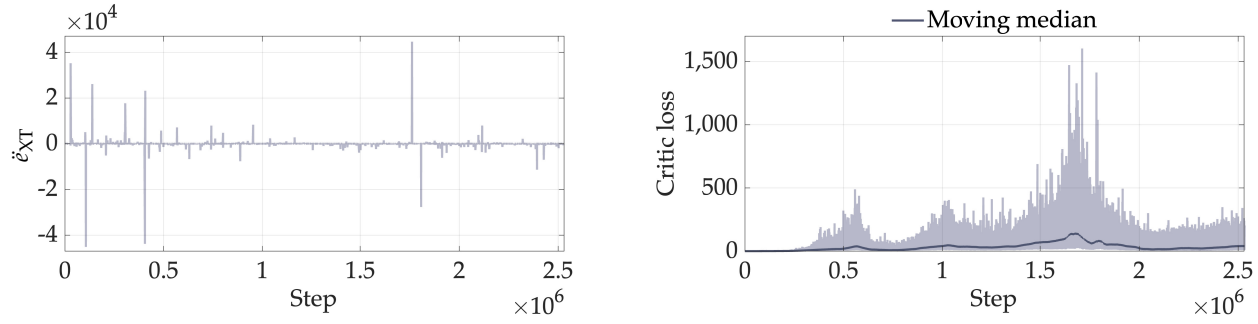
[1] At episode 5820. Sliding window of 150 episodes. [2] At episode 5820. Sliding window of 25,000 steps.

The exploding values of the critic loss can be attributed to the poor observations normalisation or issues related to numerical differentiation. Specifically, some observations are found to explode during the training, such as the $\ddot{e}_{XT}$, as presented in Figure 11. Retraining the agent$_A$, while using hard limiters on these observations, improves the convergence of the critic loss. Similar effects can be derived by adjusting the timestep, using a different solver, or employing other differentiation techniques, but potentially at the expense of longer simulation periods. However, this is presented only as a critical discussion on the effect of exploding observations on the training performance. The initial settings are maintained as the optimisation of the observations normalisation and numerical differentiation exceed the scope of this study.



(**a**) (**b**)

Figure 10. Distributions of (**a**) random noise variances σ^2 and (**b**) random vertical positions y_{obs} generated during the training of agent$_{\text{A}}$. The number of bins for each histogram is 250.



(**a**) (**b**)

Figure 11. (**a**) Exploding values of the $\ddot{e}_{\text{XT}}$ observation during the training of agent$_{\text{A}}$. (**b**) Critic loss per step during the training of agent$_{\text{A}}$ using hard limiters on the exploding observations. The sliding window of the moving median is 25,000 steps.

4.2. Simulation Results

The trained agent$_{\text{A}}$ is simulated in various scenarios within the trained noise levels envelope, but for brevity, only the results pertaining to a single obstacle position on the middle of the path $x_{\text{obs}} = 2500$ m and $y_{\text{obs}} = 2500$ m and minimum and maximum noise variances $\sigma^2 = 0$ m^2 and $\sigma^2 = 25$ m^2, respectively, are presented. The agent's decision-making in terms of the conducted manoeuvring in each scenario is presented in Figure 12 and in Video S1. The results are derived 3.57 times faster than the defined simulation timestep, which suggests the efficiency of employing the agent in real-time applications. Also, the same random seeds are used for the reproducibility of the generated noises in the LIDAR measurements and consistency of the results. Finally, considering that the timesteps $\times$ and $*$ refer to the first and minimum obstacle detections, respectively, each manoeuvring simulation is decomposed into three distinct stages as path following manoeuvre, evasive manoeuvre, and path recovery manoeuvre.

The returns that the agent accumulates are 695.72 when $\sigma^2 = 25$ m^2 and 708.32 when $\sigma^2 = 0$ m^2, which suggest that the agent's decision-making is different in each scenario. Specifically, the decomposition of each return to its reward functions, as presented prior in Equation (13), indicates that R_1 corresponds to 74.9% of the difference from $\sigma^2 = 25$ m^2 to $\sigma^2 = 0$ m^2, followed by the R_4, R_2, and R_3 that correspond to -39.9%, 35.4%, and 29.6%, respectively. This implies that the higher accumulated return when the noise variance is lower is due to the agent's decision-making exploiting more the path following, nominal navigation, and actuator control rewards against the collision avoidance reward, which is manifested as smaller evasive manoeuvres. The opposite applies when the noise variance is higher resulting in larger evasive manoeuvres. This lends credence to the notion of

the agent's sophisticated decision-making, capable of prioritising safety over efficiency depending on the noise variance, which is desirable from safety perspective considering the increased uncertainties under higher noise variances and the limited temporal and spatial margins for error during reactive collision avoidance.

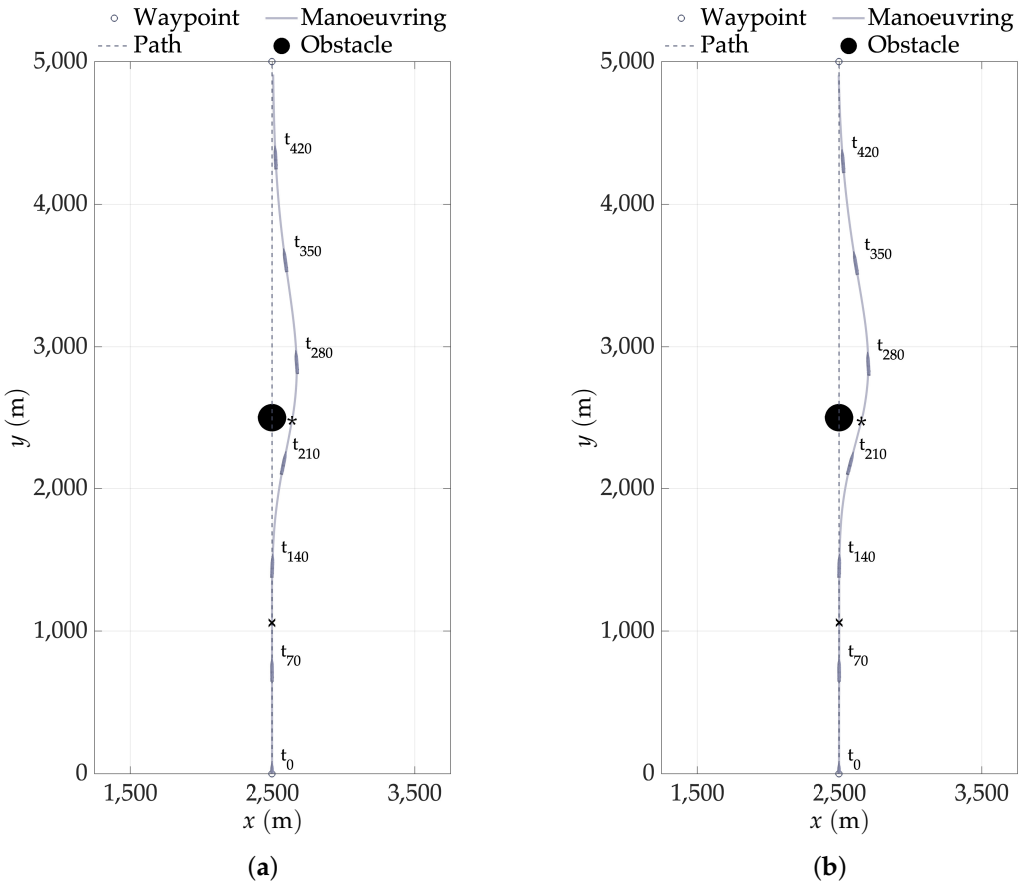
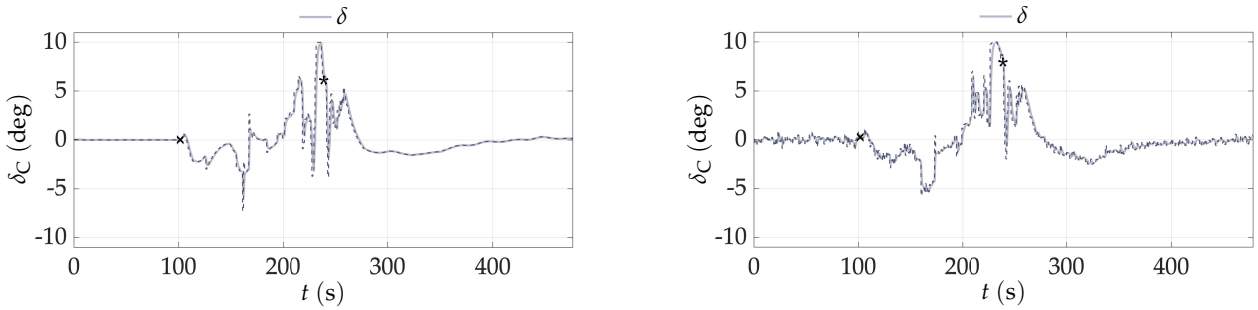


Figure 12. Decision-making of the agent$_A$ in terms of the conducted manoeuvring when (**a**) $\sigma^2 = 0$ m^2 and (**b**) $\sigma^2 = 25$ m^2. The position of the OS is plotted every 70 s. The $\times$ and $*$ represent the timesteps of the first and minimum obstacle detection, respectively.

The actor's policy in terms of the commanded rudder angle δ_C per timestep t is presented in Figure 13. It is noted that the commanded rudder angles become more erratic when $\sigma^2 = 25$ m^2, potentially due to the uncertainties of the noisy measurements. For instance, a LIDAR measurement less than its maximum detecting range clearly indicates a detection of an obstacle when $\sigma^2 = 0$ m^2, whereas the distinction between noisy LIDAR measurements and detection of an obstacle becomes more uncertain when $\sigma^2 = 25$ m^2. However, despite this uncertainty, the actor deflects the rudder angle to initiate an evasive manoeuvre only after the first obstacle detection, as indicated by the timesteps $\times$. This can be attributed to the great feature extraction capabilities of DNNs, enabling them to discern genuine obstacle detections amidst the noisy measurements.

In addition, it is noted that the actor's policy is inherently different in each scenario, not only in terms of its erratic output. Specifically, the absolute area under the commanded rudder angles is increased by 25.2% when $\sigma^2 = 25$ m^2 compared to when $\sigma^2 = 0$ m^2, where the normalised area per timestep for the evasive manoeuvre corresponds to 55.3% of the difference, followed by the path following and path recovery manoeuvres that correspond to 28.7% and 16.0%, respectively. In addition, the magnitude of the commanded rudder angles at the minimum obstacle detections are 7.76 deg when $\sigma^2 = 25$ m^2 and 5.97 deg $\sigma^2 = 0$ m^2, as indicated by the timesteps $*$. It can be derived that the prior finding on the

agent's sophisticated decision-making conducting different evasive manoeuvres depending on the noise variance is attributed to the difference between the commanded rudder angles, which is particularly evident during the evasive manoeuvres. This lends credence to the notion of the actor's expressiveness, stemming from its policy's ability to capture the complexity and richness across different noise variances and provide a different output, which is desirable considering the dynamic operating conditions affecting the stochastic nature of noise differently.



(**a**) (**b**)

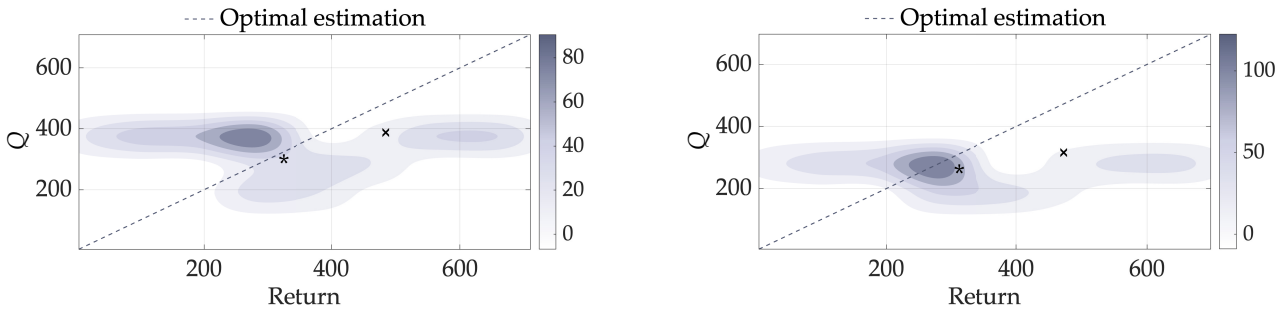
Figure 13. Actor's policy of the agent$_A$ in terms of the commanded rudder angle δ_C per timestep t when (**a**) $\sigma^2 = 0$ m^2 and (**b**) $\sigma^2 = 25$ m^2. The $\times$ and $*$ represent the timesteps of the first and minimum obstacle detection, respectively.

The critic's estimation in terms of the expected return Q-value per return is presented in Figure 14. It is worth noting that despite the role of the critic's estimation on the actor's policy improvement taking place only during the training of the agent, the investigation of the Q-value can give meaningful insight to interpret the learned policy, as discussed herein. It is noted that the Q-values are more closely spread throughout the diagonal of the optimal estimation when $\sigma^2 = 0$ m^2 compared to when $\sigma^2 = 25$ m^2, which suggests that the critic makes more accurate estimations consistently when the noise variance is lower. Specifically, the respective errors between the Q-value and actual return at the first and minimum obstacle detections are -156.82 and -49.92, respectively, when $\sigma^2 = 25$ m^2 and -96.88 and -25.43, respectively, when $\sigma^2 = 0$ m^2, as indicated by the timesteps $\times$ and $*$, respectively.

In addition, it is noted that the estimations are inherently different in each scenario, not only in terms of its accuracy. Specifically, the absolute area under the Q-values is reduced by -18.4% when $\sigma^2 = 25$ m^2 compared to when $\sigma^2 = 0$ m^2, where the normalised area per timestep for the path following manoeuvre corresponds to 57.2% of the difference, followed by the evasive manoeuvre and path recovery manoeuvre that correspond to 39.5% and 3.3%, respectively. The difference between the Q-values that is particularly evident during the path following manoeuvre even when no obstacle detection has occurred, suggests that the critic exhibits a general trend towards conservatism over optimism when the noise variance is higher. Considering that the reward functions remain unchanged regardless of the noise variance, the critic's conservatism can be attributed to the uncertainties of triggering the collision criterion. For instance, given equal obstacle positions and evasive manoeuvres in two episodes during the agent's training, the probability of the collision criterion being triggered is higher in the episode with the higher noise variance, which if triggered, leads to the automatic termination of the episode and consequently to the accumulation of a lower return.

To compensate that, the critic learns to underestimate its expected return when the noise variance is higher, which encourages the actor not to exploit the current policy but rather to explore. Exploitation of the policy corresponds to increased tendency towards maximising the return manifested as smaller evasive manoeuvres according to the prior finding on the agent's decision-making over the rewards. Exploration of the policy cor-

responds to increased tendency towards exploring different commanded rudder angles manifested as larger evasive manoeuvres, which verifies the prior finding on the actor's learned policy to output larger commanded rudder angles when the noise variance is higher. This lends credence to the notion of the critic's adaptability, stemming from its ability to adjust its estimation of the expected return depending on the noise variance, which is desirable from safety perspective considering conservatism during higher collision risk.



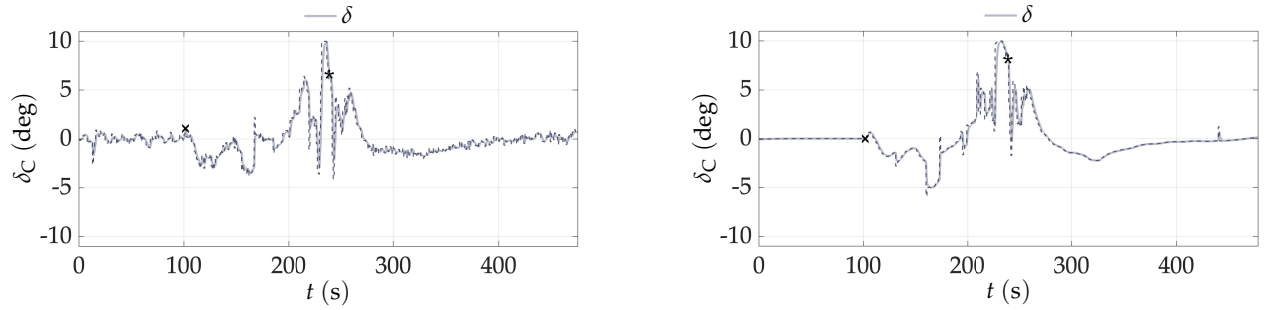
(**a**) (**b**)

Figure 14. Critic's estimation of the agent$_A$ in terms of the expected return Q-value per return when (**a**) $\sigma^2 = 0$ m^2 and (**b**) $\sigma^2 = 25$ m^2. The colour bar of each density map represents the density values, where darker colours indicate higher concentrations and lighter colours indicate lower concentrations. The $\times$ and $*$ represent the timesteps of the first and minimum obstacle detections, respectively.

4.3. Sensitivity Analysis

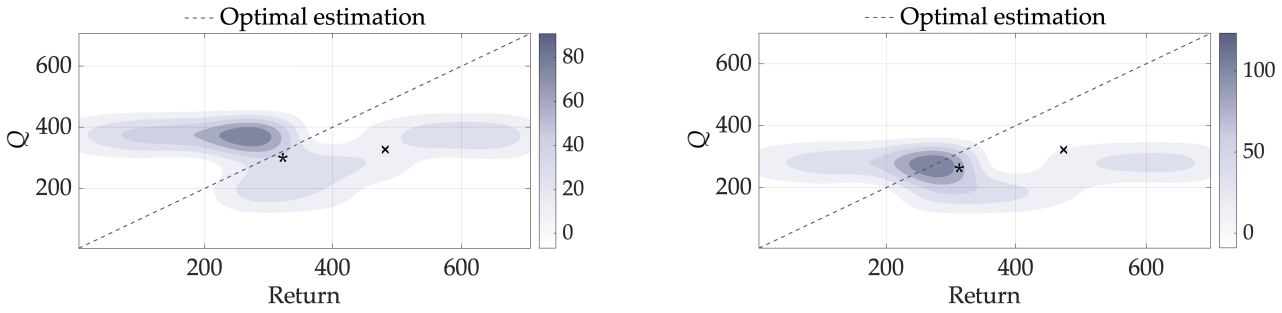
Sensitivity analysis of the trained agent$_A$'s actor and critic to noise variance is conducted, considering the prior findings suggesting its impact on their output. Specifically, the noise variance σ^2 affects the noise level of the LIDAR measurements observation d and the value of the noise variance observation $\mathcal{N}$, as presented prior in Equation (14). To investigate the sensitivity to each of these observations, they are decoupled by assigning two independent noise variances as σ_d^2 and $\sigma_{\mathcal{N}}^2$, respectively. The sensitivity is quantified by measuring the errors of the actor's and critic's outputs in terms of the commanded rudder angles and expected returns e_{δ_C} and e_Q, respectively, using as baselines for the analysis their original outputs, as presented prior in Section 4.2.

The agent is simulated in various scenarios investigating different combinations of noise variances $\sigma_d^2 = [0, 25]$ m^2 and $\sigma_{\mathcal{N}}^2 = [0, 25]$ m^2, but for brevity only the results pertaining to their minimum and maximum values are discussed. The actor's and critic's outputs when $\sigma_d^2 = 25$ m^2 and $\sigma_{\mathcal{N}}^2 = 0$ m^2 and when $\sigma_d^2 = 0$ m^2 and $\sigma_{\mathcal{N}} = 25$ m^2 are presented in Figures 15 and 16. Qualitative evaluation suggests that their outputs resemble the original outputs, as presented prior in Figures 13 and 14, where the similarities are found only when the noise variance $\sigma_{\mathcal{N}}^2$ is equal to the σ^2, regardless of the σ_d^2. The erratic output is introduced only when $\sigma_d^2 = 25$ m^2, which suggests that the erratic output is intrinsic to the LIDAR measurements observation d and not to the noise variance observation $\mathcal{N}$.



(**a**) (**b**)

Figure 15. Actor's policy of the agent$_{\mathrm{A}}$ in terms of the commanded rudder angle δ_{C} per timestep t when (**a**) $\sigma_d^2 = 25\ \mathrm{m}^2$ and $\sigma_{\mathcal{N}}^2 = 0\ \mathrm{m}^2$ and (**b**) $\sigma_d^2 = 0\ \mathrm{m}^2$ and $\sigma_{\mathcal{N}}^2 = 25\ \mathrm{m}^2$. The $\times$ and $*$ represent the timesteps of the first and minimum obstacle detection, respectively.



(**a**) (**b**)

Figure 16. Critic's estimation of the agent$_{\mathrm{A}}$ in terms of the expected return Q-value per return when (**a**) $\sigma_d^2 = 25\ \mathrm{m}^2$ and $\sigma_{\mathcal{N}}^2 = 0\ \mathrm{m}^2$ and (**b**) $\sigma_d^2 = 0\ \mathrm{m}^2$ and $\sigma_{\mathcal{N}}^2 = 25\ \mathrm{m}^2$. The colour bar of each density map represents the density values, where darker colours indicate higher concentrations and lighter colours indicate lower concentrations. The $\times$ and $*$ represent the timesteps of the first and minimum obstacle detection, respectively.

The results of the sensitivity analysis in terms of the distributions of the actor's and critic's errors $e_{\delta_{\mathrm{C}}}$ and e_Q, respectively, are presented in Table 3, Figures 17 and 18. It is noted that the actor's error $e_{\delta_{\mathrm{C}}}$ exhibits major difference on the standard deviation values $\sigma_{e_{\delta_{\mathrm{C}}}}$, which denote variability between the commanded rudder angles similar to the prior finding on the actor's output. The critic's error e_Q exhibits major difference on the mean values $\overline{e}_Q$, which denote central tendency deviation between the expected returns similar to the prior findings on the critic's output. Specifically, the differences are evident during the variations of the noise variance $\sigma_{\mathcal{N}}^2$ with the standard deviation and mean values of the actor's and critic's errors $\sigma_{e_{\delta_{\mathrm{C}}}}$ and $\overline{e}_Q$ differing up to 347.2% and −164,700.0%, respectively, compared to their respective values during the variations of the noise variance σ_d^2, which denotes that both the actor and critic are significantly more sensitive to the noise variance $\sigma_{\mathcal{N}}^2$. This lends credence to the notion that the noise variance observation $\mathcal{N}$ is the most critical one, to which the sophisticated decision-making of the agent, expressiveness of the actor's policy, and adaptability of the critic's estimation can be attributed. The practical implications of this notion is the importance of signal processing techniques not only in filtering out the noise, but also in accurately estimating the noise variance value from the noisy measurements in order to guarantee the expected decision-making from the agent. It provides a way forward towards a more effective and efficient integration between signal processing and decision-making techniques, which is pivotal to ensure safety of autonomous systems.

Table 3. Sensitivity analysis of the agent$_A$'s actor and critic to different noise variance combinations σ_d^2 and $\sigma_{\mathcal{N}}^2$.

σ^2 (m^2)	σ_d^2 (m^2)	$\sigma_{\mathcal{N}}^2$ (m^2)	$\bar{e}_{\delta_C}$ (deg)	$\sigma_{e_{\delta_C}}$ (deg)	$e_{\delta_C,\min}$ (deg)	$e_{\delta_C,\max}$ (deg)
0	25	0	-3.95×10^3	0.48	-5.47	2.60
0	0	25	3.35×10^3	1.53	-13.60	7.54
25	25	0	-7.23×10^3	1.61	-7.52	13.60
25	0	25	1.08×10^3	0.36	-2.38	3.46
σ^2 (m^2)	σ_d^2 (m^2)	$\sigma_{\mathcal{N}}^2$ (m^2)	$\bar{e}_Q$	σ_{e_Q}	$e_{Q,\min}$	$e_{Q,\max}$
0	25	0	4.26	16.41	-25.90	109.03
0	0	25	53.72	17.96	-26.14	80.98
25	25	0	-49.38	21.05	-87.22	59.03
25	0	25	0.03	2.37	-10.49	42.22

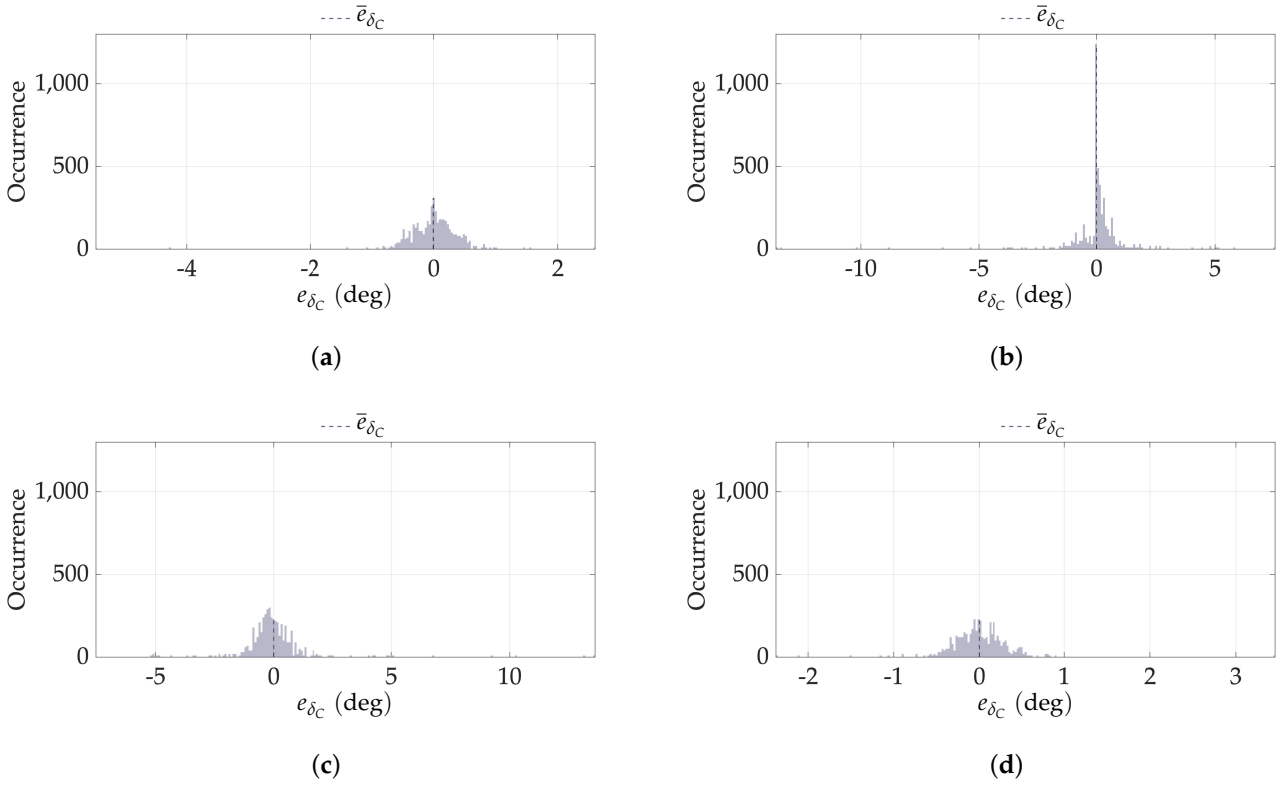


Figure 17. Distribution of the actor's output error of the agent$_A$ in terms of the commanded rudder angles e_{δ_C} when (**a**) $\sigma_d^2 = 25$ m^2 and $\sigma_{\mathcal{N}}^2 = 0$ m^2 from baseline $\sigma^2 = 0$ m^2, (**b**) $\sigma_d^2 = 0$ m^2 and $\sigma_{\mathcal{N}}^2 = 25$ m^2 from baseline $\sigma^2 = 0$ m^2, (**c**) $\sigma_d^2 = 25$ m^2 and $\sigma_{\mathcal{N}}^2 = 0$ m^2 from baseline $\sigma^2 = 25$ m^2, and (**d**) $\sigma_d^2 = 0$ m^2 and $\sigma_{\mathcal{N}}^2 = 25$ m^2 from baseline $\sigma^2 = 25$ m^2. The number of bins for each histogram is 250.

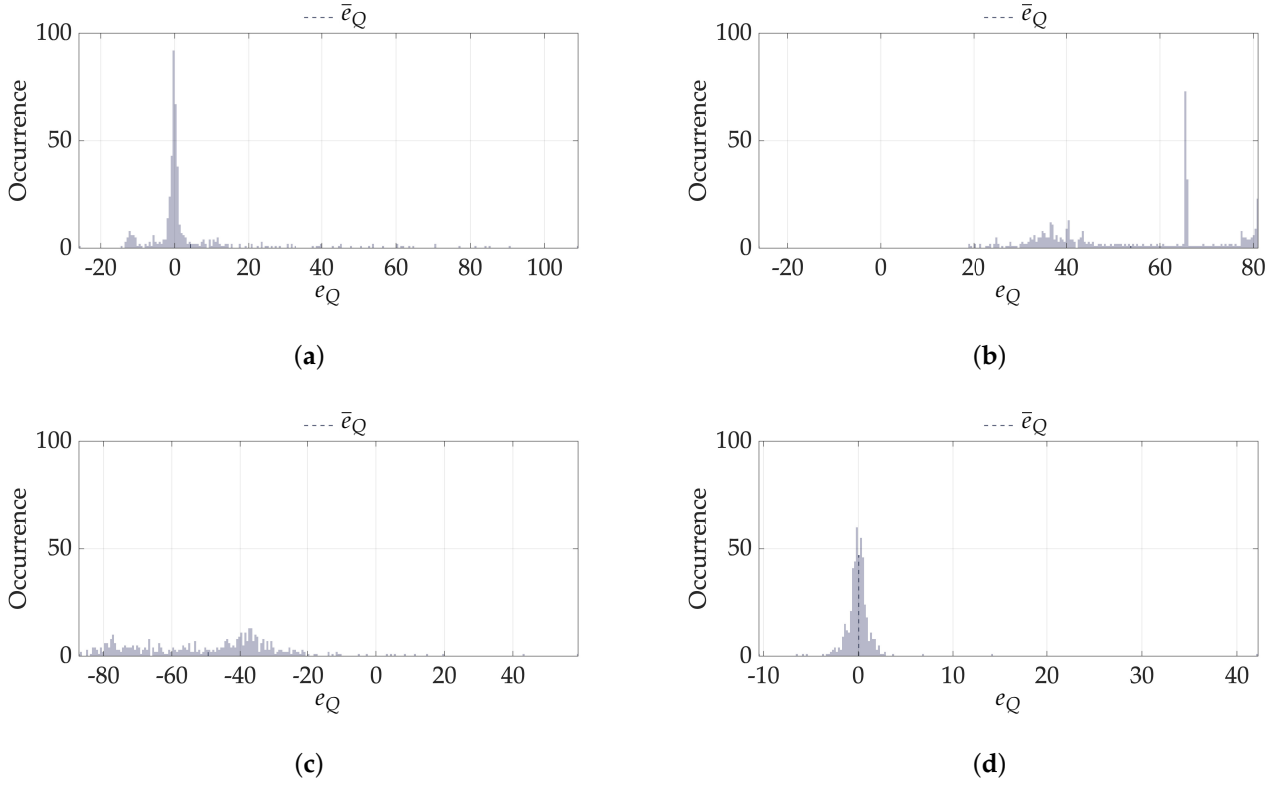


Figure 18. Distribution of the critic's output error of the agent$_A$ in terms of the expected returns e_Q when (**a**) $\sigma_d^2 = 25$ m^2 and $\sigma_{\mathcal{N}}^2 = 0$ m^2 from baseline $\sigma^2 = 0$ m^2, (**b**) $\sigma_d^2 = 0$ m^2 and $\sigma_{\mathcal{N}}^2 = 25$ m^2 from baseline $\sigma^2 = 0$ m^2, (**c**) $\sigma_d^2 = 25$ m^2 and $\sigma_{\mathcal{N}}^2 = 0$ m^2 from baseline $\sigma^2 = 25$ m^2, and (**d**) $\sigma_d^2 = 0$ m^2 and $\sigma_{\mathcal{N}}^2 = 25$ m^2 from baseline $\sigma^2 = 25$ m^2. The number of bins for each histogram is 250.

4.4. Robustness Verification

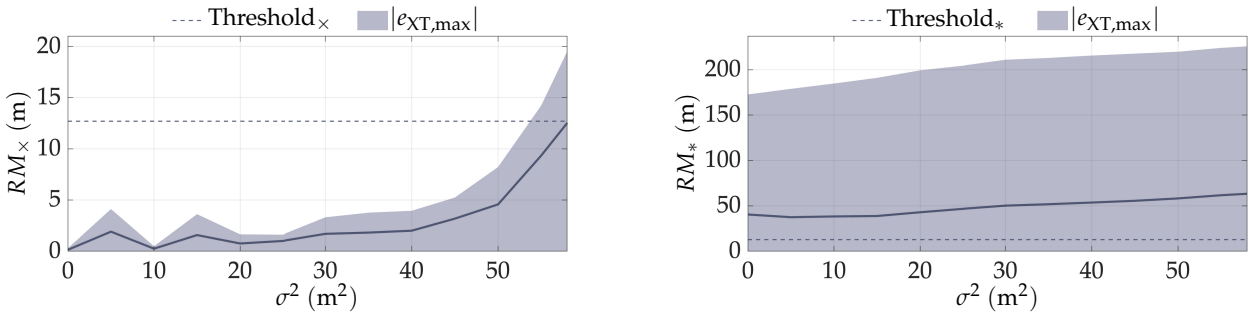
The robustness of the agent$_A$'s decision-making against noise variances in terms of the robustness metrics for path following and collision avoidance $RM_\times$ and RM_*, respectively, is presented in Table 4 and Figure 19. Specifically, the robustness metrics are measured within and outside the trained noise variance envelope, until the robustness thresholds are violated. It is worth noting that the maximum cross-track errors $|e_{\text{XT,max}}|$ are measured in the absence and presence of the obstacle as an indication of the expected maximum path deviation and maximum manoeuvre for the respective robustness metrics. It is noted that the robustness threshold for path following is compliant until when $\sigma^2 = 58$ m^2, which is an increase of 132% from its maximum trained value. The agent's decision-making in terms of the conducted manoeuvring when $\sigma^2 = 0$ m^2 and $\sigma^2 = 58$ m^2 is presented in Video S2. Specifically, the robustness metric for path following $RM_\times$ decreases approximately from $\sigma^2 = 40$ m^2 due to early path deviation prior to obstacle detection. However, it is worth noting that the robustness thresholds for path following as $B/2$ is already a stringent requirement considering that typical XTLs are of a few NM [87].

The robustness threshold for collision avoidance is compliant even when the threshold for path following is violated. Specifically, the robustness metric for collision avoidance RM_* exhibits an increasing trend, where it is increased by 15.5% when $\sigma^2 = 25$ m^2 and by 52.6% when $\sigma^2 = 58$ m^2 compared to when $\sigma^2 = 0$ m^2. It is worth noting that the increase of the robustness metric for collision avoidance is noted even prior to the early path deviation of the robustness metric for path following taking effect from $\sigma^2 = 40$ m^2. It can be derived that the agent's decision-making for path following and collision avoidance is robust against noise variances, due to the agent's ability to generalise its sophisticated decision-making of prioritising safety over efficiency across higher noise variances. This lends credence to the notion that the noise variance observation $\mathcal{N}$ continuous to exhibit

critical importance on the agent's decision-making even beyond the trained noise level envelope.

Table 4. Robustness metrics of the agent$_A$ against noise variances σ^2.

σ^2 (m^2)	$RM_{\times,A}$ (m)	$RM_{*,A}$ (m)
0	0.14	40.41
10	0.24	38.27
20	0.75	42.87
30	1.69	50.17
40	2.00	53.59
50	4.57	58.14
60	14.63	64.33



(**a**) (**b**)

Figure 19. Robustness of the agent$_A$'s decision-making against noise variance σ^2 in terms of the (**a**) robustness metric for path following $RM_\times$ and (**b**) robustness metric for collision avoidance RM_*.

4.5. Effectiveness of the Proposed Methodology

Agent$_B$ that exhibits the same hyperparameter settings as the agent$_A$ is trained within the same training framework, but without the noise variance observation $\mathcal{N}$, to investigate the effectiveness of the proposed methodology. The training performance of the agent$_B$ in terms of the return per episode, actor loss per step, and critic loss per step is presented in Figure 20. The training of the agent$_B$ is terminated at episode 5970, where it exhibits similar convergence trends as the agent$_A$. The minimum and maximum values of the training performance metrics derived during the training and their respective moving medians are presented in Table 5.

Table 5. Training performance metrics during the training of agent$_B$. The training performance metrics of the agent$_A$ are presented for comparison.

Return$_{A,min}$	**Return$_{A,max}$**	**Moving Median$_A$** [1]	**Return$_{B,min}$**	**Return$_{B,max}$**	**Moving Median$_B$** [2]
−67.42	749.72	633.84	−65.25	751.75	719.89
Actor loss$_{A,min}$	**Actor loss$_{A,max}$**	**Moving median$_A$** [3]	**Actor loss$_{B,min}$**	**Actor loss$_{B,max}$**	**Moving median$_B$** [4]
−344.62	−0.45	−288.36	−296.91	−0.21	−281.76
Critic loss$_{A,min}$	**Critic loss$_{A,max}$**	**Moving median$_A$** [3]	**Critic loss$_{B,min}$**	**Critic loss$_{B,max}$**	**Moving median$_B$** [4]
2.10×10^{-3}	3,750,911	19.39	1.69×10^{-3}	651,757.60	18.97

[1] At episode 5820. Sliding window of 150 episodes. [2] At episode 5970. Sliding window of 150 episodes. [3] At episode 5820. Sliding window of 25,000 steps. [4] At episode 5970. Sliding window of 25,000 steps.

The trained agent$_B$ is simulated in the same scenarios as the agent$_A$, but for brevity only the results pertaining to the minimum and maximum noise variances $\sigma^2 = 0$ m^2 and $\sigma^2 = 25$ m^2, respectively, are presented. The returns that the agent$_B$ accumulates

are 770.68 when $\sigma^2 = 25$ m^2 and 771.46 when $\sigma^2 = 0$ m^2, which suggest that the agent's decision-making is not significantly different in each scenario. The actor's policy in terms of the commanded rudder angle δ_C per timestep t is presented in Figure 21. The absolute area under the commanded rudder angles is increased only by 9.8% when $\sigma^2 = 25$ m^2 compared to when $\sigma^2 = 0$ m^2 and the magnitude of the commanded rudder angles at the minimum obstacle detections are 9.71 deg when $\sigma^2 = 25$ m^2 and 9.70 deg when $\sigma^2 = 0$ m^2, as indicated by the timesteps $*$, which suggest that the commanded rudder angles are not significantly different in each scenario, except for the erratic output noted when $\sigma^2 = 25$ m^2.

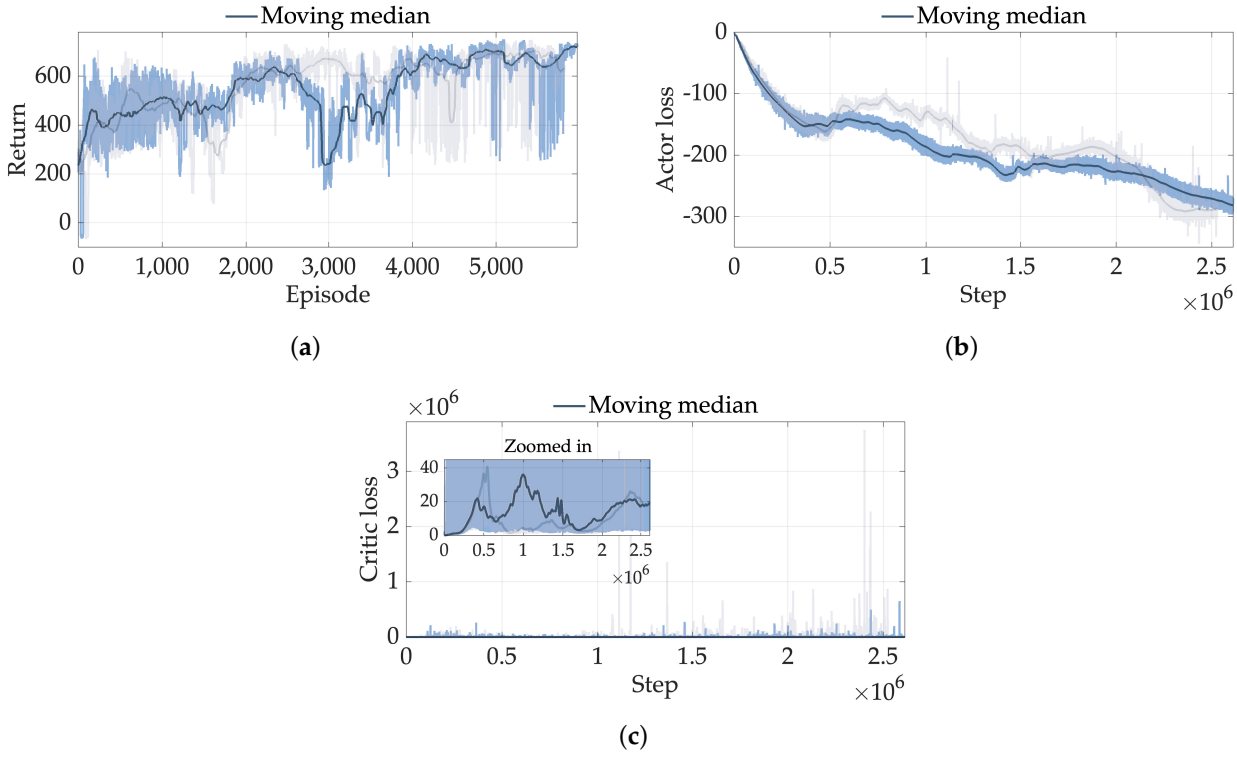


Figure 20. (**a**) Return per episode, (**b**) actor loss per step, and (**c**) critic loss per step during the training of agent$_B$. The sliding windows of their respective moving medians are 150 episodes and 25,000 steps. The training performance of the agent$_A$ are presented for comparison.

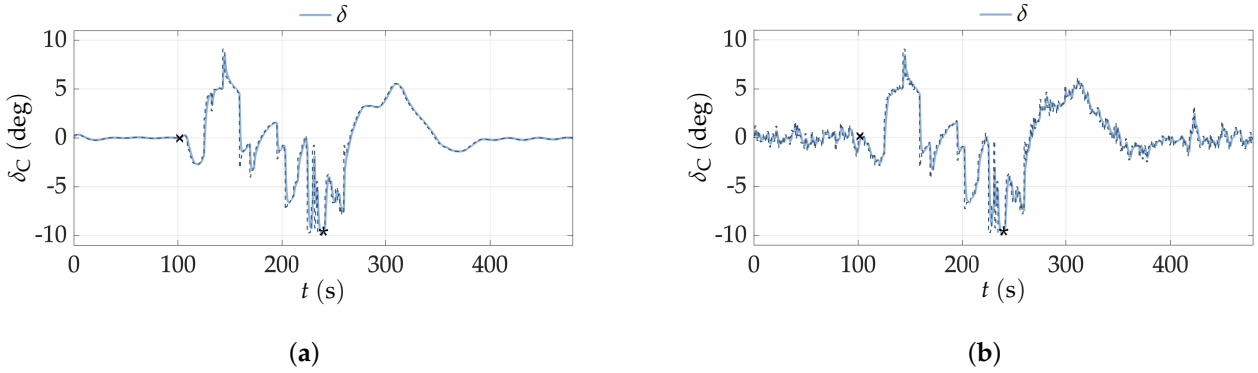
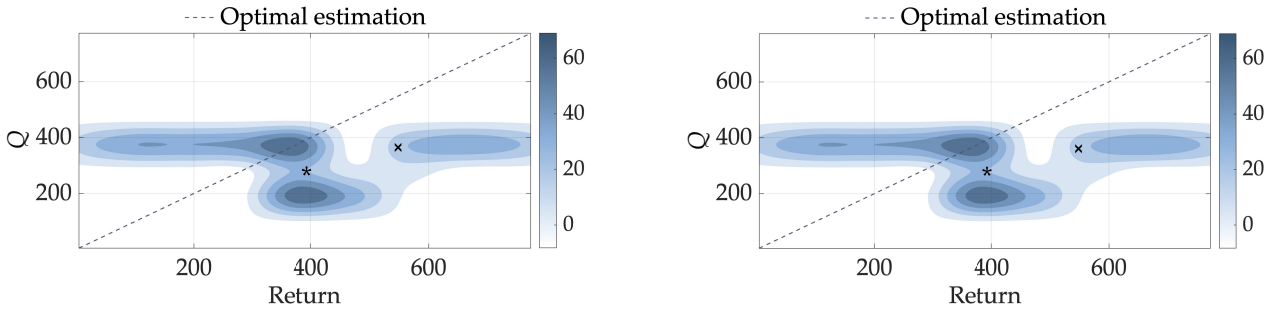


Figure 21. Actor's policy of the agent$_B$ in terms of the commanded rudder angle δ_C per timestep t when (**a**) $\sigma^2 = 0$ m^2 and (**b**) $\sigma^2 = 25$ m^2. The $\times$ and $*$ represent the timesteps of the first and minimum obstacle detection, respectively.

The critic's estimation in terms of the expected return Q-value per return is presented in Figure 22. The absolute area under the Q-values is reduced only by -0.1% when $\sigma^2 = 25$ m^2 compared to when $\sigma^2 = 0$ m^2 and the errors between the estimated Q-value

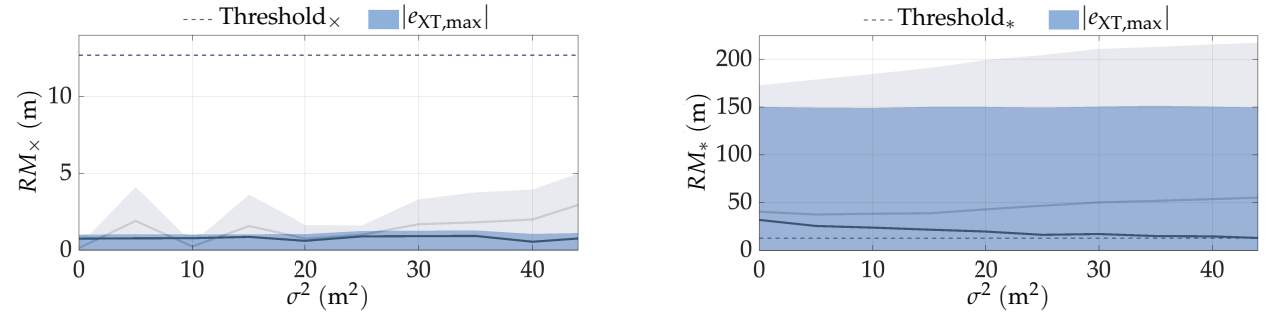
and actual return at the first and minimum obstacle detections are −183.83 and −117.16, respectively, when $\sigma^2 = 25$ m^2 and −187.78 and −116.74, respectively, when $\sigma^2 = 0$ m^2, as indicated by the timesteps $\times$ and $*$, respectively, which suggest that the Q-values are not significantly different in each scenario.

The robustness of the agent$_B$'s decision-making against noise variances in terms of the robustness metrics for path following and collision avoidance $RM_\times$ and RM_*, respectively, is presented in Table 6 and Figure 23. It is noted that the robustness threshold for path following is compliant even outside of the trained noise levels envelope. However, the robustness threshold for collision avoidance is violated when $\sigma^2 = 44$ m, which is an increase of only 76% from its maximum trained value. The agent$_B$'s decision-making in terms of the conducted manoeuvring when $\sigma^2 = 0$ m^2 and $\sigma^2 = 44$ m^2 is presented in Video S3. The maximum manoeuvres indicated by the maximum cross-track errors $|e_{XT,max}|$ suggest that the agent$_B$'s decision-making across different noise variances remain the same, consequently degrading the robustness against noise variances, especially for collision avoidance. This lends credence to the notion that training the agent without the noise variance observation $\mathcal{N}$ comes at the expense of reduced sophisticated decision-making, lack of expressiveness of the actor's policy and adaptability of the critic's estimation, with worse robustness and generalisation against noise variances, highlighting the effectiveness of the proposed methodology.



(**a**) (**b**)

Figure 22. Critic's estimation of the agent$_B$ in terms of the expected return Q-value per return when (**a**) $\sigma^2 = 0$ m^2 and (**b**) $\sigma^2 = 25$ m^2. The colour bar of each density map represents the density values, where darker colours indicate higher concentrations and lighter colours indicate lower concentrations. The $\times$ and $*$ represent the timesteps of the first and minimum obstacle detections, respectively.



(**a**) (**b**)

Figure 23. Robustness of the agent$_B$'s decision-making against noise variance σ^2 in terms of the (**a**) robustness metric for path following $RM_\times$ and (**b**) robustness metric for collision avoidance RM_*. The robustness metrics of the agent$_A$ are presented for comparison.

Table 6. Robustness metrics of the agent$_B$ against noise variances σ^2. Robustness metrics of the agent$_A$ are presented for comparison.

σ^2 (m^2)	$RM_{\times,A}$ (m)	$RM_{*,A}$ (m)	σ^2 (m^2)	$RM_{\times,B}$ (m)	$RM_{*,B}$ (m)
0	0.14	40.41	0	0.75	31.79
10	0.24	38.27	10	0.78	23.70
20	0.75	42.87	20	0.60	19.60
30	1.69	50.17	30	0.91	16.99
40	2.00	53.59	40	0.55	14.53
50	4.57	58.14	50	0.65	11.44

5. Conclusions

In this study, a methodology was developed to enhance the robustness of decision-making for the reactive collision avoidance of autonomous ships against various perception sensor noise levels. Digital twins of the ship manoeuvrability, perception sensor, and map were employed. A Gaussian-based noise model was employed to simulate the noisy measurements of the perception sensor. The decision-making problem was formulated as a Markov decision process, where rewards pertaining to path following, nominal navigation, actuator control, and collision avoidance objectives were identified. The noisy measurements of the perception sensor and its noise variance were incorporated into the decision-making as observations. A training framework was developed pertaining to the trained noise levels envelope. Robustness metrics that quantify the robustness of the agent's decision-making for path following and collision avoidance were defined, which were measured in various scenarios within and outside the trained noise levels envelope. A deep deterministic policy gradient agent was employed, where the actor was set to output the commanded rudder angles and the critic to estimate the expected return. A case study of a container ship using a light detection and ranging (LIDAR) in a single static obstacle environment was investigated. The main findings of this study are as follows.

1. The trained agent exhibited enhanced sophisticated decision-making prioritising safety over efficiency when the noise variance was higher. Specifically, the decision-making of the agent over the rewards exploited less the path following, nominal navigation, and actuator control rewards over the collision avoidance reward, manifested as larger evasive manoeuvres.
2. The actor's policy exhibited enhanced expressiveness by outputting different commanded rudder angles depending on the noise variance. Specifically, major difference in the commanded rudder angles was evident during the evasive manoeuvre verifying the agent's sophisticated decision-making. Also, the actor initiated an evasive manoeuvre by successfully discerning genuine obstacle detections amidst noisy measurements, which was attributed to the great feature extraction capabilities of deep neural networks.
3. The critic's estimation exhibited enhanced adaptability by adjusting its optimism and conservatism depending on the noise variance. Specifically, the critic learned to underestimate the expected return when the noise variance was higher verifying the actor's policy expressiveness. The critic's conservatism was attributed to the higher probability of triggering the collision criterion in higher noise variances.
4. Sensitivity analysis indicated the criticality of the noise variance observation for the agent's decision-making. Specifically, major differences in the actor's and critic's outputs were noted during the variations of the noise variance observation and not during the variations of the LIDAR measurements observation. However, the LIDAR measurements observation was found to be responsible for the erratic output.
5. The robustness of the agent's decision-making against noise variance was verified up to 132% from its maximum trained value. Specifically, the robustness for path following decreased with the increase of noise variance, but the robustness for collision avoidance increased by 52.6% from its initial value considering minimum noise

variance. The increase of robustness was attributed to the agent's ability to generalise its sophisticated decision-making across higher noise variances.

6. The robustness of the agent's decision-making against noise variance was verified only up to 76% from its maximum trained value, when trained without the noise variance observation. Specifically, the robustness for collision avoidance exhibited a decreasing trend with the increase of noise variances. This was attributed to the far less sophisticated decision-making, lack of expressiveness of the actor's policy and adaptability of the critic's estimation, with worse generalisation capabilities, highlighting the effectiveness of the proposed methodology.

The limitations of this study include the assumption of ideal environmental conditions with no disturbances and consideration of a single static obstacle. The noise model was also generic, whereas noise was only considered on the distance measurements and not on the angular measurements. Another limitation is pertaining to the heavy computation effort required for the training of deep reinforcement learning agents, considering the extensive time and expensive computation setup. Future studies entail the investigation of more realistic environments, including environmental disturbances and specific noise models pertaining to specific environmental conditions, data fusion of multiple perception sensors, and comparative analysis of alternative deep reinforcement learning algorithms. Nonetheless, this study contributes towards the development of autonomous systems that can make safe and robust decisions under uncertainty.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/jmse12040557/s1, Video S1: Agent_A against noise variance $\sigma^2 = 0\ \text{m}^2$ and $\sigma^2 = 25\ \text{m}^2$, Video S2: Agent_A against noise variance $\sigma^2 = 0\ \text{m}^2$ and $\sigma^2 = 58\ \text{m}^2$, Video S3: Agent_B against noise variance $\sigma^2 = 0\ \text{m}^2$ and $\sigma^2 = 44\ \text{m}^2$.

Author Contributions: Conceptualisation, P.L. and G.T.; methodology, P.L., G.T., and E.B.; software, P.L.; validation, P.L., G.T., and E.B.; resources, P.L.; data curation, P.L.; writing—original draft preparation, P.L. and G.T.; writing—review and editing, P.L., G.T., and E.B.; visualization, P.L.; supervision, G.T. and E.B.; project administration, G.T.; funding acquisition, G.T. All authors have read and agreed to the published version of the manuscript.

Funding: Part of this study was carried out under the framework of the AUTOSHIP project, which is funded by the European Union's Horizon 2020 research and innovation program under agreement No. 815012.

Data Availability Statement: Datasets can be made available upon request.

Acknowledgments: The authors greatly acknowledge the funding from DNV AS and RCCL for the MSRC establishment and operation. The opinions expressed herein are those of the authors and should not be construed to reflect the views of EU, DNV AS, RCCL, or the AUTOSHIP partners.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

2D, 3D	two or three dimensional
3DOF, 4DOF	three or four degrees of freedom
AI	artificial intelligence
CFD	computational fluid dynamics
COLREGs	International Regulations for Preventing Collisions at Sea
DDPG	deep deterministic policy gradient
DNN	deep neural networks
DQN	deep Q network
DRL	deep reinforcement learning
DW	dynamic window
EKF	extended Kalman filter
LIDAR	light detection and ranging

MASS	maritime autonomous surface ship
MDP	Markov decision-process
OS	own ship
RADAR	radio detection and ranging
SMCR	specified maximum continuous rating
SOLAS	International Convention for the Safety of Life at Sea
TD3	twin delayed deep deterministic policy gradient
XTL	cross-track limit

References

1. Sepehri, A.; Vandchali, H.R.; Siddiqui, A.W.; Montewka, J. The impact of shipping 4.0 on controlling shipping accidents: A systematic literature review. *Ocean Eng.* **2022**, *243*, 110162. [CrossRef]
2. Benamara, H.; Hoffmann, J.; Youssef, F. Maritime transport: The sustainability imperative. In *Sustainable Shipping: A Cross-Disciplinary View*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 1–31.
3. Lv, H.; Lloret, J.; Song, H. Guest Editorial Introduction to the Special Issue on Internet of Things in Intelligent Transportation Infrastructure. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 12843–12851. [CrossRef]
4. Guerrero-Ibanez, J.A.; Zeadally, S.; Contreras-Castillo, J. Integration challenges of intelligent transportation systems with connected vehicle, cloud computing, and internet of things technologies. *IEEE Wirel. Commun.* **2015**, *22*, 122–128. [CrossRef]
5. Zhu, L.; Yu, F.R.; Wang, Y.; Ning, B.; Tang, T. Big data analytics in intelligent transportation systems: A survey. *IEEE Trans. Intell. Transp. Syst.* **2018**, *20*, 383–398. [CrossRef]
6. Jabbar, R.; Dhib, E.; Said, A.B.; Krichen, M.; Fetais, N.; Zaidan, E.; Barkaoui, K. Blockchain technology for intelligent transportation systems: A systematic literature review. *IEEE Access* **2022**, *10*, 20995–21031. [CrossRef]
7. Xiong, G.; Zhu, F.; Liu, X.; Dong, X.; Huang, W.; Chen, S.; Zhao, K. Cyber-physical-social system in intelligent transportation. *IEEE/CAA J. Autom. Sin.* **2015**, *2*, 320–333. [CrossRef]
8. Wang, Z.; Gupta, R.; Han, K.; Wang, H.; Ganlath, A.; Ammar, N.; Tiwari, P. Mobility digital twin: Concept, architecture, case study, and future challenges. *IEEE Internet Things J.* **2022**, *9*, 17452–17467. [CrossRef]
9. Böckin, D.; Tillman, A.M. Environmental assessment of additive manufacturing in the automotive industry. *J. Clean. Prod.* **2019**, *226*, 977–987. [CrossRef]
10. Wicaksana, I.P.R.E.; Christian, J.; Achmad, S.; Sutoyo, R. Effect of Visual Augmented Reality in the Transportation Sector. In Proceedings of the 2022 International Conference on Informatics Electrical and Electronics (ICIEE), Yogyakarta, Indonesia, 5–7 October 2022; pp. 1–5.
11. Abduljabbar, R.; Dia, H.; Liyanage, S.; Bagloee, S.A. Applications of artificial intelligence in transport: An overview. *Sustainability* **2019**, *11*, 189. [CrossRef]
12. IMO. IMO Takes First Steps to Address Autonomous Ships. 2018 Available online: https://www.imo.org/en/MediaCentre/PressBriefings/Pages/08-MSC-99-MASS-scoping.aspx#:~:text=For%20the%20purpose%20of%20the,operate%20independently%20of%20human%20interaction.&text=Remotely%20controlled%20ship%20without%20seafarers,and%20operated%20from%20another%20location (accessed on 7 November 2023).
13. Aiello, G.; Giallanza, A.; Mascarella, G. Towards Shipping 4.0. A preliminary gap analysis. *Procedia Manuf.* **2020**, *42*, 24–29. [CrossRef]
14. Van Brummelen, J.; O'Brien, M.; Gruyer, D.; Najjaran, H. Autonomous vehicle perception: The technology of today and tomorrow. *Transp. Res. Part Emerg. Technol.* **2018**, *89*, 384–406. [CrossRef]
15. Shahian Jahromi, B.; Tulabandhula, T.; Cetin, S. Real-time hybrid multi-sensor fusion framework for perception in autonomous vehicles. *Sensors* **2019**, *19*, 4357. [CrossRef] [PubMed]
16. Campbell, S.; O'Mahony, N.; Krpalcova, L.; Riordan, D.; Walsh, J.; Murphy, A.; Ryan, C. Sensor technology in autonomous vehicles: A review. In Proceedings of the 2018 29th Irish Signals and Systems Conference (ISSC), Belfast, UK, 21–22 June 2018; pp. 1–4.
17. Schuster, M.; Blaich, M.; Reuter, J. Collision avoidance for vessels using a low-cost radar sensor. *IFAC Proc. Vol.* **2014**, *47*, 9673–9678. [CrossRef]
18. Namgung, H. Local route planning for collision avoidance of maritime autonomous surface ships in compliance with COLREGs rules. *Sustainability* **2021**, *14*, 198. [CrossRef]
19. Song, A.L.; Su, B.Y.; Dong, C.Z.; Shen, D.W.; Xiang, E.Z.; Mao, F.P. A two-level dynamic obstacle avoidance algorithm for unmanned surface vehicles. *Ocean Eng.* **2018**, *170*, 351–360. [CrossRef]
20. Lee, P.; Theotokatos, G.; Boulougouris, E.; Bolbot, V. Risk-informed collision avoidance system design for maritime autonomous surface ships. *Ocean Eng.* **2023**, *279*, 113750. [CrossRef]
21. Li, B.; Chan, P.H.; Baris, G.; Higgins, M.D.; Donzella, V. Analysis of automotive camera sensor noise factors and impact on object detection. *IEEE Sens. J.* **2022**, *22*, 22210–22219. [CrossRef]
22. Li, B.; Chan, P.H.; Higgins, M.; Donzella, V.; Baris, G. Analysis of Automotive Camera Sensor Noise Factors. *Authorea Prepr.* **2023**, *22*, 22210–22219.

23. Clunie, T.; DeFilippo, M.; Sacarny, M.; Robinette, P. Development of a perception system for an autonomous surface vehicle using monocular camera, lidar, and marine radar. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 14112–14119
24. Saxena, S.; Isukapati, I.K.; Smith, S.F.; Dolan, J.M. Multiagent sensor fusion for connected & autonomous vehicles to enhance navigation safety. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019; pp. 2490–2495.
25. Chan, P.H.; Dhadyalla, G.; Donzella, V. A framework to analyze noise factors of automotive perception sensors. *IEEE Sens. Lett.* **2020**, *4*, 1–4. [CrossRef]
26. Xu, H.; Hinostroza, M.A.; Guedes Soares, C. Modified Vector Field Path-Following Control System for an underactuated Autonomous surface ship model in the presence of static obstacles. *J. Mar. Sci. Eng.* **2021**, *9*, 652. [CrossRef]
27. Blindheim, S.; Gros, S.; Johansen, T.A. Risk-based model predictive control for autonomous ship emergency management. *IFAC PapersOnLine* **2020**, *53*, 14524–14531. [CrossRef]
28. Gao, D.; Zhou, P.; Shi, W.; Wang, T.; Wang, Y. A dynamic obstacle avoidance method for unmanned surface vehicle under the international regulations for preventing collisions at sea. *J. Mar. Sci. Eng.* **2022**, *10*, 901. [CrossRef]
29. Serigstad, E.; Eriksen, B.O.H.; Breivik, M. Hybrid collision avoidance for autonomous surface vehicles. *IFAC PapersOnLine* **2018**, *51*, 1–7. [CrossRef]
30. Wang, W.; Huang, L.; Liu, K.; Wu, X.; Wang, J. A COLREGs-Compliant Collision Avoidance Decision Approach Based on Deep Reinforcement Learning. *J. Mar. Sci. Eng.* **2022**, *10*, 944. [CrossRef]
31. Cheng, Y.; Zhang, W. Concise deep reinforcement learning obstacle avoidance for underactuated unmanned marine vessels. *Neurocomputing* **2018**, *272*, 63–73. [CrossRef]
32. Zhou, C.; Wang, Y.; Wang, L.; He, H. Obstacle avoidance strategy for an autonomous surface vessel based on modified deep deterministic policy gradient. *Ocean Eng.* **2022**, *243*, 110166. [CrossRef]
33. Meyer, E.; Robinson, H.; Rasheed, A.; San, O. Taming an autonomous surface vehicle for path following and collision avoidance using deep reinforcement learning. *IEEE Access* **2020**, *8*, 41466–41481. [CrossRef]
34. Heiberg, A.; Larsen, T.N.; Meyer, E.; Rasheed, A.; San, O.; Varagnolo, D. Risk-based implementation of COLREGs for autonomous surface vehicles using deep reinforcement learning. *Neural Netw.* **2022**, *152*, 17–33. [CrossRef] [PubMed]
35. Kim, J.S.; Lee, D.H.; Kim, D.W.; Park, H.; Paik, K.J.; Kim, S. A numerical and experimental study on the obstacle collision avoidance system using a 2D LiDAR sensor for an autonomous surface vehicle. *Ocean Eng.* **2022**, *257*, 111508. [CrossRef]
36. Gonzalez-Garcia, A.; Collado-Gonzalez, I.; Cuan-Urquizo, R.; Sotelo, C.; Sotelo, D.; Castañeda, H. Path-following and LiDAR-based obstacle avoidance via NMPC for an autonomous surface vehicle. *Ocean Eng.* **2022**, *266*, 112900. [CrossRef]
37. Villa, J.; Aaltonen, J.; Koskinen, K.T. Path-following with lidar-based obstacle avoidance of an unmanned surface vehicle in harbor conditions. *IEEE/ASME Trans. Mechatron.* **2020**, *25*, 1812–1820. [CrossRef]
38. Wang, Y.; Yu, X.; Liang, X.; Li, B. A COLREGs-based obstacle avoidance approach for unmanned surface vehicles. *Ocean Eng.* **2018**, *169*, 110–124. [CrossRef]
39. Peng, Y.; Yang, Y.; Cui, J.; Li, X.; Pu, H.; Gu, J.; Xie, S.; Luo, J. Development of the USV 'JingHai-I'and sea trials in the Southern Yellow Sea. *Ocean Eng.* **2017**, *131*, 186–196. [CrossRef]
40. Han, J.; Cho, Y.; Kim, J.; Kim, J.; Son, N.s.; Kim, S.Y. Autonomous collision detection and avoidance for ARAGON USV: Development and field tests. *J. Field Robot.* **2020**, *37*, 987–1002. [CrossRef]
41. Han, J.; Kim, S.Y.; Kim, J. Enhanced target ship tracking with geometric parameter estimation for unmanned surface vehicles. *IEEE Access* **2021**, *9*, 39864–39872. [CrossRef]
42. Kim, J.; Lee, C.; Chung, D.; Cho, Y.; Kim, J.; Jang, W.; Park, S. Field experiment of autonomous ship navigation in canal and surrounding nearshore environments. *J. Field Robot.* **2023**, *41*, 470–489. [CrossRef]
43. Nielsen, R.E.; Papageorgiou, D.; Nalpantidis, L.; Jensen, B.T.; Blanke, M. Machine learning enhancement of manoeuvring prediction for ship Digital Twin using full-scale recordings. *Ocean Eng.* **2022**, *257*, 111579. [CrossRef]
44. Glaessgen, E.; Stargel, D. The digital twin paradigm for future NASA and US Air Force vehicles. In Proceedings of the 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Honolulu, HI, USA, 23–26 April 2012 ; p. 1818.
45. Zai, A.; Brown, B. *Deep Reinforcement Learning in Action*; Manning Publications: Shelter Island, NY, USA, 2020.
46. MathWorks. Navigation Toolbox. Design, Simulate, and Deploy Algorithms for Autonomous Navigation. Available online: https://www.mathworks.com/products/navigation.html (accessed on 1 January 2024).
47. MathWorks. Lidar Toolbox. Design, Analyze, and Test Lidar Processing Systems. Available online: https://www.mathworks.com/products/lidar.html (accessed on 1 January 2024).
48. MathWorks. Robotics System Toolbox. Design, Simulate, Test, and Deploy Robotics Applications. Available online: https://www.mathworks.com/products/robotics.html (accessed on 1 January 2024).
49. Zhang, C.; Liu, X.; Wan, D.; Wang, J. Experimental and numerical investigations of advancing speed effects on hydrodynamic derivatives in MMG model, part I: Xvv, Yv, Nv. *Ocean Eng.* **2019**, *179*, 67–75. [CrossRef]
50. Kim, D.; Song, S.; Jeong, B.; Tezdogan, T. Numerical evaluation of a ship's manoeuvrability and course keeping control under various wave conditions using CFD. *Ocean Eng.* **2021**, *237*, 109615. [CrossRef]

51. Song, S.; Kim, D.; Dai, S. CFD investigation into the effect of GM variations on ship manoeuvring characteristics. *Ocean Eng.* **2024**, *291*, 116472. [CrossRef]
52. He, S.; Kellett, P.; Yuan, Z.; Incecik, A.; Turan, O.; Boulougouris, E. Manoeuvring prediction based on CFD generated derivatives. *J. Hydrodyn.* **2016**, *28*, 284–292. [CrossRef]
53. Jin, Y.; Yiew, L.J.; Zheng, Y.; Magee, A.R.; Duffy, J.; Chai, S. Dynamic manoeuvres of KCS with CFD free-running computation and system-based modelling. *Ocean Eng.* **2021**, *241*, 110043. [CrossRef]
54. Suzuki, R.; Ueno, M.; Tsukada, Y. Numerical simulation of 6-degrees-of-freedom motions for a manoeuvring ship in regular waves. *Appl. Ocean Res.* **2021**, *113*, 102732. [CrossRef]
55. Wang, J.; Zou, L.; Wan, D. CFD simulations of free running ship under course keeping control. *Ocean Eng.* **2017**, *141*, 450–464. [CrossRef]
56. Abkowitz, M.A. *Lectures on Ship Hydrodynamics–Steering and Manoeuvrability*; Technical Report; Stevens Institute of Technology: Hoboken, NJ, USA, 1964.
57. Ogawa, A.; Koyama, T.; Kijima, K. MMG report-I, on the mathematical model of ship manoeuvring. *Bull. Soc. Naval Archit. Jpn.* **1977**, *575*, 22–28.
58. Budak, G.; Beji, S. Controlled course-keeping simulations of a ship under external disturbances. *Ocean Eng.* **2020**, *218*, 108126. [CrossRef]
59. Sui, C.; De Vos, P.; Hopman, H.; Visser, K.; Stapersma, D.; Ding, Y. Effects of adverse sea conditions on propulsion and manoeuvring performance of low-powered ocean-going cargo ship. *Ocean Eng.* **2022**, *254*, 111348. [CrossRef]
60. Yasukawa, H.; Sakuno, R. Application of the MMG method for the prediction of steady sailing condition and course stability of a ship under external disturbances. *J. Mar. Sci. Technol.* **2020**, *25*, 196–220. [CrossRef]
61. Guo, H.; Zou, Z. System-based investigation on 4-DOF ship maneuvering with hydrodynamic derivatives determined by RANS simulation of captive model tests. *Appl. Ocean Res.* **2017**, *68*, 11–25. [CrossRef]
62. Wu, T.; Li, R.; Chen, Q.; Pi, G.; Wan, S.; Liu, Q. A Numerical Study on Modeling Ship Maneuvering Performance Using Twin Azimuth Thrusters. *J. Mar. Sci. Eng.* **2023**, *11*, 2167. [CrossRef]
63. Yasukawa, H.; Yoshimura, Y. Introduction of MMG standard method for ship maneuvering predictions. *J. Mar. Sci. Technol.* **2015**, *20*, 37–52. [CrossRef]
64. Son, K.H.; Nomoto, K. On the coupled motion of steering and rolling of a high-speed container ship. *Nav. Archit. Ocean Eng.* **1982**, *20*, 73–83. [CrossRef] [PubMed]
65. Szlapczynski, R.; Krata, P.; Szlapczynska, J. A ship domain-based method of determining action distances for evasive manoeuvres in stand-on situations *J. Adv. Transp.* **2018**. [CrossRef]
66. Vivacqua, R.; Vassallo, R.; Martins, F. A low cost sensors approach for accurate vehicle localization and autonomous driving application. *Sensors* **2017**, *17*, 2359. [CrossRef]
67. Hernandez-Aceituno, J.; Arnay, R.; Toledo, J.; Acosta, L. Using kinect on an autonomous vehicle for outdoors obstacle detection. *IEEE Sens. J.* **2016**, *16*, 3603–3610. [CrossRef]
68. Rosique, F.; Navarro, P.J.; Fernández, C.; Padilla, A. A systematic review of perception system and simulators for autonomous vehicles research. *Sensors* **2019**, *19*, 648. [CrossRef] [PubMed]
69. Thombre, S.; Zhao, Z.; Ramm-Schmidt, H.; García, J.M.V.; Malkamäki, T.; Nikolskiy, S.; Hammarberg, T.; Nuortie, H.; Bhuiyan, M.Z.H; Särkkä, S.; et al. Sensors and AI techniques for situational awareness in autonomous ships: A review. *IEEE Trans. Intell. Transp. Syst.* **2020**, *23*, 64–83. [CrossRef]
70. Robinette, P.; Sacarny, M.; DeFilippo, M.; Novitzky, M.; Benjamin, M.R. Sensor evaluation for autonomous surface vehicles in inland waterwaysp. *Oceans 2019-Marseille* **2019**, *254*, 1–8.
71. Kim, K.; Kim, J.; Kim, J. Robust data association for multi-object detection in maritime environments using camera and radar measurements. *IEEE Robot. Autom. Lett.* **2021**, *6*, 5865–5872. [CrossRef]
72. Peeters, G.; Kotzé, M.; Afzal, M.R.; Catoor, T.; Van Baelen, S.; Geenen, P.; Vanierschot, M.; Boonen, R.; Slaets, P. An unmanned inland cargo vessel: Design, build, and experiments. *Ocean Eng.* **2020**, *201*, 107056. [CrossRef]
73. Zhang, C.; Huang, Z; Ang, M.H.; Rus, D. 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). *Ocean Eng.* **2021**, *201*, 3458–3464.
74. Fang, H.; Huang, D. Noise reduction in lidar signal based on discrete wavelet transform. *Opt. Commun.* **2004**, *233*, 67–76 [CrossRef]
75. Gimmestad, G.G. Roberts, D.W. *Lidar Engineering: Introduction to Basic Principles*; Cambridge University Press: Cambridge, UK, 2023; Volume 233, pp. 67–76
76. Incoronato, A.; Locatelli, M.; Zappa, F. Statistical modelling of SPADs for time-of-flight LiDAR. *Sensors* **2021**, *21*, 4481. [CrossRef] [PubMed]
77. Yasuda, Y.DV.; Martins, L.E.G.; Cappabianco, F.AM. Autonomous visual navigation for mobile robots: A systematic literature review. *ACM Comput. Surv. (CSUR)* **2020**, *53*, 1–34. [CrossRef]
78. Guivant, J.; Nebot, E.; Nieto, J.; Masson, F. Navigation and mapping in large unstructured environments. *Int. J. Robot. Res.* **2004**, *23*, 449–472. [CrossRef]
79. Ebrahimi S.B.; Razzaghpour, M.; Valiente, R.; Raftari, A.; Fallah, Y.P. High-definition map representation techniques for automated vehicles. *Electronics* **2022**, *11*, 3374. [CrossRef]

80. Filliat, D.; Meyer, J.A. Map-based navigation in mobile robots: I a review of localization strategies. *Cogn. Syst. Res.* **2003**, *4*, 243–282. [CrossRef]
81. Chen, E.; Guo, J. Real time map generation using sidescan sonar scanlines for unmanned underwater vehicles. *Ocean Eng.* **2014**, *91*, 252–262. [CrossRef]
82. Han, S.; Wang, L.; Wang, Y.; He, H. An efficient motion planning based on grid map: Predicted Trajectory Approach with global path guiding. *Ocean Eng.* **2021**, *238*, 109696. [CrossRef]
83. Graesser, L.; Keng, W.L. *Foundations of Deep Reinforcement Learning*; Addison-Wesley Professional, Pearson Education: Boston, MA, USA, 2019.
84. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 2018.
85. Morales, M. *Grokking Deep Reinforcement Learning*; Manning Publications: Shelter Island, NY, USA, 2020.
86. Fossen, T.I.; Breivik, M.; Skjetne, R. Line-of-sight path following of underactuated marine craft. *IFAC Proc. Vol.* **2003**, *36*, 211–216. [CrossRef]
87. Kristić, M.; Žuškin, S.; Brčić, D.; Valčić, S. Zone of confidence impact on cross track limit determination in ECDIS passage planning. *J. Mar. Sci. Eng.* **2020**, *8*, 566. [CrossRef]
88. Wang, X.; Wang, S.; Liang, X.; Zhao, D.; Huang, J.; Xu, X.; Dai, B.; Miao, Q. Deep reinforcement learning: A survey. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, 1–15.
89. Mousavi, S.S.; Schukat, M.; Howley, E. Deep reinforcement learning: An overview. In *Proceedings of SAI Intelligent Systems Conference (IntelliSys) 2016: Volume 2*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 426–440.
90. Woo, J.; Kim, N. Collision avoidance for an unmanned surface vehicle using deep reinforcement learning. *Ocean Eng.* **2020**, *199*, 107001. [CrossRef]
91. Jaritz, M.; De Charette, R.; Toromanoff, M.; Perot, E.; Nashashibi, F. End-to-end race driving with deep reinforcement learning. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 2070–2075.
92. Kilinc, O.; Montana, G. Multi-agent deep reinforcement learning with extremely noisy observations. *arXiv* **2018**, arXiv:1812.00922.
93. 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* **2015**, *518*, 529–533. [CrossRef] [PubMed]
94. MathWorks. Deep Learning Toolbox. Design, Train, and Analyze Deep Learning Networks. Available online: https://www.mathworks.com/products/deep-learning.html (accessed on 1 January 2024).
95. MathWorks. Reinforcement Learning Toolbox. Design and Train Policies Using Reinforcement Learning. Available online: https://www.mathworks.com/products/reinforcement-learning.html (accessed on 1 January 2024).
96. Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; Moritz, P. Trust region policy optimization. *Int. Conf. Mach. Learn.* **2015**, *37*, 1889–1897.
97. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. *arXiv* **2015**, arXiv:1509.02971.
98. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. *arXiv* **2017,** arXiv:1707.06347.
99. Haarnoja, T.; Zhou, A.; Abbeel, P.; Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *Int. Conf. Mach. Learn.* **2018**, *80*, 1861–1870.
100. Fujimoto, S.; Hoof, H.; Meger, D. International conference on machine learning. *arXiv* **2018**, arXiv:1587.1596.
101. Dutta, D.; Upreti, S.R. A survey and comparative evaluation of actor-critic methods in process control. *Can. J. Chem. Eng.* **2022**, *100*, 2028–2056. [CrossRef]
102. Zhang, B.; Jin, C.; Cao, K.; Lv, Q.; Zhang, P.; Li, Y.; Li, M. Ultra-wide-scanning conformal heterogeneous phased array antenna based on deep deterministic policy gradient algorithm. *IEEE Trans. Antennas Propag.* **2022**, *70* 5066–5077. [CrossRef]
103. Torben, T.; Smogeli, Ø.; Utne, I.B.; Sørensen, A.J. On Formal Methods for Design and Verification of Maritime Autonomous Surface Ships. 2022. Available online: https://hdl.handle.net/11250/3058210 (accessed on 1 January 2024).
104. Hamon, R.; Junklewitz, H.; Sanchez, I. *Robustness and Explainability of Artificial Intelligence*; Publications Office of the European Union: Luxembourg, 2020; Volume 207.
105. Kim, M.; Hizir, O.; Turan, O.; Day, S.; Incecik, A. Estimation of added resistance and ship speed loss in a seaway. *Ocean Eng.* **2017**, *141* 465–476. [CrossRef]
106. Fossen. Marine Systems Simulator (MSS). 2004. Available online: http://www.marinecontrol.org (accessed on 7 January 2024).
107. Yasukawa, H.; Sakuno, R.; Yoshimura, Y. Practical maneuvering simulation method of ships considering the roll-coupling effect. *J. Mar. Sci. Technol.* **2019**, *24* 1280–1296. [CrossRef]
108. Li, S.; Liu, C.; Chu, X.; Zheng, M.; Wang, Z.; Kan, J. Ship maneuverability modeling and numerical prediction using CFD with body force propeller. *Ocean Eng.* **2022**, *264*, 112454. [CrossRef]
109. Moreira, L.; Soares, C.G. Dynamic model of manoeuvrability using recursive neural networks. *Ocean Eng.* **2003**, *30*, 1669–1697. [CrossRef]
110. Van Amerongen, J. Adaptive steering of ships—A model reference approach. *Automatica* **1984**, *20*, 3–14. [CrossRef]
111. Perez, T. *Ship Motion Control: Course Keeping and Roll Stabilisation Using Rudder and Fins*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2006.

112. IMO. Consolidated Edition 2020 Available online: https://wwwcdn.imo.org/localresources/en/publications/Documents/Flyers/Flyers/IG110E.pdf (accessed on 1 January 2024).
113. Xiang, J.; Chen, Y.; Li, W. Identification method of offset interference rudder angle and its application in ship steering control system. In Proceedings of the 2021 China Automation Congress (CAC), Beijing, China, 22–24 October 2021; pp. 5924–5928.
114. MAN. Propulsion Trends in Container Vessels. 2019. Available online: https://www.man-es.com/docs/default-source/marine/tools/propulsion-trends-in-container-vessels.pdf?sfvrsn=c48bba16_12 (accessed on 1 January 2024).
115. Hull, G. Real-Time Occupancy Grid Mapping Using LSD-SLAM. Ph.D. Thesis, Stellenbosch University, Stellenbosch, South Africa, 2017.
116. Ugurlu, H.; Cicek, I. Analysis and assessment of ship collision accidents using Fault Tree and Multiple Correspondence Analysis. *Ocean Eng.* **2022**, *245*, 110514. [CrossRef]
117. Pawar, K.S.; Teli, S.N.; Shetye, P.; Shetty, S.; Satam, V.; Sahani, A. Blind-spot monitoring system using LiDAR. *J. Inst. Eng. Ser. C* **2022**, *103* 1071–1082. [CrossRef]
118. Zixin, W.; Weimin, C.; Hao, H. CFD numerical simulation of a container ship turning motion. *J. Phys. Conf. Ser.* **2021**, *1834* 012020. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.