Neural Networks 152 (2022) 17–33

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Risk-based implementation of COLREGs for autonomous surface vehicles using deep reinforcement learning

Amalie Heiberg [a], Thomas Nakken Larsen [b], Eivind Meyer [c], Adil Rasheed [b,*], Omer San [d], Damiano Varagnolo [b]

[a] Equinor, Norway
[b] Department of Engineering Cybernetics, Norwegian University of Science and Technology, Norway
[c] Institute of Informatics, Technical University of Munich, Germany
[d] School of Mechanical and Aerospace Engineering, Oklahoma State University, United States of America

ARTICLE INFO

Article history:
Received 30 November 2021
Received in revised form 6 March 2022
Accepted 11 April 2022
Available online 16 April 2022

Keywords:
Deep reinforcement learning
Collision avoidance
Path following
Collision risk indices
Machine learning controller
Autonomous surface vehicle

ABSTRACT

Autonomous systems are becoming ubiquitous and gaining momentum within the marine sector. Since the electrification of transport is happening simultaneously, autonomous marine vessels can reduce environmental impact, lower costs, and increase efficiency. Although close monitoring is still required to ensure safety, the ultimate goal is full autonomy. One major milestone is to develop a control system that is versatile enough to handle any weather and encounter that is also robust and reliable. Additionally, the control system must adhere to the International Regulations for Preventing Collisions at Sea (COLREGs) for successful interaction with human sailors. Since the COLREGs were written for the human mind to interpret, they are written in ambiguous prose and therefore not machine-readable or verifiable. Due to these challenges and the wide variety of situations to be tackled, classical model-based approaches prove complicated to implement and computationally heavy. Within machine learning (ML), deep reinforcement learning (DRL) has shown great potential for a wide range of applications. The model-free and self-learning properties of DRL make it a promising candidate for autonomous vessels. In this work, a subset of the COLREGs is incorporated into a DRL-based path following and obstacle avoidance system using collision risk theory. The resulting autonomous agent dynamically interpolates between path following and COLREG-compliant collision avoidance in the training scenario, isolated encounter situations, and AIS-based simulations of real-world scenarios.

1. Introduction

Over the last few years, the promising idea of autonomous ships has gained traction through projects like ReVolt (DNV GL, 2020) and Yara Birkeland (KONGSBERG, 2020). Such research projects are increasingly incentivized as the funding bodies recognize the potential benefits of autonomy at sea. A notable example is the EU-funded four-year project Autoship Horizon 2020, which seeks to speed up the transition towards autonomous ships in the EU (Autoship, 2020). For the first time in history, the promise of lower emissions, higher efficiency, and fewer accidents via autonomy is becoming tangible.

Human error is a leading cause of accidents on the road (Dingus et al., 2016; Thomas et al., 2013), and reports show that accidents at sea are no different. According to the Annual Overview of Marine Casualties and Incidents published by the European Maritime Safety Agency (EMSA), human error was attributed to over 50% of accidental events between 2011–17 (European Maritime Safety Agency (EMSA), 2018). In addition to reducing accidents (and thereby fatalities), environmental damage, and costs, autonomous marine operations allow for optimized route planning. This optimization can be done with respect to time spent and fuel costs. Furthermore, autonomous ships can move cargo transport from the road to the sea, leading to less trafficked roads. For instance, the autonomous container ship Yara Birkeland is expected to reduce the number of trips made by diesel trucks by 40,000 a year after its launch in 2020 (Skredderberget, 2018). With the widespread electrification taking place, reduced air pollution is another likely and desirable effect.

An overall reduction of errors from introducing autonomy depends on developing robust and reliable systems, which is no trivial task. For autonomous navigation at sea, the vessel's control system must deal appropriately with a wide range of situations depending on the position of the ownship (OS) and other ships within a certain radius and environmental factors such as wind and ocean currents, and waves. Another crucial element is detection, classification, and tracking of objects, which might be

* Corresponding author.
E-mail address: adil.rasheed@ntnu.no (A. Rasheed).

https://doi.org/10.1016/j.neunet.2022.04.008

challenging in certain weather conditions. The currently proposed solutions generally make significant simplifications and assumptions. Low-level controllers, or autopilots, are already commercially available, but more research on high-level path planning and collision avoidance is needed to ensure safe autonomous navigation in real situations. For collision avoidance, compliance with the International Regulations for Preventing Collisions at Sea (COLREGs) is crucial to ensure safety when encountering other vessels.

Due to the complex nature of autonomy at sea, classical model-based methods may be challenging to implement for full autonomy. Modern machine learning (ML) methods are proficient in approximating such complex models. Supervised learning approaches are powerful but limited by their dependency on labeled training data. Reinforcement learning (RL) circumvents this by producing the training data as the decision-making agent interacts with its environment. However, there exists limited research on the combined topic of COLREG-compliant RL controllers. Therefore, this work aims to incorporate a subset of the COLREGs, directly related to collision avoidance, into an autonomous path following and collision avoidance system based on deep reinforcement learning (DRL) conditioned on measures of collision risk. Thus, the contributions of this work are as follows.

- We use state-of-the-art collision-risk theory in defining a reward-design strategy to guide an autonomous DRL agent towards COLREG compliance.
- We train a risk-based DRL agent using Proximal Policy Optimization (PPO) to solve a simultaneous path following and collision avoidance task.
- We conduct qualitative and quantitative analyses on the risk-conditioned agent's COLREG compliance in synthetic and high-risk scenarios. Lastly, we assess the agent's ability to generalize to previously unseen (simulated) real-world environments.

Section 2 describes the state-of-the-art in collision avoidance for marine guidance, in which the COLREGs are generally ignored. Section 3 introduces the COLREGs relevant for collision avoidance and essential concepts within guidance and control, collision risk theory, and DRL. Section 4 defines the simulation environments, DRL problem formulation, and evaluation methods. Section 5 presents and discusses the COLREG-compliance and general path following and collision avoidance performance of the resulting autonomous controller. Section 6 summarizes the findings and suggests future work.

2. Background

Collision alert systems (CAS) aid the captain and crew on board a marine vessel. Such systems primarily extend exteroceptive sensors, converting raw measurements into more interpretable information. Examples of CAS systems are Automatic Radar Plotting Aid (ARPA) and Automatic Identification System (AIS) (compared in Lin and Huang (2006)), routinely used for collision risk evaluation (Xu & Wang, 2014). As we move into the fourth industrial revolution, solutions such as digital twins and remote sensing are making their way into the maritime industry (DNV GL, 2019). Decision-making is thus gradually being taken from the cognitive realm and into the digital domain, and the need for highly robust and flexible guidance, navigation, and control (GNC) systems is growing. Since collision avoidance (COLAV) systems are responsible for one of the most safety-critical aspects of a vessel's operation, any GNC system operating in a dynamic environment requires a robust COLAV strategy (Aniculaesei et al., 2016). Therefore, reliable and transparent COLAV systems are crucial to reach full autonomy at sea.

All vessels above 300 tonnes engaged on international voyages, all cargo ships above 500 tonnes, and all passenger ships are required to carry an AIS (International Maritime Organization (IMO), 2020). The AIS transmits and receives information such as identity, position, course, and speed, which can be incorporated into a COLAV system. Such systems can thus enhance the quality of information about other vessels but may also depend on the communication infrastructure. However, in crowded areas, the AIS data may not update frequently enough to sufficiently provide situational awareness. Since one cannot expect complete availability, ships typically utilize additional exteroceptive sensors such as cameras, LiDARs, and RADARs. LiDARs have become particularly common in the field of autonomous driving and are often used for object classification, localization, and tracking (Mekala et al., 2021). Field-programmable gate arrays (FPGAs) are have become typical choices for deploying low-power sensor processing methods in hardware applications, e.g. in vision systems (Suresh et al., 2021). DOA tracking can utilize beamforming to provide continuous connectivity in crowded areas (Balamurugan et al., 2021). For such inter-vehicular connectivity, securing the data exchange and logs between vehicles is crucial (Kamal et al., 2021). In systems relying on exteroceptive sensing, reliable predictions on their remaining useful life (RUL) is critical for autonomous operation and maintenance scheduling (Bhargava et al., 2020). Ideally, an autonomous COLAV system uses redundant information to tackle sensor failures.

Before autonomous vessels became a possibility, the International Regulations for Preventing Collisions at Sea (COLREGs) were formulated to prevent collisions between two or more vessels (International Maritime Organization, 1972a). Although technological advancement has been significant since their publication in 1972, COLREG-compliance for autonomous vessels is still understudied. One of the main challenges is that the COLREGs were written for humans to interpret and require a translation to a machine-readable and verifiable format. Another potential challenge is the indirect communication that occurs when two vessels meet in a situation with a high risk of collision. For instance, the COLREGs require sharp maneuvers for clear communication between vessels when a high-risk situation is encountered. However, this is often not the optimal behavior from an energy efficiency (or even collision risk) point of view. So long as there may be both human and autonomous operators of marine vessels at sea, the autonomous controller should behave in a way that a human-operated vessel can interpret its intent.

In addition to the challenges inherent to the COLREGs, autonomous collision avoidance can be demanding due to the complex dynamics of ships, varying speeds, and changing environmental conditions (Tam et al., 2009). The majority of the proposed solutions for autonomy make assumptions that do not represent reality. Examples of such assumptions are the constant speed of the OS or other ships, good weather conditions, or that the system only operates while the ship is at open sea. An adequate autonomous vessel must master all the situations the current fleet handles. For instance, given sufficient situational awareness, a full-fledged autonomous COLAV system should be expected to handle situations involving all sorts of moving and stationary objects, from container ships to kayaks. For generalization, the system must track a high number of objects simultaneously and perform well in congested waters.

A plethora of COLAV algorithms and architectures for autonomous control have been, and still are, researched. Here, we distinguish between *classical* and *soft* systems (Statheros et al., 2008). Classical systems find an optimal strategy analytically and numerically from mathematical models and logic, which are typically accompanied by convergence proofs. Model predictive control (MPC) can be used to develop COLAV systems

compliant with the primary rules of COLREGs. MPC can also be applied to nonlinear systems with uncertain environmental disturbances (Soloperto et al., 2019). The Velocity Obstacle (VO) method (Fiorini & Shiller, 1998) models artificial obstacles representing the velocities that would result in a collision, and Kuwata et al. (2014) show that maritime navigation using the VO method can be COLREGs-compliant. Interval Programming (IvP), a multi-objective optimization approach, has successfully produced COLREGs-compliant COLAV systems (Benjamin et al., 2006; Woernner, 2014). Dynamic Window (DW) is an optimization-based method that has been researched for marine applications (Serigstad et al., 2018), the strength of which can be found in its focus on fast dynamics through reducing the search space to the reachable velocities within a short time interval (Fox et al., 1997).

Based on artificial intelligence (AI), *soft systems* assume that the problem is not readily quantified. Heuristics are experience-based methods for finding an acceptable solution to a problem. The A* heuristic (Hart et al., 1968) might be the most well-known and widely used soft approach; A* is a greedy search algorithm for finding the shortest distance between two nodes in a graph, in which a heuristic measure weights the edges between nodes. It is often used for high-level path and trajectory planning, as was done in Eriksen (2019). Another well-known heuristic is the genetic algorithm (GA) based on evolutionary theory. Smierzchalski (1999) applies a genetic algorithm for trajectory planning in an environment with static and dynamic obstacles. Kim et al. (2015) showed that Distributed Tabu Search, a metaheuristic method, can be used for collision avoidance in highly congested areas. Another group of soft systems is machine learning (ML). ML techniques such as deep learning (DL) and reinforcement learning (RL) have recently gotten significant attention in the context of autonomous systems and decision-making problems, as they benefit from neural networks' currently unmatched function approximation capabilities. Model-free RL methods can find a control law even without any mathematical model of the system (Silver et al., 2021). However, only a limited amount of research has been devoted to autonomous marine vessels compared to driver-less cars, for instance. In Xu et al. (2017), a deep convolutional neural network (CNN) is trained for COLREGs-compliant collision avoidance for a crewless surface vehicle. This method is based on image recognition, using the CNN's ability to process spatially structured data. The Deep Deterministic Policy Gradient (DDPG) algorithm has demonstrated successful path following and simple collision avoidance for marine vessel models (Martinsen, 2018; Martinsen & Lekkas, 2019; Vallestad, 2019). In addition, Meyer, Robinson et al. (2020) and Zhao and Roh (2019) showcased the Proximal Policy Optimization (PPO) algorithm for multi-ship collision avoidance.

Alternatively, COLAV systems can be classified as *deliberative* or *reactive* systems (Siciliano & Khatib, 2008). Deliberative systems work in a "sense-plan-act" fashion. Intuitively, reactive systems are then considered "sense-act" systems. Hybrid COLAV systems emerge when combining different system categories, e.g., deliberate and reactive systems. This approach is made with increasing frequency (Ding et al., 2011). Multi-layered systems are also being developed, where each subsystem lies on a spectrum between reactive and deliberative. Such hybrid architectures are able to harvest the strengths of several methods, using each where they perform best. Loe (2008) applies a two-layered approach where deliberation is done by a Rapidly-Exploring Random Tree (RRT) algorithm combined with the deliberative A* heuristic, and the reactive component consists of a modified DW algorithm. In Eriksen (2019), A* is combined with a mid-layer and a reactive MPC-based algorithm, forming a three-layered COLAV system. Casalino et al. (2009) and Svec et al. (2013) have proposed similar layered architectures. Ultimately, traditional model-based approaches are applicable for COLAV problems; however, they require a dynamics model. Deriving a dynamics model, when possible, is often a time-consuming process, and the resulting model may not be computational in real-time without loss of generality or accuracy.

In summary, a wide range of COLAV systems have been proposed in literature, generally disregarding the COLREGs. At the same time, the increased focus on autonomous systems in later years requires COLREG-compliance for sufficient safety. Deriving a dynamics model and control law that simultaneously considers path following, COLAV, and COLREGs is an infeasible task using traditional control methods. Therefore, we argue that a model-free method is preferable to a model-based one in this application. This gap combined with the promise of DRL for autonomous navigation, shapes the objective of the article — to investigate COLREG-compliance in a path following and collision avoidance system based on deep reinforcement learning, conditioned on measures of collision risk.

3. Theory

3.1. Dynamics of a marine vessel

The dynamical model considered in this work is CyberShip II: a 1:70 scale replica of a supply ship (Skjetne et al., 2004b). This model is simulated in a calm ocean surface environment with the following assumptions.

Assumption 1 (*State Space Restriction*)**.** The vessel is always located on the surface, and thus there is no heave motion. Also, there is no pitching or rolling motion.

Assumption 2 (*Calm Sea*)**.** There are no external disturbances to the vessel, such as wind, ocean currents, or waves.

Following SNAME notation (SNAME, The Society of Naval Architecture and Marine Engineers, 1950), the navigation state vector then consists of the generalized coordinates, $\boldsymbol{\eta} = [x^n, y^n, \psi]^T$, where x^n and y^n are the North and East positions, respectively, in the North-East-Down (NED) reference frame $\{n\}$, and ψ is the yaw angle, i.e., the current angle between the vessel's longitudinal axis x_b and the North axis x_n, illustrated by Fig. 1. Correspondingly, the translational and angular velocity vector $\boldsymbol{\nu} = [u, v, r]^T$ consists of the surge (i.e., forward) velocity u, the sway (i.e., sideways) velocity v and yaw rate r.

3.1.1. Vessel model

Given the established assumptions, the 3-DOF vessel dynamics can be expressed in a compact matrix–vector form

$$\dot{\boldsymbol{\eta}} = \mathbf{R}_{z,\psi}(\boldsymbol{\eta})\boldsymbol{\nu}$$
$$\mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} = \mathbf{B}\boldsymbol{f},$$

where $\mathbf{R}_{z,\psi}$ represents a rotation of ψ radians around the z_n-axis as defined by

$$\mathbf{R}_{z,\psi} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Furthermore, $\mathbf{M} \in \mathbb{R}^{3\times3}$ is the mass matrix and includes the effects of both rigid-body and added mass, $\mathbf{C}(\boldsymbol{\nu}) \in \mathbb{R}^{3\times3}$ incorporates centripetal and Coriolis effects, and $\mathbf{D}(\boldsymbol{\nu}) \in \mathbb{R}^{3\times3}$ is the damping matrix. Finally, $\mathbf{B} \in \mathbb{R}^{3\times2}$ is the actuator configuration matrix. The numerical values of the matrices are found in Skjetne et al. (2004a), where the model parameters were estimated experimentally for CyberShip II in a marine control laboratory.

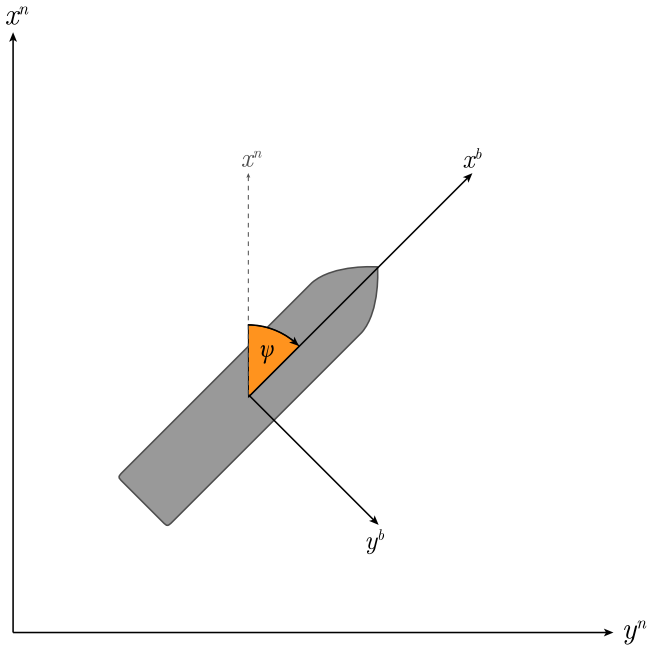


Fig. 1. Illustration of the NED and body coordinate frames. Here, y^n, x^n point in the North and East directions, respectively, and thus describes a NED inertial frame. The body frame's origin is positioned at the vessel's center of mass and rotated by ψ degrees to align x^b with the vessel's longitudinal axis.

We disregard the ship's bow thruster and allow only the aft thrusters and control surfaces to be applied by the Reinforcement Learning (RL) agent as control signals. This omission simplifies the RL agent's action space and is further motivated by the bow thrusters' limited effectiveness at higher speeds (Sørensen et al., 2017). Thus, the control vector, $\boldsymbol{f} = [T_u, T_r]^T$, consists of the surge force input, T_u, and the yaw's moment input, T_r.

3.1.2. COLREG rules

Among the 41 rules in the International Regulations for Preventing Collisions at Sea (Organization, 1972b), only the directly relevant rules for COLAV are considered. Appendix describes rules 6, 8, and 14–16. The two main takeaways from these rules are that (1) the give-way vessel should take early and substantial action, and (2) safe speed should be ensured at all times, such that course alteration is effective towards avoiding collisions where there is sufficient sea-room. Since rules 6 and 8 are particularly tough to quantify, this work focuses on compliance to rules 14–16.

3.2. Measures of collision risk

The rules presented above are intended for human interpretation and contain ambiguities such as "large enough" (Rule 8) and "substantial action" (Rule 16). How can they be translated into a form suitable for reinforcement learning? An essential first step is recognizing the relationship between the COLREGs and collision risk. The COLREGs are in place to reduce collision risk and indirectly affect the risk level by influencing the probable behavior of the target ship (TS). Since there is a correlation between the rules and the risk level, employing a measure of risk as a proxy for the COLREGs may enable the RL agent to learn COLREG-compliant behavior.

By analyzing the historical trends of measuring collision risk, three main developments can be observed (Xu & Wang, 2014): traffic flow theory, ship safety domains, and collision risk indices. The initial efforts to quantify collision risk were based on *traffic flow theory*, a method built on empirical studies and statistical traffic analysis in specific waters. For instance, Cockcroft (1981) investigated the collision rates for ships of varying tonnage relative to their position in a water area. Goodwin (1978) took it further and studied the rate of dangerous encounters. As statistical analysis of historical data was deemed insufficient for dynamic collision avoidance, *ship safety domains* were introduced. The ship safety domain defines a region around the ship in question that other ships should not enter. Hence, there is a risk of collision if one ship is inside the safety domain of another, and the ship domain can be said to be a generalization of a safe distance (Szlapczynski & Szlapczynska, 2017). When applying the ship domain to an encounter situation in order to determine risk, one of the four *safety criteria* are normally used: (1) the OS domain should not be violated by a TS, (2) a TS domain should not be violated by the OS, (3) neither of the ship domains should be violated, or (4) ship domains should not overlap, such that they remain mutually exclusive. Rawson et al. (2014) and Wang and Chin (2016) use the latter criterion of non-overlapping ship domains.

It is important to note that a ship domain is usually defined depending on which situation the ship finds itself in to respect the COLREGs. For instance, the domain used while the OS is overtaking another ship is symmetrical, with its origin coinciding with the center of the OS. Conversely, the origin is shifted to the right in a head-on situation, as close encounters on the starboard side should be avoided.

Davis et al. (1980) expanded the theory of ship safety domains in their well-known work on ship arenas. The ship arena defines the distances around the OS at which action should be made to avoid a dangerous encounter and is, therefore, larger than the ship safety domains proposed initially. In addition to the OS's length and velocity, the distance to the closest point of approach (DCPA) and the time to the closest point of approach (TCPA) are used to construct the limits of the ship arena. A geometrical representation of DCPA and TCPA are presented in Fig. 2, giving rise to the equations

$$DCPA = R\sin(\chi_R - \chi_{OS} - \theta_T - \pi) \tag{1}$$

and

$$TCPA = \frac{R}{V_R}\cos(\chi_R - \chi_{OS} - \theta_T - \pi) \tag{2}$$

where R is the absolute distance between the OS and TS, and V_R and χ_R are the relative speed and course between them. In addition, χ_{OS} is the course of the OS, while θ_T is the bearing of the TS relative to the OS.

This leads to the subsequent development in collision risk evaluation, namely *collision risk indices* (CRIs), which are primarily based on the DCPA and TCPA. In addition, a CRI can include the absolute distance from the OS to the TS R, velocity ratio K of two encountering ships, relative course χ_R, and other key features. Recently, simple CRIs alone are considered unable to capture collision risk's gradual and complex nature. As a result, combining the CRI with fuzzy logic or the fuzzy comprehensive evaluation method has become the norm. In fuzzy logic, fuzzy IF-THEN rules are applied to the parameters involved, such as DCPA and TCPA, to determine the risk level. In the fuzzy comprehensive evaluation method, on the other hand, *membership functions*, $u(\cdot) \in [0, 1]$, are used instead of IF-THEN rules, taking more details into account. The final CRI is then given as the weighted sum of the membership function outputs, as exemplified below:

$$\begin{aligned} CRI = {} & \alpha_{DCPA} \cdot u_{DCPA}(DCPA) \\ & + \alpha_{TCPA} \cdot u_{TCPA}(TCPA) \\ & + \alpha_R \cdot u_R(R) \end{aligned} \tag{3a}$$

$$\alpha_{DCPA} + \alpha_{TCPA} + \alpha_R = 1 \tag{3b}$$

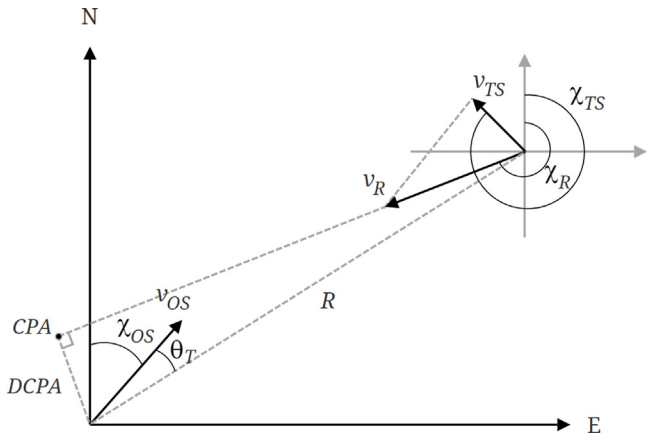


Fig. 2. Geometric representation of CPA and DCPA. Here, the axes correspond to the North and East directions in the NED frame, respectively.

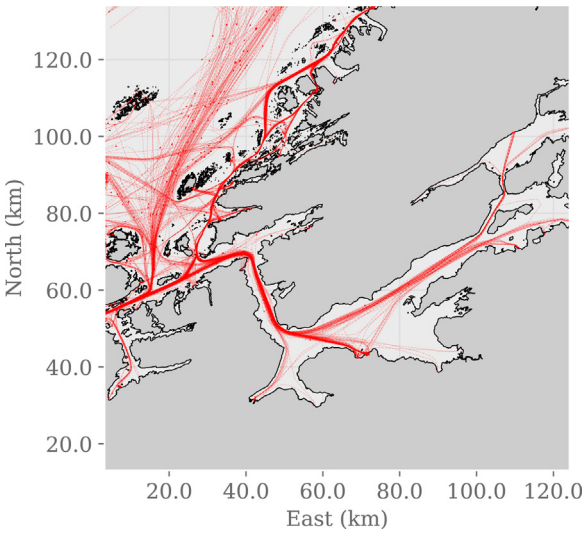


Fig. 3. Snapshot of the marine traffic from 01.01.2020 to 06.02.2020 in the Trondheim fjord, based on AIS data. Each red line represents a recorded travel.

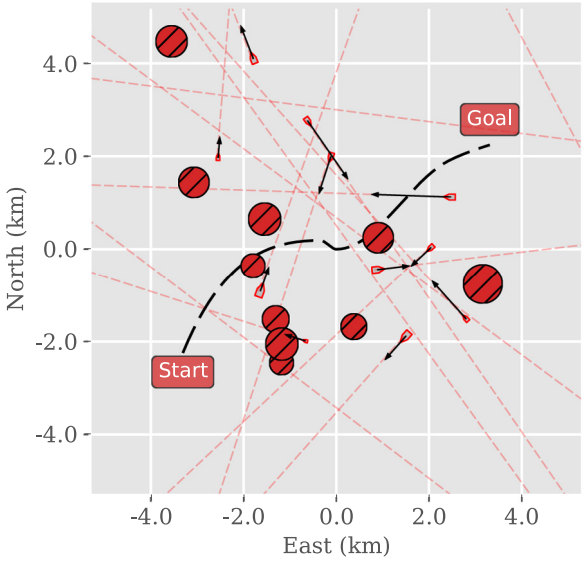


Fig. 4. Random sample of the stochastically generated path following training scenario with moving obstacles. The circles are static obstacles representing landmasses, and the vessel-shaped objects are moving according to the trajectory lines and velocity vectors.

3.3. Deep reinforcement learning

Model-free reinforcement learning (RL) methods train a decision-making agent through trial and error, where the agent is gathering experience from an environment supplying only a situational observation state and a corresponding reward. Applications of RL on high-dimensional, continuous control tasks heavily rely on function approximators to generalize over the state space. Even if classical, tabular solution methods such as Q-learning can be made to work (provided a discretizing of the continuous action space), this is not considered an efficient approach for control applications (Lillicrap et al., 2015). In recent years, given their remarkable generalization ability over high-dimensional input spaces, the dominant approach has been the application of deep neural networks optimized using gradient methods. Several algorithms built on this principle have gained significant traction in the RL research community, most notably Deep Deterministic Policy Gradient (DDPG) (Lillicrap et al., 2015), Asynchronous Advantage Actor Critic (A3C) (Mnih et al., 2016), Proximal Policy Optimization (PPO) (Schulman et al., 2017), and Soft Actor–Critic (SAC) (Haarnoja et al., 2017). For continuous control tasks, this family of policy gradient methods is commonly considered the more efficient approach (Tai et al., 2016). Based on previous work, where the PPO algorithm significantly outperformed other methods on a learning problem similar to the one covered in this work (Larsen et al., 2021; Meyer, Robinson et al., 2020), we focus our efforts on this method.

4. Methodology

4.1. Training environment

DRL-based autonomous agents have a remarkable ability to generalize their policy over the observation space, including the domain of unseen observations. Moreover, given the complexity and heterogeneity of the Trondheim Fjord environment, with archipelagos, shorelines, and skerries (see Fig. 3), this ability will be fundamental to the agent's performance. However, the training environment in which the agent evolves from a blank slate to an intelligent vessel controller must be representative, challenging, and unpredictable to facilitate the generalization. If not for the generalization issues associated with this approach (Codevilla et al., 2019), it would also allow the agent to train via behavior cloning based on historical AIS data. However, given the resolution of our terrain data, the resulting obstacle geometry is typically very complex, leading to overly high computational demands for simulating the functioning of the distance sensor suite. Moreover, the agent's perceptive observation space (Section 4.2.2) undergoes significant dimensionality reduction, resulting in the agent not benefiting from such high-frequency details in the simulation. Thus, the better choice is to craft an artificial training scenario with simple obstacle geometries. To reflect the dynamics of a real-world marine environment, we let the stochastic initialization method of the training scenario spawn other target vessels with deterministic, linear trajectories. Additionally, circular obstacles scattered around the environment substitute the real-world terrain. Fig. 4 illustrates an instantiation of the training environment.

4.2. Observation vector

To facilitate the learning of a decision-making policy, the RL agent requires an observation vector, s, containing sufficient

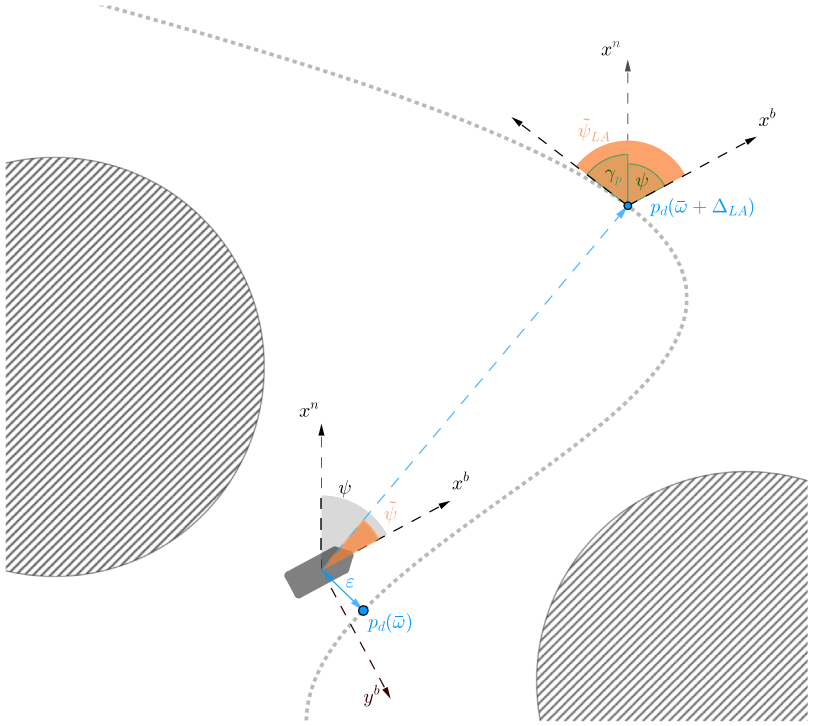


Fig. 5. Illustration of key path-following concepts in vessel guidance and control. The path reference point, $\boldsymbol{p}_d(\bar{\omega})$, describes the point on the path with the closest Euclidean distance to the vessel, while the look-ahead reference point, $\boldsymbol{p}_d(\bar{\omega} + \Delta_{LA})$, is located a distance, Δ_{LA}, further along the path.

information about the vessel's state relative to the path in addition to situational sensor information. The complete observation vector is then constructed by concatenating navigation-based and perception-based features, which formally translates to $s = [s_n, s_p]^T$. In the context of this paper, we consider the term *navigation* as the characterization of the vessel's state, i.e., its position, orientation, and velocity, with respect to the desired path. On the other hand, *perception* refers to the observations made via the rangefinder sensor measurements. In the following, the path navigation feature vector, s_n, and the perceptive feature vector, s_p, are covered in detail.

4.2.1. Navigation features

A sufficiently information-rich path navigation feature vector would be such that it, on its own, could facilitate a satisfactory path-following controller. A few concepts often used in vessel guidance and control are helpful to formalize this. First, we introduce the mathematical representation of the parameterized path, which is expressed as

$$\boldsymbol{p}_d(\omega) = [x_d(\omega), y_d(\omega)]^T \tag{4}$$

where $x_d(\omega)$ and $y_d(\omega)$ are defined in the NED frame. Navigating the path necessitates a reference point, which is continuously updated based on the vessel's position. We define this reference point as the point on the path that has the closest Euclidean distance to the vessel, given its current position, as illustrated in Fig. 5. To find this, we calculate the corresponding value of the path variable $\bar{\omega}$ at each time step. This is an equivalent problem formulation because the path is defined implicitly by the value of ω. Formally, this translates to the optimization problem

$$\bar{\omega} = \arg\min_{\omega} \left(x^n - x_d(\omega)\right)^2 + \left(y^n - y_d(\omega)\right)^2, \tag{5}$$

which, using the Newton–Raphson method, can be calculated accurately and efficiently at each time step. We define the corresponding Euclidean distance to the path, i.e., the deviation between the desired path and the current track, as the cross-track error (CTE) ϵ. Formally, we thus have that

$$\epsilon = \left\| \left[x^n, y^n\right]^T - \boldsymbol{p}_d(\bar{\omega}) \right\|. \tag{6}$$

Next, we consider the look-ahead point, $\boldsymbol{p}_d(\bar{\omega} + \Delta_{LA})$, to be the point that lies a constant distance further along the path from the reference point $\boldsymbol{p}_d(\bar{\omega})$. Look-ahead based steering, i.e., setting the look-ahead point direction as the desired course angle, is a commonly used guidance principle (Fossen, 2011). The look-ahead distance, Δ_{LA}, is set by the user and controls how aggressively the vessel should reduce the distance to the path.

We then define the heading error, $\tilde{\psi}$, as the change in heading needed for the vessel to navigate straight towards the look-ahead point from its position, as illustrated in Fig. 5. Formally, $\tilde{\psi}$ is defined as

$$\tilde{\psi} = \text{atan2}\left(\frac{y_d(\bar{\omega} + \Delta_{LA}) - y^n}{x_d(\bar{\omega} + \Delta_{LA}) - x^n}\right) - \psi, \tag{7}$$

where ψ is the vessel's heading and x^n, y^n are the NED-frame vessel coordinates as defined earlier.

However, even if minimizing the heading error will yield good path adherence, taking into account the path direction at the look-ahead point might improve the smoothness of the resulting vessel trajectory. Referring to the first-order path derivatives as $x'_p(\bar{\omega})$ and $y'_p(\bar{\omega})$, we have that the path angle, γ_p, in general, can be expressed as a function of arc-length, ω, such that

$$\gamma_p(\bar{\omega}) = \text{atan2}\left(y'_p(\bar{\omega}), x'_d(\bar{\omega})\right). \tag{8}$$

As visualized in Fig. 5, the path direction at the look-ahead point is then given by $\gamma_p(\bar{\omega} + \Delta_{LA})$. We then define the look-ahead heading error, which is zero in the case when the vessel is heading in a direction that is parallel to the path direction at the look-ahead point, as

$$\tilde{\psi}_{LA} = \gamma_p(\bar{\omega} + \Delta_{LA}) - \psi \tag{9}$$

Table 1
Path-following feature vector s_n at timestep t.

Feature	Definition
Surge velocity	$u^{(t)}$
Sway velocity	$v^{(t)}$
Yaw rate	$r^{(t)}$
Cross-track error	$\epsilon^{(t)}$
Heading error	$\tilde{\psi}^{(t)}$
Look-ahead heading error	$\tilde{\psi}_{LA}^{(t)}$

Our assumption is then that the navigation feature vector s_n, defined as outlined in Table 1, should provide a sufficient basis for the agent to intelligently adhere to the desired path. The navigation features are then formally defined as

$$s_n^{(t)} = \left[u^{(t)}, v^{(t)}, r^{(t)}, \epsilon^{(t)}, \tilde{\psi}^{(t)}, \tilde{\psi}_{LA}^{(t)}\right]^T. \tag{10}$$

4.2.2. Perception features

Using a set of rangefinder sensors as the basis for obstacle avoidance is a natural choice, as it yields a comprehensive yet intuitive representation of any neighboring obstacles. This configuration should also enable a relatively straightforward transition from the simulated environment to a real-world one, given that rangefinder sensors such as lidars, radars, sonars, or depth cameras are commonly used

In our setup, the vessel is equipped with N distance sensors with a maximum detection range of S_r, distributed uniformly with 360° coverage. While the area behind the vessel is obviously of lesser importance, e.g., unnecessary to consider when navigating purely static terrain, the possibility of overtaking situations where the agent must react to another vessel approaching from behind makes full sensor coverage necessary. The most natural approach to constructing the final observation vector would be to concatenate the path information feature vector with the array of sensor outputs. However, initial experiments with this approach resulted in the training process stagnating at an unsatisfactory agent performance level. A likely explanation for this failure is the size of the observation vector, which was fed to the agent's fully connected policy and value networks; as the input size becomes large, the agent suffers from the well-known *curse of dimensionality*. Due to the resulting network complexity and the exponential relationship between the dimensionality and volume of the observation space, the agent fails to generalize new, unseen observations intelligently (Goodfellow et al., 2016). An obvious solution is to reduce the observation space's dimensionality significantly. However, simply reducing the resolution is infeasible, as this would accordingly degrade the agent's situational awareness.

In this work, we partition the sensor suite into D sectors, each of which produces a scalar measurement included in the final observation vector, effectively summarizing the local sensor readings within the sector. However, given our desire to minimize its dimensionality, dividing the sensors into sectors of uniform size is sub-optimal as obstacles located in front of the vessel are significantly more critical and thus require higher perceptive accuracy than those located at its rear. In order to realize such a non-uniform partitioning, we use a logistic function — a choice that also fulfills our general preference for symmetry. Assuming a counter-clockwise ordering of sensors and sectors starting at the rear of the vessel, we map a given sensor index, $i \in \{1, \dots, N\}$, to a sector index, $k \in \{1, \dots, D\}$, according to

$$\kappa : i \mapsto \kappa(i) = \left\lfloor \underbrace{D\sigma\left(\frac{\gamma_C i}{N} - \frac{\gamma_C}{2}\right)}_{\text{Non-linear mapping}} - \underbrace{D\sigma\left(-\frac{\gamma_C}{2}\right)}_{\text{Constant offset}} \right\rfloor, \tag{11}$$

where σ is the logistic sigmoid function, and γ_C is a scaling parameter controlling the density of the sector distribution such that decreasing it will yield a more evenly distributed partitioning. We can then formally define the distance measurement vector for the kth sector, which we denote by $\boldsymbol{w}_k$, according to

$$\boldsymbol{w}_{k,i} = x_i \quad \text{for } i \in \{1, \dots, N\} \text{ such that } \kappa(i) = k$$

Next, we select a mapping $f : \mathbb{R}^n \mapsto \mathbb{R}$, which takes the vector of distance measurements $\boldsymbol{w}_k$, for an arbitrary sector index k, as input, and outputs a scalar value based on the current sensor readings within that sector. The *feasibility pooling* procedure, introduced in Meyer, Robinson et al. (2020), calculates the maximum reachable distance within each sector, taking into account the obstacle sensor readings' location and the vessel's width. This method iterates over the sector's distance measurements in ascending order and checks whether it is feasible for the vessel to advance beyond this level. As soon as the broadest available opening within a distance level is deemed too narrow given the vessel's width, the maximum reachable distance has been reached. Formally, we define f as the feasibility pooling algorithm, and the resulting perceptive distance observation is summarized in Fig. 6. To finalize the processing of distance measurements, we introduce the concept of *closeness*. An obstacle's closeness is zero if it is at a distance further than S_r away from the vessel and unity if the vessel has collided with the obstacle. Furthermore, within this range, is it reasonable to map distance to closeness in a logarithmic fashion, such that, following human intuition, the difference between 10m and 100m is more significant than the difference between, for instance, 510m and 600m. Formally, the maximum reachable distance, d, maps to closeness, $c(d) : \mathbb{R} \mapsto [0, 1]$, according to

$$c(d) = \text{clip}\left(1 - \frac{\log(d+1)}{\log(S_r+1)},\ 0,\ 1\right). \tag{12}$$

Ultimately, the choice of D is a user-defined hyperparameter with differing optimal values depending on the environment and traffic complexity. In this work, we choose $D = 9$ as used in Meyer, Robinson et al. (2020).

4.2.3. Motion detection

The maximum reachable distance in a sector may equal the maximum sensor range even though there is an obstacle in that sector. Thus, by applying the feasibility pooling algorithm to reduce the dimensionality of the rangefinder suite, the resulting closeness observation may fail to inform the RL agent about nearby obstacles. To make the agent aware of nearby moving obstacles, we incorporate the velocities of the nearest obstacle in each sector into the observation vector. Admittedly, while this implementation is trivial in a simulated environment, a real-world implementation will necessitate a reliable way of estimating obstacle velocities based on sensor data. However, even if this can be challenging due to uncertainty in the sensor readings, object tracking is a well-researched computer vision discipline. We reserve the implementation of such a method to future research but refer the reader to Granstrom et al. (2016) for a comprehensive overview of the current state of this field.

Specifically, the decomposition, which yields the x and y component of the obstacle velocity, considers the coordinate frame in which the y-axis is parallel to the centerline of the sensor sector in which the obstacle is present. Thus, we provide the decomposed velocity of the closest moving obstacle within each sector as features for the agent's observation vector. For each sector k, we denote the corresponding decomposed x and y velocities as $v_{x,k}$ and $v_{y,k}$, respectively. Naturally, if no moving obstacles are present within the sector, both components are zero.

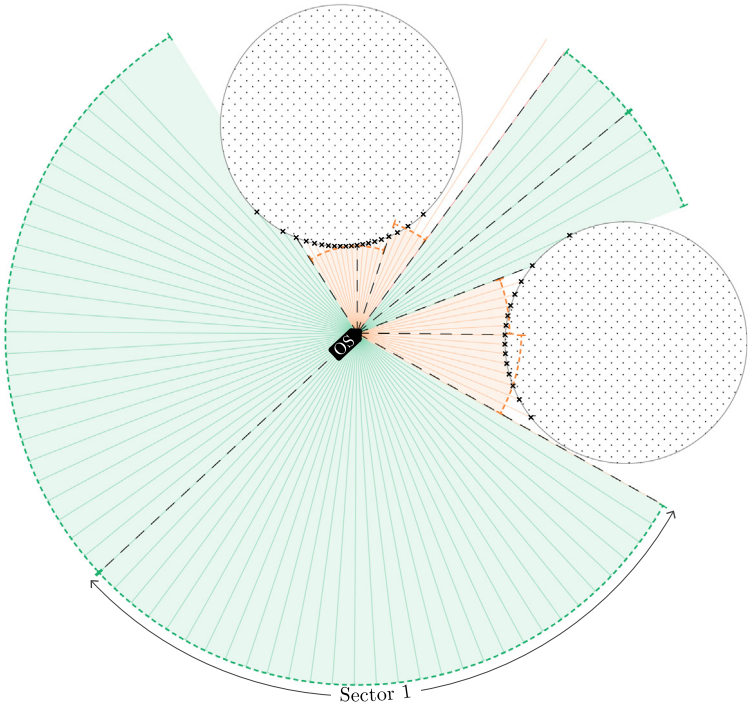


Fig. 6. The ownship (OS) features an onboard rangefinder sensor suite containing N distance sensors, partitioned into D sectors (separated by black dashed lines) according to the mapping function κ. The dashed edges illustrate the maximum reachable distance in each sector, as calculated by the feasibility pooling algorithm. The perceptive distance component of the RL-agent's observation space consists of the closeness mapping of these distances.

4.2.4. Perception state vector

By concatenating the closeness of the maximum reachable distance and the decomposed obstacle velocity for each sector, we then define the perception state vector, s_p, as

$$s_p^{(t)} = \left[\underbrace{c\left(\left(\boldsymbol{w}_1^{(t)}\right)\right),\ v_{x,1}^{(t)},\ v_{y,1}^{(t)}}_{\text{First sector}},\ \dots \right]^T. \tag{13}$$

4.3. Risk-based implementation of COLREGs

In model-free RL, the trained agent will assume a policy that maximizes the expected reward. To lead this policy to adhere to the COLREGs, we must incorporate them into the reward function. As previously mentioned, the rules are ambiguous and cannot be implemented explicitly. Instead, we use collision risk indices (CRIs) as analogs, and the following motivates how they are intended to guide the RL agent towards COLREG-compliance.

4.3.1. Risk-based reward function

Building on the theory presented in Section 3.2, a collision risk index (CRI) is calculated using fuzzy evaluation. Here, this translates to a weighted sum of evaluated risk factors, a method described in detail in Section 4.3.2. This method encapsulates collision risk's continuous and fuzzy nature, making it a convincing choice for translating the COLREGs into a DRL-based framework. Collision risk is typically only applied to encounter situations between two dynamic objects, and the collision risk index to be presented here is no exception. Thus, the reward components for path following, static obstacle avoidance, collision penalty, and living penalty must be defined separately. The corresponding components from a previous approach (Meyer, Heiberg et al., 2020) are applied here due to the excellent path following and obstacle avoidance results. The reward components for path following and static obstacle avoidance are given in Eqs. (14) and (15), while the collision and living penalties are negative constants. As a result, the total reward function has the same structure, reiterated in Eq. (16), except for a risk-based penalty for dynamic obstacles ($r_{colav,dyn}$).

$$r_{path}^{(t)} = \underbrace{\left(\frac{u^{(t)}}{U_{max}}\cos\tilde{\psi}^{(t)} + \gamma_r\right)}_{\text{Velocity-based reward}} \underbrace{\left(\exp\left(-\gamma_\epsilon|\epsilon^{(t)}|\right) + \gamma_r\right)}_{\text{CTE-based reward}} - \gamma_r^2 \tag{14}$$

$$r_{colav,stat}^{(t)} = -\frac{\sum_{i=1}^{N} \frac{\alpha_x}{1+\gamma_{\theta,stat}|\theta_i|}\exp\left(-\gamma_x x_i\right)}{\sum_{i=1}^{N} \frac{1}{1+\gamma_{\theta,stat}|\theta_i|}} \tag{15}$$

$$r = \begin{cases} r_{collision}, & \text{if collision} \\ \lambda r_{path} + (1-\lambda)\, r_{colav} + r_{exists}, & \text{otherwise} \end{cases} \tag{16}$$

The penalty for dynamic obstacles makes part of the overall penalty for collision avoidance, denoted r_{colav} and given by

$$r_{colav} = r_{colav,dyn} + r_{colav,stat}. \tag{17}$$

For every TS within the OS's sensor range, a collision risk index (CRI) $\in [0, 1]$ is calculated (see Section 4.3.2). Since the CRI increases proportionally to collision risk, it can be used semi-directly in the reward function. By multiplying the CRI_i of each target vessel, i, with a scaling factor $\beta_{CRI} > 0$, the penalty level can be weighted relative to the rest of the reward function:

$$r_{colav,dyn} = -\sum \beta_{CRI} \cdot CRI_i \tag{18}$$

4.3.2. Calculating the collision risk index

In order to determine the collision risk in an encounter situation, one must first define what constitutes a collision risk and how much each risk factor contributes to the overall risk. The state-of-the-art methods of computing CRIs generally use fuzzy evaluation (Xu & Wang, 2014), making it a natural choice here too. In short, three steps should be followed:

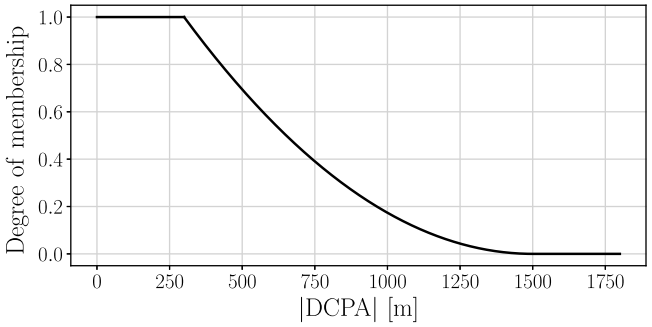


Fig. 7. Membership function for DCPA with $d_L = 320$ m and $d_U = 1500$ m.

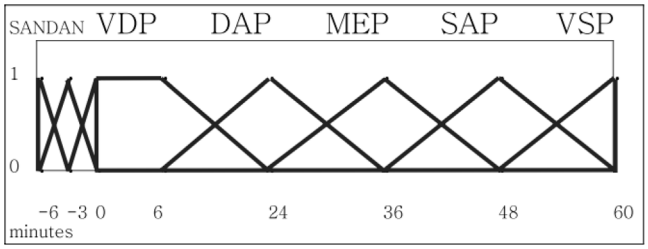


Fig. 8. Membership function for TCPA employed in Park et al. (2006). SAN = Safe Negative, DAN = Dangerous Negative, VDP = Very Dangerous Positive, DAP = Dangerous Positive, MEP = Medium Positive, SAP = Safe Positive, VSP = Very Safe Positive.

1. Define individual risk factors.
2. Define membership functions.
3. Design overall CRI as a function of membership functions.

The chosen risk factors and their membership functions are elaborated on in the following, leading up to the CRI function design.

A common starting point for defining risk is looking at the distance and time to the point of closest approach, denoted DCPA and TCPA. As the descriptive name suggests, the closest point of approach (CPA) is the closest point relative to the OS that the TS in question will come, given that the relative course and relative velocity between the two ships stay the same. The DCPA, then, is the distance to the CPA, while the TCPA is the time until the TS arrives at the CPA. Put differently, the DCPA quantifies the severity of a potential collision situation, while the TCPA quantifies its urgency. When determining the risk level associated with them, it is customary to employ upper and lower bounds for these quantities, denoted d_L and d_U for DCPA, and t_L and t_U for TCPA. Doing so, the membership functions u_{DCPA} and u_{TCPA} output unity (highest risk level) whenever $|\text{DCPA}| \leq d_L$ and $|\text{TCPA}| \leq t_L$, respectively. Conversely, their outputs are zero when $|\text{DCPA}| \geq d_U$ and $|\text{TCPA}| \geq t_U$. As was done in Gang et al. (2016), a second-order function is used between the two extremities. Chen et al. (2014) use a sinusoidal function instead. Although the latter has the virtue of being smooth, it was deemed inexpedient due to the large outputs for a wide interval of values, overshadowing other elements of the CRI. Since the sensor range used in this work is relatively short (1500 m), the steeper second-order function improved learning. It is worth noting that the sinusoidal function may be more suited in a setup with fewer obstacles and vessels where AIS data from a larger region is used.

The values for the lower and upper bounds depend largely on the application. In general, d_L defines the minimal safe encounter distance, and d_U is the absolute safe encounter distance (Gang et al., 2016). For DCPA, the membership function is defined as

$$u_{DCPA} = \begin{cases} 1 & \text{if } |DCPA| \leq d_L \\ 0 & \text{if } |DCPA| \geq d_U \\ \left(\frac{d_U - |DCPA|}{d_U - d_L}\right)^2 & \text{otherwise} \end{cases} \tag{19}$$

with d_L and d_U as positive integers. The DCPA membership function is presented graphically in Fig. 7.

For the bounds on TCPA, the method used in Gang et al. (2016) and presented in Eq. (20) is employed. Doing so adjusts the output of u_{TCPA} according to the distance between the OS and TS, accurately presenting the high risk when the distance is below or close to the lower bound d_L and low risk when it is closer to the upper bound d_U. It is assumed that DCPA never exceeds d_U, meaning that d_U is set to the maximum detectable DCPA.

$$t_L = \begin{cases} \frac{\sqrt{d_L^2 - DCPA^2}}{v_R} & \text{if } |DCPA| \leq d_L \\ \frac{d_L - DCPA}{v_R} & \text{if } |DCPA| > d_L \end{cases} \tag{20a}$$

$$t_U = \frac{\sqrt{d_U^2 - DCPA^2}}{v_R} \tag{20b}$$

In Gang et al. (2016), equal importance has been given to positive and negative values of TCPA through the membership function below:

$$u_{TCPA} = \begin{cases} 1 & \text{if } |TCPA| \leq t_L \\ 0 & \text{if} |TCPA| \geq t_U \\ \left(\frac{t_U - |TCPA|}{t_U - t_L}\right)^2 & \text{else} \end{cases} \tag{21}$$

However, noting that negative values of TCPA indicate that the OS and TS have passed each other, it makes sense to pay attention to the sign of TCPA. This is supported by Park et al. (2006), where a fuzzy case-based reasoning system for collision avoidance is proposed. In their work, the TCPA membership function in Fig. 8 is applied, indicating the significantly higher risk associated with positive values of TCPA.

Following this line of reasoning, a distinction between positive and negative values of TCPA is made according to Eq. (22). The cut-off value for negative values (negative limit) was chosen as $t_{NL} = \frac{d_L}{v_R}$, such that the degree of membership is larger than zero whenever the OS is less than t_{NL} time steps away from the TS. The membership function for TCPA is plotted in Fig. 9.

$$u_{TCPA} = \begin{cases} \begin{cases} 1 & \text{if } TCPA \leq t_L \\ 0 & \text{if } TCPA \geq t_U \\ \left(\frac{t_U - TCPA}{t_U - t_L}\right)^2 & \text{else} \end{cases} & \text{if } TCPA \geq 0 \\ \begin{cases} 0 & \text{if } TCPA \leq t_L \\ \left(\frac{t_{NL} - |TCPA|}{t_{NL}}\right)^2 & \text{else} \end{cases} & \text{if } TCPA < 0 \end{cases} \tag{22}$$

Further, the collision risk depends on the position of the TS relative to the OS, which can be expressed through the absolute distance, R, between them and the bearing angle of the TS, θ_T. Since the risk is higher on the starboard side of the OS, as expressed in Rule 14 (head-on situation) of the COLREGs, the membership functions should be designed with a bias on that side. Inspired by Davis et al. (1980), it is customary to introduce a bias of 19° starboard. Davis developed the concept of ship arena, briefly described in Section 3.2, and designed a scaling of the

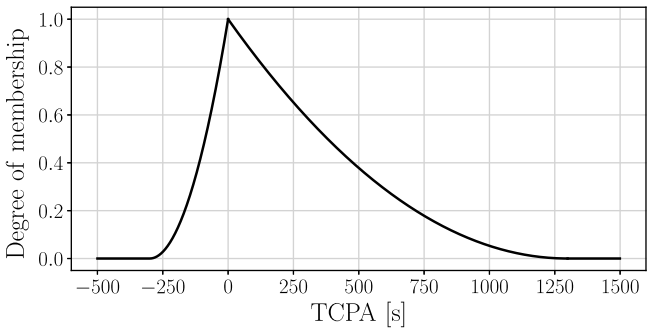


Fig. 9. Membership function for TCPA with $d_L = 320$ m, $d_U = 1500$ m and $v_R = 1$ m/s.

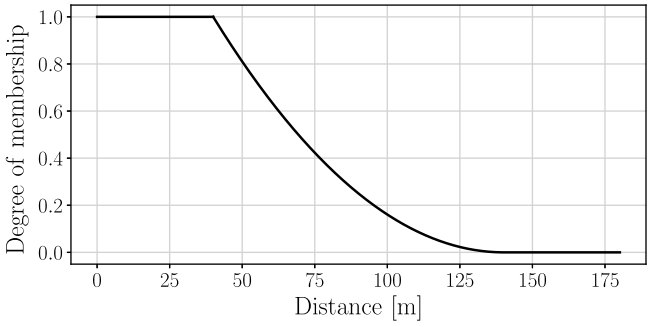


Fig. 10. Membership function for distance to the target ship, with $\theta_T = 0°$.

upper bound:

$$R_D = 1.7\cos\left(\theta_T \frac{\pi}{180} - 19°\right) + \sqrt{\left(4.4 + 2.89\cos^2\left(\theta_T \frac{\pi}{180} - 19°\right)\right)}, \tag{23}$$

while the lower bound is usually 12 times the OS length L_{pp} (Gang et al., 2016) but set to $8L_{pp}$ here due to the smaller scale. Initially, the upper bound given by R_D was implemented, but it quickly became apparent that adjustments had to be made to ensure that the agent received sufficiently negative reward when approaching TSs, regardless of their bearing angle. The difference in scaling of 4.4 times for ships detected at 19° and 161° (180° − 19°) was too large considering the relatively densely populated training and testing scenarios and a restricted sensor range of 1500 m. Through testing, it was observed that the distance membership function could be made uniform while still preserving the correct behavior in head-on situations as long as the membership function for the bearing angle, θ_T, was given enough weight. As a result, the lower and upper bounds for the absolute distance, R, were chosen as

$$R_L = \beta_{RL} L_{pp} \tag{24a}$$

$$R_U = \beta_{RU} L_{pp} \tag{24b}$$

with β_{RL} and β_{RU} chosen as appropriate scaling constants (see Fig. 10).

Following the logic applied to the membership functions for TCPA and DCPA, we arrive at the following membership function for the absolute distance between the OS and TS:

$$u_R = \begin{cases} 1 & \text{if } R \leq R_L \\ 0 & \text{if } R \geq R_U \\ \left(\frac{R_U - R}{R_U - R_L}\right)^2 & \text{else} \end{cases} \tag{25}$$

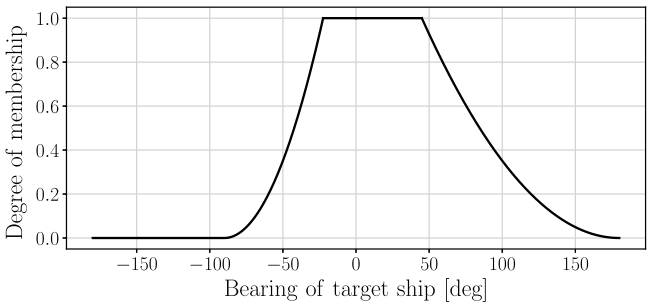


Fig. 11. Membership function for the bearing angle, θ_T, of the target ship, with bounds $\theta_{PU} = 180°$, $\theta_{PL} = 45°$, $\theta_{NU} = 90°$, and $\theta_{NL} = 22.5°$.

To encourage the appropriate behavior in head-on situations, the function for the bearing angle of the TS relative to the OS should be largest on the starboard side. Defining θ_{PU}, θ_{PL}, θ_{NU}, and θ_{NL} as the positive upper, positive lower, negative upper, and negative lower bounds on θ_T, the membership function for the bearing angle can be defined as below and illustrated in Fig. 11.

$$u_{\theta_T} = \begin{cases} \text{clip}\left(\left(\frac{\theta_{PU} - \theta_T}{\theta_{PU} - \theta_{PL}}\right)^2, 0, 1\right) & \text{if } \theta_T \geq 0 \\ \text{clip}\left(\left(\frac{\theta_{NU} - |\theta_T|}{\theta_{NU} - \theta_{NL}}\right)^2, 0, 1\right) & \text{if } \theta_T < 0 \end{cases} \tag{26}$$

After implementing a CRI containing the four membership functions introduced so far, it became clear that it was necessary to add an element to the CRI to deter the OS from crossing ahead of a TS. Since the TS's speed towards the OS can quantify whether the OS is ahead of the TS and is readily available in the observation vector (v_y and v_x), an additional membership function is designed. Hence, we define $u_V(\cdot)$ as the ratio of the TS's speed towards the OS to its absolute speed, as described in Eq. (27). Such a ratio was chosen to avoid issues with differences in speed among the TSs, which quickly could have arisen if the numerical value of v_y had been used instead. On the other hand, it might be desirable to distinguish between crossing ahead ships traveling at different speeds, as faster ships naturally pose a higher risk. However, this is considered to be outside the scope of this work. It is worth noting that $u_V(\cdot)$ is negative when v_y is negative, emphasizing the advantage of astern crossings.

$$u_V = \frac{v_y}{\sqrt{v_x^2 + v_y^2}} \tag{27}$$

Integrating the introduced membership functions into a collision risk index, we have that

$$CRI = \max\left(0, \alpha_{CPA}\sqrt{u_{DCPA} \cdot u_{TCPA}} + \alpha_{\theta_T} u_{\theta_T} + \alpha_R u_R + \alpha_V u_V\right) \tag{28}$$

where the CPA composite term was designed in such that a combination of low values for both DCPA and TCPA gives rise to a high CRI. It also accurately expresses how a low value of either DCPA or TCPA significantly reduces the overall risk. The `max`-function is applied to ensure that the CRI is always larger or equal to zero.

Finally, values are assigned to the weights such that the sum is equal to unity, giving

$$\alpha_{CPA} + \alpha_{\theta_T} + \alpha_R + \alpha_V = 1 \tag{29}$$

In this work, the parameter values specified in Table 3 are used. Initial choices were made based on values suggested in the literature (Chen et al., 2014; Yan, 2002), emphasizing DCPA and TCPA. However, it was discovered that more weight had to be placed on the target bearing angle, absolute distance, and

Table 2
Hyperparameters for the PPO algorithm.

Parameter	Description	Value
γ	Discount factor	0.999
T	Timesteps per training iteration	1024
N_A	Number of parallel actors	8
K	Training timesteps	$6 * 10^6$
η	Learning rate	$2 * 10^{-4}$
N_{MB}	Number of minibatches	32
λ	GAE bias vs. variance parameter	0.95
c_2	Value function coefficient	0.01
c_2	Entropy coefficient	0.01
ϵ	Clipping parameter	0.2

approaching velocity to achieve the desired behavior. The configuration of the path following and static obstacle rewards listed in Meyer, Heiberg et al. (2020) have been applied in this work.

4.4. Simulation configuration

Having established the simulation environment and reward functions suitable for Reinforcement Learning, the final step is configuring the RL algorithm. We choose PPO in this work considering its superior performance compared to other RL algorithms in similar environments (Larsen et al., 2021). PPO is summarized in Algorithm 1, and Table 2 specifies the hyperparameters chosen for the algorithm. The neural networks parameterizing the value and policy functions are chosen as fully connected networks with 2 hidden layers consisting of 64 nodes each.

The RL agent will train in the synthetic training environment (Fig. 13) until it is consistently able to solve the environment over several episodes, i.e., the training ends when the agent consistently succeeds in following the path without colliding. An episode starts with the vessel randomly initialized at the starting point, and ends when (1) the vessel reaches the path's end, (2) the agent spends more than 3500 time steps, (3) the agent's cumulative reward becomes smaller than -10000, or (4) the vessel has collided. When any of these conditions are true, the environments resets.

Algorithm 1 The core elements of the Proximal Policy Optimization algorithm.

1: $\theta_0, \phi_0 \leftarrow$ Randomly initialized policy and value networks.
2: **for** $k \leftarrow 1$ **to** K **do**
3: Collect trajectories $\mathcal{D}_k = \tau_i$ by running policy π_{θ_k} in the environment for T timesteps.
4: Compute rewards-to-go $\hat{R}_t$.
5: Compute advantage estimates $\hat{A}_t$, using value function V_{ϕ_k}
6: Update the policy parameters according to the PPO objective:

$$\theta_{k+1} = \arg\max_{\theta} \frac{1}{|\mathcal{D}_k|T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \min \left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)} A^{\pi_{\theta_k}}(s_t, a_t), \right.$$
$$\left. g\left(\epsilon, A^{\pi_{\theta_k}}(s_t, a_t)\right) \right)$$

by stochastic gradient ascent using the Adam optimizer.
7: $\theta_k \longleftarrow \theta_{k+1}$
8: Fit value function by regression:

$$\phi_{k+1} = \arg\min_{\phi} \frac{1}{|\mathcal{D}_k|T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \left(V_{\phi_k}(s_t) - \hat{R}_t \right)^2,$$

by gradient descent.

4.5. Performance evaluation

A three-step evaluation process is employed to assess the performance of the RL agent. First, the agent's behavior and performance in the training environment are assessed, and snippets from situations relevant to rules 14–16 of the COLREGs are presented. Next, two-vessel testing scenarios are constructed to test for COLREG-compliance specifically. Lastly, the agents are evaluated in AIS-based environments. These modes of assessment are described individually in the following subsections.

4.5.1. Performance in the training environment

A natural starting point for performance evaluation is assessing the agent's behavior in its training environment. The overall performance can be evaluated by collecting statistics on the collision rate, level of path completion, and reward. These statistics serve as a guide for when to stop the training and a point of comparison between approaches. Moreover, a qualitative assessment is made by observing the agent's behavior through video recordings. Snippets are chosen from the videos to highlight the behavior in situations where the COLREGs apply. This is not always the case since the training environment often presents the agent with difficult situations containing various static and dynamic obstacles, which cannot be accurately subjected to the COLREGs.

4.5.2. Testing of COLREG-compliance

The next step in the testing process is subjecting the agent to scenarios specifically designed to capture COLREG-compliance. This is especially useful since it is challenging to find scenarios that perfectly showcase COLREG-compliance in the training environment. However, the agent's success can easily be quantified through simpler two-vessel scenarios. One scenario to be tested is self-evident, namely the head-on scenario. In addition, two different crossing situations, one from the starboard and one from the port side, were chosen. For each scenario, the TS's initial angles and path angles are varied slightly within a range of $\pm 5^\circ$ of the default angles, which allows for an accumulation of statistics on success rate in the respective scenarios. It should be noted that the target ships have been modeled exclusively large in the testing scenarios to reflect the size of the large ships encountered in the AIS-based scenarios and for visual clarity.

4.5.3. AIS-based testing

Lastly, three environments based on real-world high-fidelity terrain data are used to assess the generalization performance of the agent. These environments were developed by Meyer (2020) using AIS tracking data and terrain data from the Trondheim Fjord area and are distinctly different. A dashed black line represents the desired OS trajectory in the following illustrations. Each TS is drawn at its initial position, and trajectories are drawn as dotted red lines. Note that these are examples of spawned environments and that a set amount of target ships are chosen from the AIS database each time an instance of the specific scenario is created. Additionally, the apparent density of TS trajectories does not directly reflect the number of encounters, as this depends on the speed of each vessel.

The first AIS-based scenario is the **Trondheim** scenario (Fig. 16(b)), in which the agent is required to cross a fjord of width $\sim$12 km while following a straight path. Doing so, it mainly meets crossing traffic consisting of larger vessels. In the challenging **Ørland-Agdenes** scenario (Fig. 16(a)), the agent encounters two-way traffic in a narrow fjord entrance region. It must blend into the heavy traffic to complete the path while avoiding head-on collisions. In addition, the ability to overtake other vessels is assessed. As in the Trondheim scenario, the vessels are primarily

Table 3
Reward configuration for the risk-based approach.

Parameter	Interpretation	Value
β_{CRI}	Scaling factor for overall risk level	10
β_{RL}	Scaling factor for the lower bound on distance	8
β_{RU}	Scaling factor for the upper bound on distance	18
θ_{PU}	Positive upper limit for the bearing angle θ_T	180°
θ_{PL}	Positive lower limit for the bearing angle θ_T	45°
θ_{NU}	Negative upper limit for the bearing angle θ_T	90°
θ_{NL}	Negative lower limit for the bearing angle θ_T	22.5°
α_{CPA}	Weighting of CPA membership function	0.3
α_{θ_T}	Weighting of target bearing angle membership function	0.2
α_R	Weighting of absolute distance membership function	0.3
α_V	Weighting of approaching velocity membership function	0.2
d_L	Minimal safe encounter distance	320 m
d_U	Absolute safe encounter distance	1500 m

Table 4
Results from repetitive testing of COLREG-compliance with slightly varying scenarios, 100 episodes.

Scenario	Success rate
Head-on	100%
Crossing from starboard	100%
Crossing from port	100%

bigger than the OS. Lastly, the **Froan** scenario (Fig. 16(c)) offers a demanding terrain with hundreds of small islands. As a result, it tests the ability of the agent to generalize to a challenging environment with a high density of static obstacles in varying shapes and sizes. The area is less trafficked, and the vessels encountered are physically similar to the OS.

5. Results and discussion

In this section, the results from the risk-based implementation of the COLREGs are presented and evaluated. First, the RL agent is evaluated in the synthetic training environment, considering its general path following and collision avoidance performance. Second, the presence and consistency of COLREG-compliant behavior are assessed in isolated, high-risk encounters. Finally, the agent is presented the simulated real-world AIS-based scenarios to see how the learned policy generalizes to complex and unseen situations.

5.1. Training and testing in the synthetic environment

After training the RL agent in the synthetic environment (Fig. 4) for approximately 4000 episodes, its collision rate dropped to near zero, and the progress rate rose to 100%. Fig. 12 shows PPO's learning curves during the training phase. Snippets from the training environment have been included in Fig. 13, showcasing training scenarios in which the agent behaves in a COLREG-compliant manner. The COLREGs clearly define these situations: passing on the right in head-on situations, slowing down and passing astern instead of ahead, and allowing space between it and the TS during overtaking. Although the training statistics indicate the agent's ability to navigate and avoid collisions, they do not reveal whether the COLREG-compliance is consistent, which must be evaluated separately.

5.1.1. Testing of COLREG-compliance

The next step in the evaluation process is COLREG-compliance testing with repetitive testing in different encounter scenarios. Fig. 14 shows how the agent avoids collision in a COLREG-compliant manner. In addition, the agent follows the path well once the encounter has passed. Repetitive testing reveals that these results are stable, as the correct behavior was seen in

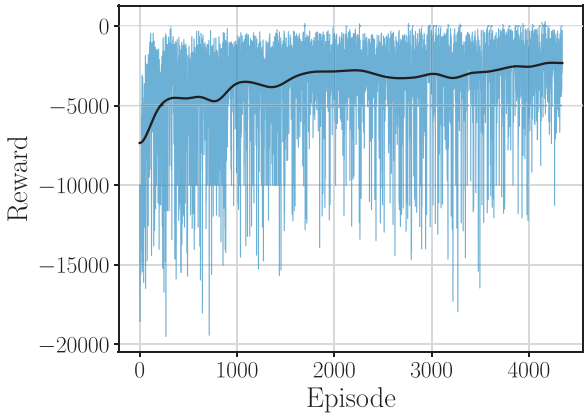


Fig. 12. Reward curves of the PPO agent during the training phase. The blue curve is the raw reward, whereas the black line represents the smoothed moving average.

100% of the episodes for each testing scenario, as summarized in Table 4. These results indicate that the agent can intelligently interpolate between path following and COLREG-compliant collision avoidance in isolated high-risk encounters. However, there is no guarantee that this behavior translates into more complex scenarios.

5.2. Testing in AIS-based environments

Finally, the risk-based agent is assessed in AIS-based real-world environments to find how well the agent generalizes to previously unseen scenarios. As these environments are modeled using real-world terrain mapping and AIS traffic data, the agent will likely encounter complex scenarios where the COLREGs do not clearly define the correct behavior. Therefore, we do not expect the agent always to find a COLREG-compliant solution. The agent's excellent static obstacle avoidance and COLREG-compliant behavior are highlighted in Fig. 15. Note that the static obstacles in Fig. 15(d) are significantly smaller than those encountered in the training scenario.

Lastly, trajectories from each environment are presented in Fig. 16. These trajectories illustrate the agent's ability to dynamically follow a predetermined path in the face of static and moving obstacles. Whether the agent is faced with heavy two-way parallel traffic (Fig. 16(a)), crossing traffic (Fig. 16(b)), or an untraversable path (Fig. 16(c)), it adapts to the situation and finds a suitable solution. Thus, the agent generalizes its decision-making policy from the synthetic and stochastic training environment to previously unseen environments.

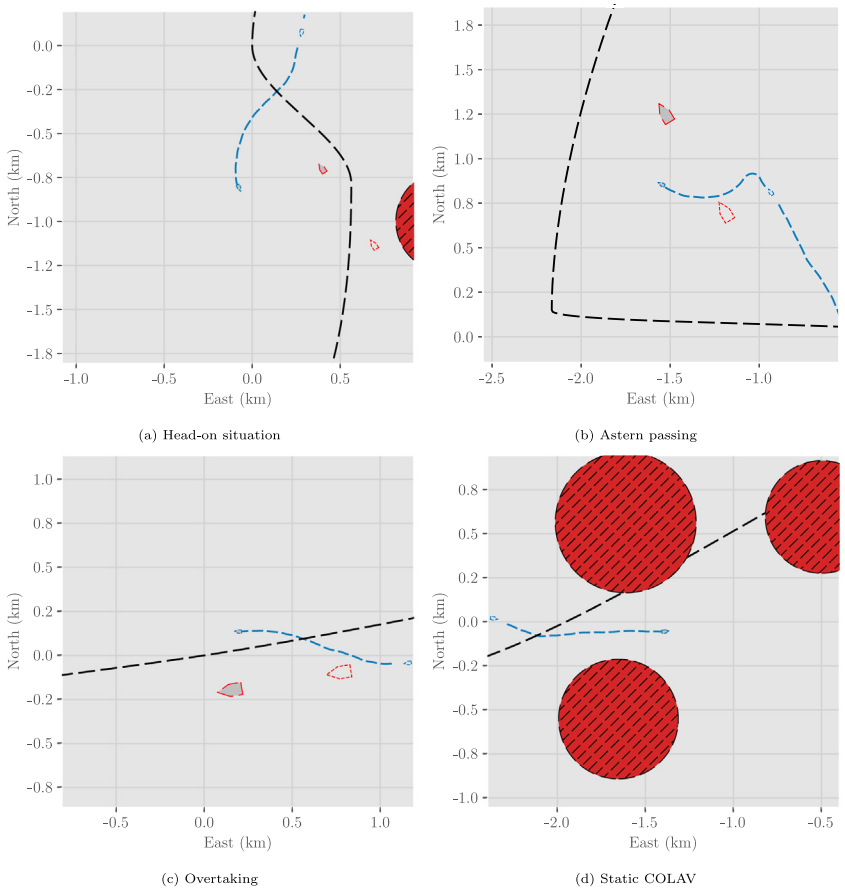


Fig. 13. Risk-based agent performing common naval collision avoidance maneuvers in the training environment. The agent's trajectory is drawn as a blue dashed line, and the target ships with trajectories are drawn in red. The dotted vessel outlines show their positions 100 time steps prior.

6. Conclusion

This work has presented a novel approach implementing state-of-the-art collision risk indices into the reward design of a model-free DRL algorithm. By training an autonomous agent in a synthetic stochastic environment, the agent learned a decision-making policy capable of simultaneous path following and collision avoidance. Moreover, this agent exhibits robust COLREG-compliant behavior when investigating high-risk two-vessel encounters. However, this compliance is restricted to a subset of the COLREGs (rules 14–16) directly relevant for COLAV.

The agent's generalization performance was tested by having it follow predetermined paths in three simulated real-world environments, in which the agent succeeded in adapting to their varying traffic dynamics. The agent's COLREG-compliance in these scenarios was, however, not assessed.

Thus, we have shown that incorporating collision risk indices into PPO facilitates a cheap and effective approach for guiding the DRL agent towards COLREG compliance. However, many COLREG rules were not considered in this work, as they are ambiguous and thus need adaptation for machine interpretability. Having unambiguous definitions of the required behavior in different scenarios is necessary before any autonomous vessel can be claimed to be fully COLREG-compliant. Moreover, deep representations are still immature in terms of safety guarantees, predictability, and reproducibility; further advancements in this field is also required before neural network based controllers can be safely applied in the real-world. Regardless, once the COLREGs are modernized for digital applications, DRL can likely produce fully COLREG-compliant and autonomous COLAV systems.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

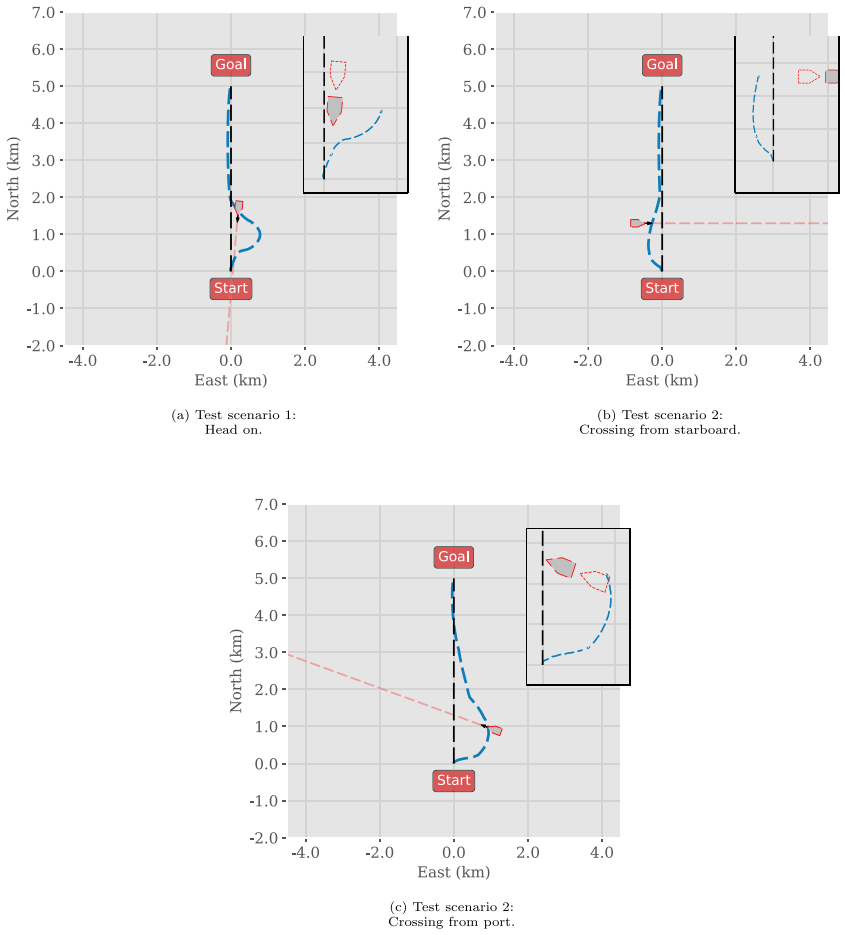
Acknowledgments

The authors acknowledge the financial support from the Norwegian Research Council and the industrial partners: DNV GL, Kongsberg and Maritime Robotics of the Autosit project. (Grant No. : 295033).

Appendix. Colreg rules

Rule 6: Safe speed

Every vessel shall at all times proceed at a safe speed so that she can take proper and effective action to avoid collision and be stopped within a distance appropriate to the prevailing circumstances and conditions.



(a) Test scenario 1: Head on.

(b) Test scenario 2: Crossing from starboard.

(c) Test scenario 2: Crossing from port.

Fig. 14. Agent behavior in COLREG-compliance test scenarios. The agent's trajectory is drawn as a blue dashed line, and the target ships with trajectories are drawn in red. The dotted vessel outlines show their positions 100 time steps prior to the present time. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Rule 8: Action to avoid collision

(b) Any alteration of course and/or speed to avoid collision shall, if the circumstances of the case admit, be large enough to be readily apparent to another vessel observing visually or by radar; a succession of small alterations of course and/or speed should be avoided.

(c) If there is sufficient sea-room, alteration of course alone may be the most effective action to avoid a close-quarters situation provided that it is made in good time, is substantial and does not result in another close-quarters situation.

(d) Action taken to avoid collision with another vessel shall be such as to result in passing at a safe distance. The effectiveness of the action shall be carefully checked until the other vessel is finally past and clear.

(e) If necessary to avoid collision or allow more time to assess the situation, a vessel shall slacken her speed or take all way off by stopping or reversing her means of propulsion.

Rule 14: Head-on situation

(a) When two power-driven vessels are meeting on reciprocal or nearly reciprocal courses so as to involve risk of collision each shall alter her course to starboard so that each shall pass on the port side of the other.

(b) Such a situation shall be deemed to exist when a vessel sees the other ahead or nearly ahead and by night she could see the masthead lights of the other in a line or nearly in a line and/or both sidelights and by day she observes the corresponding aspect of the other vessel.

(c) When a vessel is in any doubt as to whether such a situation exists she shall assume that it does exist and act accordingly.

Rule 15: Crossing situation

When two power-driven vessels are crossing so as to involve risk of collision, the vessel which has the other on her own starboard side shall keep out of the way and shall, if the circumstances of the case admit, avoid crossing ahead of the other vessel.

Rule 16: Action by give-way vessel

Every vessel which is directed to keep out of the way of another vessel shall, so far as possible, take early and substantial action to keep well clear.

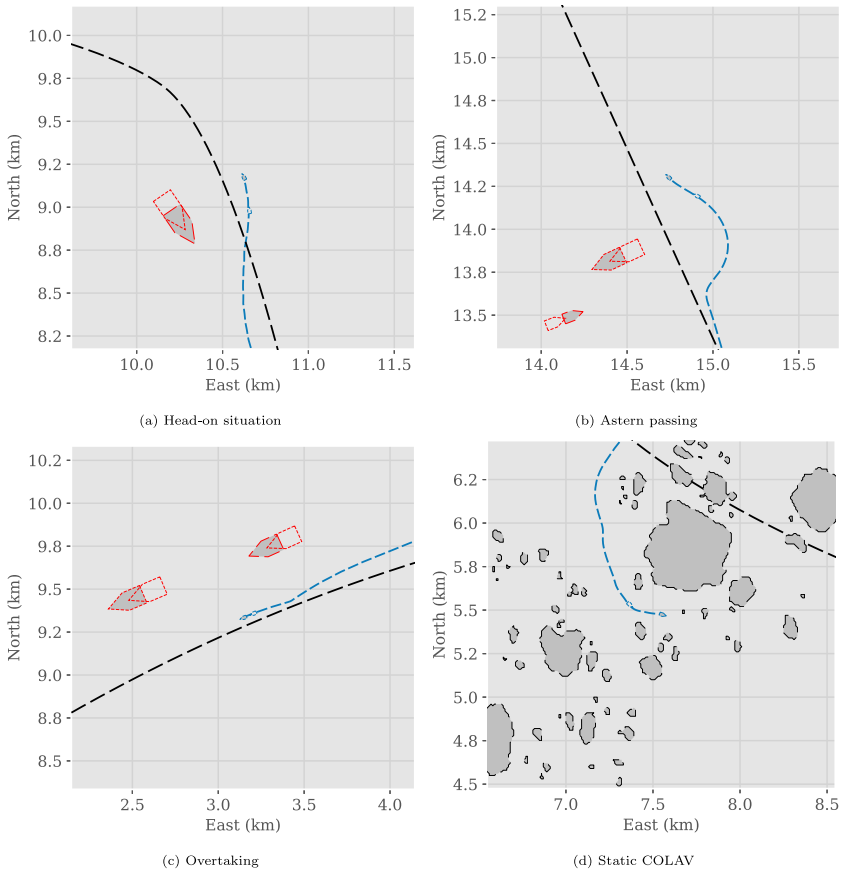
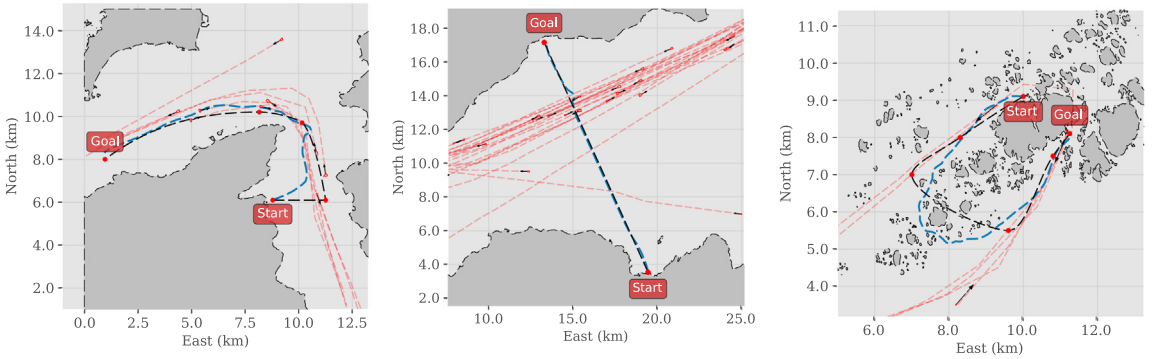


Fig. 15. Risk-based agent performing common naval collision avoidance maneuvers in the AIS-based environment. The agent trajectory is drawn as a blue dashed line, and the target ships are drawn in red. The dotted vessel outlines show their positions 100 time steps prior to the present time. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) Risk-based agent's trajectory in the Ørland-Agdenes test scenario. The agent intelligently merges with the two-way traffic, and follows it towards the goal.

(b) Risk-based agent's trajectory in the Trondheim test scenario. The agent exhibits excellent path following, and is seen to maneuver the crossing traffic before returning to the path and reaching the goal.

(c) Risk-based agent's trajectory in the Froan test scenario. When presented with an impossible path, the agent intelligently finds another solution for navigating the dense archipelago and merge into the parallel traffic.

Fig. 16. Trajectories from three different AIS-based environments drawn as blue dashed lines. Target ships and trajectories are drawn in red. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

References

Aniculaesei, A., Arnsberger, D., Howar, F., & Rausch, A. (2016). Towards the verification of safety-critical autonomous systems in dynamic environments. *Electronic Proceedings in Theoretical Computer Science, EPTCS, 232*, 79–90. http://dx.doi.org/10.4204/EPTCS.232.10.

Autoship (2020). AUTOSHIP – AUtonomous shipping initiative for European waters. https://www.autoship-project.eu/. Accessed: 2020-06-10.

Balamurugan, N., Mohan, S., Adimoolam, M., Ayeelyan, J., Gadekallu, T., & Wang, W. (2021). DOA tracking for seamless connectivity in beamformed IoT-based drones. *Computer Standards & Interfaces, 79*, Article 103564. http://dx.doi.org/10.1016/j.csi.2021.103564.

Benjamin, M. R., Leonard, J. J., Curcio, J. A., & Newman, P. M. (2006). A method for protocol-based collision avoidance between autonomous marine surface crafts. *Journal of Field Robotics, 29*(4), 554–575. http://dx.doi.org/10.1002/rob.

Bhargava, C., Sharma, P. K., Senthilkumar, M., Padmanaban, S., Ramachandaramurthy, V. K., Leonowicz, Z., Blaabjerg, F., & Mitolo, M. (2020). Review of

health prognostics and condition monitoring of electronic components. *IEEE Access*, *8*, 75163–75183. http://dx.doi.org/10.1109/ACCESS.2020.2989410.

Casalino, G., Turetta, A., & Simetti, E. (2009). A three-layered architecture for real time path planning and obstacle avoidance for surveillance USVs operating in harbour fields. In *OCEANS '09 IEEE Bremen: Balancing technology with future needs* (pp. 1–8). IEEE, http://dx.doi.org/10.1109/OCEANSE.2009.5278104.

Chen, S., Ahmad, R., Lee, B. G., & Kim, D. H. (2014). Composition ship collision risk based on fuzzy theory. *Journal of Central South University*, *21*(11), 4296–4302. http://dx.doi.org/10.1007/s11771-014-2428-z.

Cockcroft, A. N. (1981). The estimation of collision risk for marine traffic. *Journal of Navigation*, *34*(1), 145–147. http://dx.doi.org/10.1017/S0373463300024310.

Codevilla, F., Santana, E., López, A. M., & Gaidon, A. (2019). Exploring the limitations of behavior cloning for autonomous driving. arXiv:1904.08980.

Davis, P. V., Dove, M. J., & Stockel, C. T. (1980). A computer simulation of marine traffic using domains and arenas. *Journal of Navigation*, *33*(2), 215–222. http://dx.doi.org/10.1017/S0373463300035220.

Ding, J., Gillula, J. H., Huang, H., Vitus, M. P., Zhang, W., & Tomlin, C. J. (2011). Hybrid systems in robotics: Toward reachability-based controller design. *IEEE Robotics & Automation Magazine*.

Dingus, T. A., Guo, F., Lee, S., Antin, J. F., Perez, M., Buchanan-King, M., & Hankey, J. (2016). Driver crash risk factors and prevalence evaluation using naturalistic driving data. *Proceedings of the National Academy of Sciences of the United States of America*, *113*(10), 2636–2641. http://dx.doi.org/10.1073/pnas.1513271113.

DNV GL (2019). Digital twins and sensor monitoring. https://www.dnvgl.com/expert-story/maritime-impact/Digital-twins-and-sensor-monitoring.html. Accessed: 2020-02-25.

DNV GL (2020). The ReVolt – A new inspirational ship concept. https://www.dnvgl.com/technology-innovation/revolt/index.html. Accessed: 2020-27-05.

Eriksen, B.-O. H. (2019). *Collision Avoidance and Motion Control for Autonomous Surface Vehicles* (Ph.D. thesis), Norwegian University of Science and Technology.

European Maritime Safety Agency (EMSA) (2018). Annual overview of marine casualties and incidents 2018. http://www.emsa.europa.eu/news-a-press-centre/external-news/item/3406-annual-overview-of-marine-casualties-and-incidents-2018.html. Accessed: 2019-11-05.

Fiorini, P., & Shiller, Z. (1998). Motion planning in dynamic environments using velocity obstacles. *International Journal of Robotics Research*, *17*(7), 760–772. http://dx.doi.org/10.1177/027836499801700706.

Fossen, T. I. (2011). *Handbook of marine craft hydrodynamics and motion control*. http://dx.doi.org/10.1002/9781119994138.

Fox, D., Wolfram, B., & Sebastian, T. (1997). The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, *4*, 137–146.

Gang, L., Wang, Y., Sun, Y., Zhou, L., & Zhang, M. (2016). Estimation of vessel collision risk index based on support vector machine. *Advances in Mechanical Engineering*, *8*(11), 1–10. http://dx.doi.org/10.1177/1687814016671250.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. The MIT Press.

Goodwin, E. M. (1978). Marine encounter rates. *Journal of Navigation*, *31*(3), 357–369. http://dx.doi.org/10.1017/S0373463300041904.

Granstrom, K., Baum, M., & Reuter, S. (2016). Extended object tracking: Introduction, overview and applications. arXiv:1604.00970.

Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2017). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. arXiv:1801.01290. arXiv [Preprint]. Available at: https://arxiv.org/abs/1801.01290. (Accessed July 07, 2021).

Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *Journal of the Society for Industrial and Applied Mathematics*, *9*(4), 514–532. http://dx.doi.org/10.1137/0109044.

International Maritime Organization (1972a). COLREGS - International regulations for preventing collisions at sea. URL: http://www.imo.org/en/About/Conventions/ListOfConventions/Pages/COLREG.aspx.

International Maritime Organization (IMO) (2020). AIS Transponders. http://www.imo.org/en/OurWork/Safety/Navigation/Pages/AIS.aspx. Accessed: 2019-09-16.

Kamal, M., Tariq, M., Srivastava, G., & Malina, L. (2021). Optimized security algorithms for intelligent and autonomous vehicular transportation systems. *IEEE Transactions on Intelligent Transportation Systems*, *PP*, 1–7. http://dx.doi.org/10.1109/TITS.2021.3123188.

Kim, D.-G., Hirayama, K., & Okimoto, T. (2015). Ship collision avoidance by distributed tabu search. *TransNav, the International Journal on Marine Navigation and Safety of Sea Transportation*, *9*(1), 23–29. http://dx.doi.org/10.12716/1001.09.01.03.

KONGSBERG (2020). Autonomous ship project, key facts about yara birkeland. https://www.kongsberg.com/no/maritime/support/themes/autonomous-ship-project-key-facts-about-yara-birkeland/. Accessed: 2020-27-05.

Kuwata, Y., Wolf, M. T., Zarzhitsky, D., & Huntsberger, T. L. (2014). Safe maritime autonomous navigation with COLREGS, using velocity obstacles. *IEEE Journal of Oceanic Engineering*, *39*(1), 110–119. http://dx.doi.org/10.1109/JOE.2013.2254214.

Larsen, T. N., Teigen, H. Ø., Laache, T., Varagnolo, D., & Rasheed, A. (2021). Comparing deep reinforcement learning algorithms' ability to safely navigate challenging waters. *Frontiers in Robotics and AI*, *8*, 287. http://dx.doi.org/10.3389/frobt.2021.738113.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. (2015). Continuous control with deep reinforcement learning. arXiv:1509.02971.

Lin, B., & Huang, C. H. (2006). Comparison between arpa radar and AIS characteristics for vessel traffic services. *Journal of Marine Science and Technology*, *14*(3), 182–189.

Loe, Ø. A. G. (2008). *Collision Avoidance for Unmanned Surface Vehicles* (Ph.D. thesis), Norwegian University of Science and Technology, URL: http://ntnu.diva-portal.org/smash/record.jsf?pid=diva2:347606.

Martinsen, A. B. (2018). End-to-end training for path following and control of marine vehicles.

Martinsen, A. B., & Lekkas, A. M. (2019). Curved path following with deep reinforcement learning: Results from three vessel models. In *OCEANS 2018 MTS/IEEE Charleston, OCEAN 2018*. http://dx.doi.org/10.1109/OCEANS.2018.8604829.

Mekala, M. S., Park, W., Dhiman, G., Srivastava, G., Park, J. H., & Jung, H.-Y. (2021). Deep learning inspired object consolidation approaches using lidar data for autonomous driving: A review. *Archives of Computational Methods in Engineering*, http://dx.doi.org/10.1007/s11831-021-09670-y.

Meyer, E. (2020). *On course towards model-free guidance: A self-learning approach to dynamic collision avoidance for autonomous surface vehicles* (Master's thesis), Norwegian University of Science and Technology.

Meyer, E., Heiberg, A., Rasheed, A., & San, O. (2020). COLREG-Compliant collision avoidance for unmanned surface vehicle using deep reinforcement learning. arXiv e-prints, arXiv:2006.09540. arXiv:2006.09540.

Meyer, E., Robinson, H., Rasheed, A., & San, O. (2020). Taming an autonomous surface vehicle for path following and collision avoidance using deep reinforcement learning. *IEEE Access*, *8*, 41466–41481.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., Silver, D., & Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. arXiv:1602.01783.

International Maritime Organization (1972b). COLREGS - INternational regulations for preventing collisions at sea. https://www.imo.org/en/About/Conventions/Pages/COLREG.aspx. Accessed: 2021-11-15.

Park, G.-K., Benedictos, J. L. R. M., Shin, S. C., & Nam-Kyun, I. (2006). Design of a fuzzy-CBR support system for ship ' s collision avoidance. In *SCIS&ISIS2006* (pp. 240–248).

Rawson, A., Rogers, E., Foster, D., & Phillips, D. (2014). Practical application of domain analysis: Port of london case study. *Journal of Navigation*, *67*(2), 193–209. http://dx.doi.org/10.1017/S0373463313000684.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv:1707.06347.

Serigstad, E., Eriksen, B. O. H., & Breivik, M. (2018). Hybrid collision avoidance for autonomous surface vehicles. *IFAC-PapersOnLine*, *51*(29), 1–7. http://dx.doi.org/10.1016/j.ifacol.2018.09.460.

Siciliano, B., & Khatib, O. (2008). *Springer handbook of robotics, Vol. 1*. Springer.

Silver, D., Singh, S., Precup, D., & Sutton, R. S. (2021). Reward is enough. *Artificial Intelligence*, *299*, Article 103535. http://dx.doi.org/10.1016/j.artint.2021.103535.

Skjetne, R., Smogeli, Ø., & Fossen, T. I. (2004a). Modeling, identification, and adaptive maneuvering of CyberShip II: A complete design with experiments. *IFAC Proceedings Volumes*, *37*(10), 203–208, IFAC Conference on Computer Applications in Marine Systems - CAMS 2004, Ancona, Italy, 7-9 July 2004.

Skjetne, R., Smogeli, Ø. N., & Fossen, T. I. (2004b). A nonlinear ship manoeuvering model: Identification and adaptive control with experiments for a model ship.

Skredderberget, A. (2018). The first ever zero emission, autonomous ship. https://www.yara.com/knowledge-grows/game-changer-for-the-environment/. Accessed: 2019-11-12.

Smierzchalski, R. (1999). Evolutionary trajectory planning of ships in navigation traffic areas. *Journal of Marine Science and Technology*, *4*(1), 1–6. http://dx.doi.org/10.1007/s007730050001.

SNAME, The Society of Naval Architecture and Marine Engineers (1950). Nomenclature for treating the motion of a submerged body through a fluid".

Soloperto, R., Kohler, J., Allguwer, F., & Muller, M. A. (2019). Collision avoidance for uncertain nonlinear systems with moving obstacles using robust model predictive control. In *18th European control conference (ECC)* (pp. 811–817). http://dx.doi.org/10.23919/ecc.2019.8796049.

Sørensen, M. E. N., Breivik, M., & Eriksen, B. H. (2017). A ship heading and speed control concept inherently satisfying actuator constraints. In *2017 IEEE conference on control technology and applications (CCTA)* (pp. 323–330).

Statheros, T., Howells, G., & McDonald-Maier, K. (2008). Autonomous ship collision avoidance navigation concepts, technologies and techniques. *Journal of Navigation*, *61*(1), 129–142. http://dx.doi.org/10.1017/S037346330700447X.

Suresh, P., Saravanakumar, U., Iwendi, C., Mohan, S., & Srivastava, G. (2021). Field-programmable gate arrays in a low power vision system. *Computers and Electrical Engineering*, *90*, Article 106996. http://dx.doi.org/10.1016/j.compeleceng.2021.106996, URL: https://www.sciencedirect.com/science/article/pii/S0045790621000240.

Svec, P., Shah, B. C., Bertaska, I. R., Alvarez, J., Sinisterra, A. J., Von Ellenrieder, K., Dhanak, M., & Gupta, S. K. (2013). Dynamics-aware target following for an autonomous surface vehicle operating under COLREGs in civilian traffic. In *IEEE international conference on intelligent robots and systems* (pp. 3871–3878). IEEE, http://dx.doi.org/10.1109/IROS.2013.6696910.

Szlapczynski, R., & Szlapczynska, J. (2017). Review of ship safety domains: Models and applications. *Ocean Engineering*, *145*, 277–289. http://dx.doi.org/10.1016/j.oceaneng.2017.09.020, URL: https://doi.org/10.1016/j.oceaneng.2017.09.020.

Tai, L., Zhang, J., Liu, M., Boedecker, J., & Burgard, W. (2016). A survey of deep network solutions for learning control in robotics: From reinforcement to imitation. arXiv:1612.07139.

Tam, C. K., Bucknall, R., & Greig, A. (2009). Review of collision avoidance and path planning methods for ships in close range encounters. *Journal of Navigation*, *62*(3), 455–476. http://dx.doi.org/10.1017/S0373463308005134.

Thomas, P., Morris, A., Talbot, R., & Fagerlind, H. (2013). Identifying the causes of road crashes in Europe. *Annals of Advances in Automotive Medicine*, *57*(November 2014), 13–22.

Vallestad, I. J. (2019). *Path following and collision avoidance for marine vessels with deep reinforcement learning* (Master's thesis), Norwegian University of Science and Technology.

Wang, Y., & Chin, H. C. (2016). An empirically-calibrated ship domain as a safety criterion for navigation in confined waters. *Journal of Navigation*, *69*(2), 257–276. http://dx.doi.org/10.1017/S0373463315000533.

Woernner, K. (2014). COLREGS-compliant autonomous collision avoidance using multi-objective optimization with interval programming. *Naval Engineers Journal*, *126*(2), 64–65. http://dx.doi.org/10.21236/ada609415.

Xu, Q., & Wang, N. (2014). A survey on ship collision risk evaluation. *Promet - Traffic&Transportation*, *26*(6), 475–486. http://dx.doi.org/10.7307/ptt.v26i6.1386.

Xu, Q., Zhang, C., & Zhang, L. (2017). Deep convolutional neural network based unmanned surface vehicle maneuvering. In *Proceedings, 2017 Chinese automation congress (CAC)* (2), (pp. 878–881). http://dx.doi.org/10.1109/CAC.2017.8242889.

Yan, Q. (2002). A model for estimating the risk degrees of collisions. *Journal of Wuhan University of Technology (Transportation Science & Engineering)*, *26*(2).

Zhao, L., & Roh, M. I. (2019). COLREGS-compliant multiship collision avoidance based on deep reinforcement learning. *Ocean Engineering*, http://dx.doi.org/10.1016/j.oceaneng.2019.106436.