Provable Traffic Rule Compliance in Safe Reinforcement Learning on the Open Sea

Hanna Krasowski, *Graduate Student Member, IEEE*, and Matthias Althoff, *Member, IEEE*

***Abstract*—For safe operation, autonomous vehicles have to obey traffic rules that are set forth in legal documents formulated in natural language. Temporal logic is a suitable concept to formalize such traffic rules. Still, temporal logic rules often result in constraints that are hard to solve using optimization-based motion planners. Reinforcement learning (RL) is a promising method to find motion plans for autonomous vehicles. However, vanilla RL algorithms are based on random exploration and do not automatically comply with traffic rules. Our approach accomplishes guaranteed rule-compliance by integrating temporal logic specifications into RL. Specifically, we consider the application of vessels on the open sea, which must adhere to the Convention on the International Regulations for Preventing Collisions at Sea (COLREGS). To efficiently synthesize rule-compliant actions, we combine predicates based on set-based prediction with a statechart representing our formalized rules and their priorities. Action masking then restricts the RL agent to this set of verified rule-compliant actions. In numerical evaluations on critical maritime traffic situations, our agent always complies with the formalized legal rules and never collides while achieving a high goal-reaching rate during training and deployment. In contrast, vanilla and traffic rule-informed RL agents frequently violate traffic rules and collide even after training.**

***Index Terms*—Safe reinforcement learning, autonomous vessels, temporal logic, provable guarantees, collision avoidance.**

I. Introduction

Reinforcement learning (RL) has provided promising results for a variety of motion planning tasks, e.g., autonomous driving [1], [2], robotic manipulation [3], [4], and autonomous vessel navigation [5], [6], [7]. RL algorithms learn a capable policy through random exploration. As random exploration is inherently unsafe, RL agents are mainly trained and tested in simulation only. To transfer the capabilities of RL-based motion planning systems to the physical world, the agents have to be safe. Safe RL extends RL algorithms with safety considerations. Most safe RL approaches constrain the learning softly, e.g., by integrating risk measures in the reward function or by adapting the optimization problem for obtaining a policy considering constraints [8]. However, for safety-critical tasks, such as motion planning in the physical world, hard safety guarantees are necessary, which most safe RL approaches cannot provide.

Provably safe RL achieves hard safety guarantees during training and operation by combining RL with formal methods [8]. The safety specifications regarded in provably safe RL are so far mainly *avoid specifications*, i.e., it is ensured that unsafe areas and actions are always avoided. However, the notion of safety for real-world tasks is often more complex than avoiding unsafe sets. For autonomous vehicles, legal safety is usually required, meaning that vehicles do not cause collisions by obeying traffic rules [9], [10]. To apply formal methods, these traffic rules need to be formalized. Temporal logic is suited to formalize traffic rules [9], [11], [12], [13], [14], [15], as it can capture their spatial and temporal dependencies well. Still, efficient and generalizable integration of formalized traffic rules in motion planning approaches is an open research problem.

In this work, we propose a provably safe RL approach that ensures legal safety by complying with traffic rules formalized in temporal logic for the application of autonomous vessel navigation. Fig. 1 displays the concept of our approach. We develop a statechart that reflects the formalized traffic rules and their hierarchy. Regular collision avoidance rules are followed as long as there is no immediate collision risk, and an emergency operation that executes a last-minute maneuver is immediately activated once a collision becomes likely. For the regular collision avoidance rules, an application-specific maneuver synthesis method based on a search algorithm is developed to efficiently identify actions that are compliant with traffic rules. For emergency operation, we detect imminent collision of the vessels using set-based reachability analysis and design an emergency controller that aims to prevent collisions as much as possible. Rule-compliant actions for both regular and emergency operation are computed online based on our statechart and are used to constrain the RL agent so it can only select verified actions. Our main contributions are:

- We are the first to introduce a safe RL approach that ensures provable satisfaction of open-sea maritime collision avoidance rules, which are formally specified via temporal logic;
- We improve our previously formalized maritime traffic rules [15], newly formalize the last-minute maneuver rule from the Convention on the International Regulations for

Manuscript received 16 February 2024; revised 22 March 2024 and 16 April 2024; accepted 28 April 2024. Date of publication 13 May 2024; date of current version 11 July 2025. This work was supported in part by the Research Training Group ConVeY funded by the German Research Foundation under Grant GRK 2428 and in part by the Project TRAITS funded by the German Federal Ministry of Education and Research. *(Corresponding author: Hanna Krasowski.)*

The authors are with the Technical University of Munich, 80333 Munich, Germany, also with the TUM School of Computation, Information and Technology, Department of Computer Engineering, 85748 Garching, Germany, and also with the Munich Center for Machine Learning, 80333 Munich, Germany (e-mail: hanna.krasowski@tum.de; althoff@tum.de).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIV.2024.3400597.

Digital Object Identifier 10.1109/TIV.2024.3400597

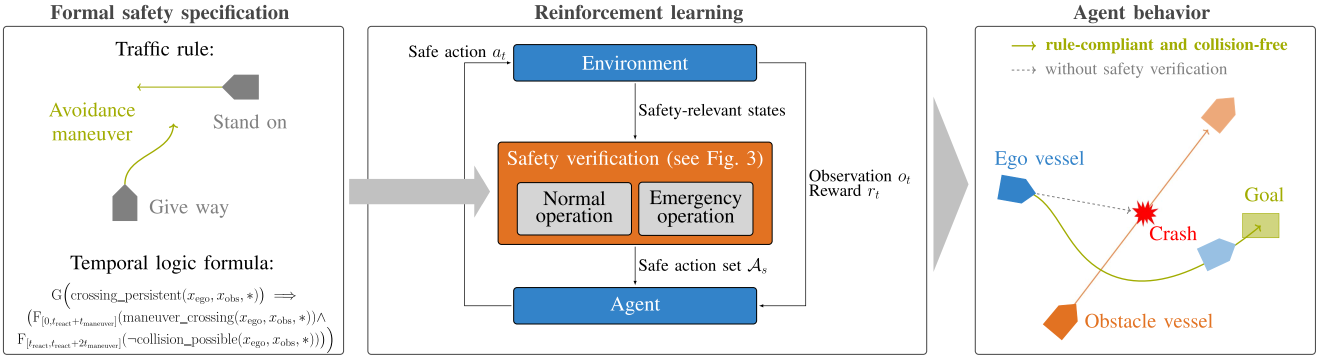

Formal safety specification
Traffic rule:
Avoidance maneuver
Stand on
Give way
Temporal logic formula:
$\mathrm{G}\Big(\text{crossing\_persistent}(x_{\text{ego}}, x_{\text{obs}}, *)\Big) \implies$
$\Big(\mathrm{F}_{[0,t_{\text{react}}+t_{\text{maneuver}}]}(\text{maneuver\_crossing}(x_{\text{ego}}, x_{\text{obs}}, *))\wedge$
$\mathrm{F}_{[t_{\text{react}},t_{\text{react}}+2t_{\text{maneuver}}]}(\neg\text{collision\_possible}(x_{\text{ego}}, x_{\text{obs}}, *))\Big)\Big)$
Reinforcement learning
Safe action a_t
Environment
Safety-relevant states
Safety verification (see Fig. 3)
Normal operation
Emergency operation
Safe action set $\mathcal{A}_s$
Agent
Observation o_t
Reward r_t
Agent behavior
rule-compliant and collision-free
without safety verification
Ego vessel
Goal
Crash
Obstacle vessel

Fig. 1. Proposed provably safe RL approach for autonomous vessels. First, traffic rules for collision avoidance are formalized with temporal logic (see Section III). Based on the formal specification, the set of rule-compliant actions is identified (see Section IV and Section V), which are integrated in the RL process so that the agent can only select actions that are rule-compliant (see Section VI). Note that the statechart in Fig. 3 details the computation of verified rue-compliant actions and comprises two modes: normal operation and emergency operation. The resulting safe agent achieves rule compliance and collision avoidance during training and deployment, while agents without the safety verification of actions violate the formalized traffic rules and collide still after training (see Section VII).

Preventing Collisions at Sea (COLREGS), and develop a rule-compliant emergency controller;

- Our provably safe maneuver synthesis for discrete action spaces efficiently identifies safe actions online;
- We train provably safe RL and safety-informed RL agents on critical maritime traffic situations and evaluate their performance in different deployment configurations on handcrafted and recorded maritime traffic data.

The remainder of this article is structured as follows: We present and discuss related literature in Section II, introduce relevant concepts published preliminarily to this article and state the problem in Section III. We present the formalized traffic rules and prove that a statechart models the traffic rules in Section IV. We describe our rule-compliant maneuver synthesis in Section V. The RL approach is detailed in Section VI. In Section VII, we discuss our experimental results on critical maritime traffic situations and conclude in Section VIII.

II. Related Work

We categorize related work into safety specifications for maritime motion planning, motion planning approaches for autonomous vessels, and provably safe RL.

a) Safety specification for maritime motion planning: The notion of safety in maritime motion planning is usually rule compliance with maritime traffic rules describing collision avoidance maneuvers [16]. The most relevant maritime traffic rules for collision avoidance are specified in the COLREGS [17]. Often, these traffic rules are indirectly integrated in the motion planning approach, e.g., through geometric thresholds [18], [19], [20], [21], [22], [23], virtual obstacles [24], or cost functions [6], [25], [26], [27], [28], [29]. However, these approaches usually do not capture the temporal properties of collision avoidance rules, and the implemented interpretation of the COLREGS is often intransparent.

Another concept is to formalize the traffic rules and directly use them in motion planning. This is a more faithful consideration of traffic rules than the previously mentioned indirect integration. Additionally, the rule formalization is usually parameterized, which eases adaptions. Temporal logic is suited to formalize COLREGS since it captures temporal dependencies and thus can model sophisticated specifications of encounter situations. There are two relevant studies that formalize maritime traffic rules with temporal logic. Torben et al. [30] formalize COLREGS with signal temporal logic for automatic testing of autonomous vessels. This has the advantage that robustness measures specified through signal temporal logic formulas can be used as costs for motion planning approaches, since they quantify rule compliance. Krasowski et al. [15] formalize COLREGS with metric temporal logic and evaluate their compliance on real-world maritime traffic data. They discuss that the COLREGS are currently not well posed for more than two vessels, which needs to be addressed by regulators to make autonomous vessels admissible for commercial deployment in the real-world. How to best employ temporal logic formalizations for motion planning approaches as presented by [15], [30] is an open research question, for which we propose a solution in this work.

b) Motion planning for vessels: The motion planning literature can be categorized into single-agent and multi-agent motion planning problems [31], where multi-agent settings are often distinguished into cooperative [32], [33], [34] and non-cooperative [35], [36] settings. In this article, we regard single-agent motion planning. Maritime motion planners are often divided into three building blocks [37]: a guidance system generating reference trajectories, a control system for tracking reference trajectories, and a state observer.[1] For example, one line of single-agent motion planning research employs search-based algorithms based on motion primitives, e.g., rapidly-exploring random trees [29], [38], [39]. Other studies employ model predictive control (MPC) [26], [28], [40] to obtain an optimal control signal. In contrast to using search algorithms based on a finite amount of motion primitives, MPC directly optimizes the controller in the continuous state and input space. In particular, the studies [26], [28] show promising results on

[1] Often referred to as a navigation system in the maritime literature.

multi-obstacle scenarios and Kufoalor et al. [28] even evaluate their approach in real-world experiments with two obstacle vessels. However, for MPC an optimization problem must be solved repeatedly, which can be computationally costly.

RL is a well-suited machine learning approach to solve single-agent motion planning tasks in uncertain environments [6], [7], [41], [42], [43], [44]. Regarded scenarios are usually on the open sea with other non-reactive dynamic obstacles [7], [42], [43] and static obstacles [6], [41], [44]. To achieve a behavior that adheres to maritime traffic rules, the reward function considers rule compliance to minimize risks, but does not guarantee compliance because the reward function is only maximized [6], [7], [41], [42], [43], [44]. In contrast, provably safe RL approaches ensure safety [8].

c) Provably safe RL: Provably safe RL approaches ensure safety during training and operations. There are three conceptual approaches for provably safe RL [8]: action replacement, action projection, and action masking. In this article, we present an action masking approach, for which the agent can only choose actions that are verified as safe. Most research on action masking considers discrete action spaces; common applications are autonomous driving [45], [46], [47], [48], [49], [50] and power systems [51]. Usually, the action verification is tailored to the specific application and, thus, cannot be directly transferred to other applications.

Another way to distinguish provably safe RL approaches is by the safety specification. Most approaches consider safety specifications that can be formalized as containment in a safe set or avoiding intersection with unsafe sets. A few works regard safety specifications based on temporal logic [52], [53], [54], which can additionally model temporal dependencies in safety specifications. The studies [52], [53] use model checking to determine whether a given action fulfills a linear temporal logic formula, which expresses the safety specification. Their approaches are transferable between applications but limited to discrete action and state spaces. In contrast, Li et al. [54] leverage linear temporal logic specifications to synthesize control barrier functions, which are used to project unsafe actions proposed by the agent to safe actions. This allows them to apply their approach to continuous action and state spaces. However, their approach cannot deal with dynamic obstacles that are not controllable, such as other traffic participants. To the best of our knowledge, we are the first to formulate a provably safe RL approach for the application of autonomous vessels and to include temporal safety specifications in the online safety verification of RL agents while operating in a continuous state space.

III. Preliminaries and Problem Statement

a) Notation and dynamics: We denote sets by calligraphic letters, vectors are boldfaced, and predicates are written in Roman typestyle. The Minkowski sum is defined as $\mathcal{Y}_1 \oplus \mathcal{Y}_2 = \{y_1 + y_2 \,|\, y_1 \in \mathcal{Y}_1, y_2 \in \mathcal{Y}_2\}$ and the set-based multiplication is defined as $\mathcal{Y}_1\mathcal{Y}_2 = \{y_1 y_2 | y_1 \in \mathcal{Y}_1, y_2 \in \mathcal{Y}_2\}$. A traffic rulebook $\langle \Phi, \leq \rangle$ is a tuple where Φ is the set of formalized rules and $\leq$ is the order [55]. We denote that the model Ξ and its initial state ξ entail the rulebook $\langle \Phi, \leq \rangle$ by $\Xi, \xi \models \langle \Phi, \leq \rangle$.

The state of a vessel $\mathbf{s} \in \mathbb{R}^4$ consists of the position $\mathbf{p} = [p_\mathrm{x}, p_\mathrm{y}] \in \mathbb{R}^2$ in the Cartesian coordinate frame as well as the orientation $\theta \in \mathbb{R}$, and the orientation-aligned velocity $v \in \mathbb{R}$. The operator $\mathtt{proj}_\square$ projects a state to the state dimensions indicated by $\square$ and $\mathrm{R}(\Upsilon) = \{\mathrm{R}(\upsilon) | \upsilon \in \Upsilon\}$ denotes the set of rotation matrices for the angles Υ with $\mathrm{R}(\upsilon)$ being the rotation matrix for the angle υ. To model the ego vessel (i.e., the autonomous vessel we control), we use a yaw-constrained model Ω_yc with orientation-aligned acceleration $\mathrm{a} \in \mathbb{R}$ and turning rate $\omega \in \mathbb{R}$ as control inputs:

$$\dot{\mathbf{s}} = \begin{bmatrix} \dot{p_\mathrm{x}} \\ \dot{p_\mathrm{y}} \\ \dot{\theta} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} \cos(\theta)\, v \\ \sin(\theta)\, v \\ \omega \\ \mathrm{a} \end{bmatrix}. \tag{1}$$

The control input is denoted as $\mathbf{u}(t) = [\mathrm{a}(t), \omega(t)]$ and the initial state as $\mathbf{s}_0$.

b) Set-based prediction of vessels: To obtain predictions that enclose all possible behaviors of a traffic participant, the concept of set-based predictions for road traffic participants [56] can be transferred to maritime traffic. The fundamental idea is to define abstract models and perform reachability analysis for them. We first specify the dynamics used for the prediction, and then introduce the reachable sets and occupancy sets. Finally, we discuss the special case of a closed-loop system.

For vessels, we assume that the abstract model is a point-mass model Ω_pm with velocity and acceleration constraints:

$$\begin{aligned} &\dot{p_\mathrm{x}}(t) = v_\mathrm{x}(t),\ \dot{p_\mathrm{y}}(t) = v_\mathrm{y}(t), \\ &\dot{v_\mathrm{x}}(t) = \mathrm{a}_\mathrm{x}(t),\ \dot{v_\mathrm{y}}(t) = \mathrm{a}_\mathrm{y}(t), \\ &\text{subject to} \quad \sqrt{v_\mathrm{x}(t)^2 + v_\mathrm{y}(t)^2} \leq v_\mathrm{pm,max} \\ &\qquad\qquad\quad\ \sqrt{\mathrm{a}_\mathrm{x}(t)^2 + \mathrm{a}_\mathrm{y}(t)^2} \leq \mathrm{a}_\mathrm{pm,max}\,. \end{aligned} \tag{2}$$

The maximum velocity and maximum acceleration are denoted by $\mathrm{a}_\mathrm{pm,max}$ and $v_\mathrm{pm,max}$, respectively. To ensure formal safety of our approach, the two constraints must be chosen such that the point-mass model over-approximates the behavior of vessels using reachset conformance [57]. The state of the model Ω_pm is abbreviated by $\mathbf{x} = [p_\mathrm{x}, p_\mathrm{y}, v_\mathrm{x}, v_\mathrm{y}]$.

The time-point reachable sets for the model Ω_pm are calculated with set-based reachability analysis [58] based on the initial state $\mathbf{s}_0$, time step size Δt, and the time horizon t_pred. Note that the state $\mathbf{s}_0$ is transformed into $\mathbf{x}_0$ by using trigonometry to convert $[v, \theta]$ into $[v_\mathrm{x}, v_\mathrm{y}]$. The time-interval reachable sets are computed as in [58], [59] and are denoted by $\mathcal{R}_{\Delta t}(\mathbf{s}_0, \Omega_\mathrm{pm}, t_\mathrm{pred})$. To obtain the occupancy sets from the time-interval reachable sets, the reachable sets are projected to the position domain and enlarged by the spatial extensions of the vessel $\mathcal{V}$ rotated by all possible reachable orientations using the Minkowski sum:

$$\begin{aligned} &\mathcal{O}_\mathrm{pm}(\mathbf{s}_0, \Omega_\mathrm{pm}, t_\mathrm{pred}, \mathcal{V}) \\ &\quad = \mathtt{proj}_\mathbf{p}\left(\mathcal{R}_{\Delta t}(\mathbf{s}_0, \Omega_\mathrm{pm}, t_\mathrm{pred})\right) \oplus \\ &\quad \left(\mathrm{R}\left(\mathtt{proj}_\theta\left(\mathcal{R}_{\Delta t}(\mathbf{s}_0, \Omega_\mathrm{pm}, t_\mathrm{pred})\right)\right) \mathcal{V}\right). \end{aligned} \tag{3}$$

For a detailed derivation of the occupancy sets, we refer the interested reader to [56, Section V-A].

The occupancy sets $\mathcal{O}_{\text{pm}}$ are calculated for the open-loop system Ω_{pm} since we do not have access to the control input of other traffic participants. However, for an ego vessel, we have a precise model Ω_{yc} and access to the control input. Thus, the forward simulation of our closed-loop system with the control input $\mathbf{u}(t)$ provides the time-point reachable sets. The occupancy is denoted by:

$$\mathcal{O}_{\text{traj}}(\mathbf{s}_0, \Omega_{\text{yc}}, t_{\text{pred}}, \mathcal{V}, \mathbf{u}(t)). \tag{4}$$

c) Problem statement: The COLREGS specify the traffic rules for collision avoidance on the open sea for power-driven vessels in natural language. These traffic rules are satisfiable for two vessels. For more than two vessels, unsatisfiable traffic situations can occur, e.g., a vessel needs to keep its course and speed with respect to one vessel and perform an avoidance maneuver with respect to another vessel. The COLREGS do not specify how to adequately resolve such conflicting situations with more than two vessels. Due to the lack of legal specifications, we regard traffic situations with two vessels only. In particular, we assume:

1) The traffic situation is an open-sea situation without traffic signs, traffic separation zones, or static obstacles;
2) There is one traffic participant vessel *obs* and one autonomous vessel *ego*, which are both power-driven;
3) The dynamics of the autonomous vessel is modeled by (1);
4) The current state of the traffic participant vessel $\mathbf{s}_{\text{obs}}$ is observed without measurement errors;
5) In the initial state of the traffic situation, none of the collision avoidance rules specified in the COLREGS apply.

We define the traffic rulebook $\langle \Phi, \leq \rangle$ that describes the legally relevant collision avoidance rules of the COLREGS given our assumptions 1) and 2). The formal traffic rules are denoted by Φ and the hierarchy between them by $\leq$. Based on the traffic rules, we search for an RL approach, which ensures that the RL agent only selects safe, i.e., rule-compliant, actions leading to rule-compliant trajectories. Thus, the overall problem is to find

$$\begin{aligned} &\pi_s : \mathcal{S} \to \mathcal{A}_s \\ &\text{where} \quad \zeta_{\pi_s} \models \langle \Phi, \leq \rangle \, . \end{aligned} \tag{5}$$

The observation space of the RL agent is $\mathcal{S}$, the set of provably rule-compliant actions is $\mathcal{A}_s$, and the trajectories ζ_{π_s} are solutions of (1) when following the RL policy π_s. To address this problem, we first introduce the rulebook $\langle \Phi, \leq \rangle$, and prove that a statechart Γ entails the rulebook in Section IV. Then, we describe the synthesis of rule-compliant maneuvers and detail the safe-by-design action selection in Section V. Finally, we describe the RL specification in Section VI.

IV. Specification

Our previous work [15] formalizes the COLREGS rules specifying collision avoidance between two power-driven vessels on the open sea. The temporal operators used are G, F, and U, and if there is a subscript, the temporal operator is evaluated over the time interval indicated by the subscript. The operator $\mathrm{G}(\phi)$ evaluates to true iff ϕ is true for all future time steps. In contrast, for the operator $\mathrm{F}(\phi)$, ϕ only has to be true for at least one future time step. The until operator $\phi_1 \mathrm{U} \phi_2$ is true iff ϕ_1 holds true for all time steps until ϕ_2 holds true. In this section, we introduce the legal specification through a rulebook and detail the novel formalization of the emergency rule. Finally, we introduce the statechart Γ and show that it models the specification.

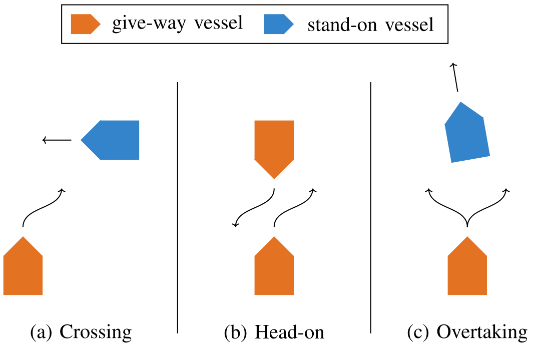


Fig. 2. Encounter situations and rule-compliant maneuvers specified in the COLREGS (adapted from [15]).

A. Traffic Rulebook

Table I lists all formalized rules considered in this work. While the predicates can be evaluated on any two vessels, the predicate arguments are set to be evaluated for the ego vessel with respect to an obstacle vessel according to the COLREGS. The traffic rule R_2 enforces a safe speed, which is trivially ensured through the ego vessel dynamics. Thus, we do not include this rule in the traffic rulebook.

Definition 1 (Rules Φ): The rulebook consist of rules R_1 and $R_3 - R_6$ specified in Table I.

We introduce the emergency rule R_1 to reflect the COLREGS specification that if the other vessel does not take appropriate actions for collision avoidance, the ego vessel has to react and perform a last-minute maneuver for collision risk minimization.

COLREGS Requirement 1 (Rulebook order $\leq$): Rule R_1 is always prioritized over rules R_3 - R_5, and R_6 has the lowest priority. Rules R_3 - R_5 are all of equal priority.

The predicates of rule R_1 are detailed in Section IV-B. Note that we use the emergency maneuver to describe the last-minute maneuver, through which the ego vessel minimizes the collision risk and thereby achieves legal safety. Yet, in the literature, the term failsafe planning is also frequently used [8], [60].

Rules R_3 - R_6 describe how vessels have to behave in a COLREGS encounter situation. In these encounter situations, the vessels are on a collision course meaning that the vessels would collide in the near future if no appropriate collision avoidance measures are taken. There are three different encounter situations specified in the COLREGS as illustrated in Fig. 2: overtaking (R_5, R_6), crossing (R_3, R_6), and head-on encounters (R_4). In an encounter, a vessel can be a give-way or a stand-on vessel. A give-way vessel is required to change course and perform a collision avoidance maneuver. A stand-on vessel has the obligation to keep its course and speed. The predicate for

TABLE I
OVERVIEW FORMALIZED MARINE TRAFFIC RULES INTEGRATED IN THE SAFETY VERIFICATION

Rule	**Temporal logic formula**
$R_1^\ddagger$	$\mathrm{G}(\text{is\_emergency}(\mathbf{s}_{\text{ego}}, \mathbf{s}_{\text{obs}}, *) \implies (\text{emergency\_maneuver}\ \mathrm{U}\ \text{is\_emergency\_resolved}(\mathbf{s}_{\text{ego}}, \mathbf{s}_{\text{obs}}, *)))$
R_2	$\mathrm{G}\Big(\text{safe\_speed}(\mathbf{s}_{ego}, v_{\max})\Big)$
$R_3^\dagger$	$\mathrm{G}\Big(\text{persistent\_crossing}(\mathbf{s}_{\text{ego}}, \mathbf{s}_{\text{obs}}, *) \implies$ $(\mathrm{F}_{[0,t_{\text{react}}+t_{\text{maneuver}}]}(\text{maneuver\_crossing}(\mathbf{s}_{\text{ego}}, \mathbf{s}_{\text{obs}}, *)) \wedge \mathrm{F}_{[t_{\text{react}},t_{\text{react}}+2t_{\text{maneuver}}]}(\neg\text{collision\_possible}(\mathbf{s}_{\text{ego}}, \mathbf{s}_{\text{obs}}, t^{\text{check}}_{\text{horizon}})))\Big)$
$R_4^\dagger$	$\mathrm{G}\Big(\text{persistent\_head\_on}(\mathbf{s}_{\text{ego}}, \mathbf{s}_{\text{obs}}, *) \implies$ $(\mathrm{F}_{[0,t_{\text{react}}+t_{\text{maneuver}}]}(\text{maneuver\_head\_on}(\mathbf{s}_{\text{ego}}, \mathbf{s}_{\text{obs}}, *)) \wedge \mathrm{F}_{[t_{\text{react}},t_{\text{react}}+2t_{\text{maneuver}}]}(\neg\text{collision\_possible}(\mathbf{s}_{\text{ego}}, \mathbf{s}_{\text{obs}}, t^{\text{check}}_{\text{horizon}})))\Big)$
$R_5^\dagger$	$\mathrm{G}\Big(\text{persistent\_overtake}(\mathbf{s}_{\text{ego}}, \mathbf{s}_{\text{obs}}, *) \implies$ $(\mathrm{F}_{[0,t_{\text{react}}+t_{\text{maneuver}}]}(\text{maneuver\_overtake}(\mathbf{s}_{\text{ego}}, \mathbf{s}_{\text{obs}}, *)) \wedge \mathrm{F}_{[t_{\text{react}},t_{\text{react}}+2t_{\text{maneuver}}]}(\neg\text{collision\_possible}(\mathbf{s}_{\text{ego}}, \mathbf{s}_{\text{obs}}, t^{\text{check}}_{\text{horizon}})))\Big)$
R_6	$\mathrm{G}\Big(\text{keep}(\mathbf{s}_{\text{ego}}, \mathbf{s}_{\text{obs}}, *) \implies (\text{no\_turning}(\mathbf{s}_{\text{ego}}, *)\mathrm{U}\,\neg\text{keep}(\mathbf{s}_{\text{ego}}, \mathbf{s}_{\text{obs}}, *))\Big)$

Note: Additional arguments are abbreviated by $*$, rules adapted from [15] are marked with †, and new rules are marked with ‡.

determining a stand-on vessel is keep (see Appendix A). The stand-on rule R_6 has the lowest priority since whenever the other vessel changes its course so that the ego vessel becomes the give-way vessel, the give-way rules R_3 to R_5 are applied (see COLREGS Requirement 1).

To formalize that a give-way encounter is persistent for at least the reaction time, we use the following temporal logic specification, where {give_way} can take the values from {crossing, head_on, overtake} (see Appendix A) and $*$ denotes additional arguments for the predicates:

$$\begin{aligned}&\text{persistent\_\{give\_way\}}(\mathbf{s}_{\text{ego}}, \mathbf{s}_{\text{obs}}, *)\\ &\quad = \neg\text{\{give\_way\}}(\mathbf{s}_{\text{ego}}, \mathbf{s}_{\text{obs}}, *) \wedge\\ &\quad\quad \mathrm{G}_{[\Delta \mathrm{t}, \mathrm{t}_{\text{react}}]}(\text{\{give\_way\}}(\mathbf{s}_{\text{ego}}, \mathbf{s}_{\text{obs}}, *)).\end{aligned}$$

We assume that both vessels keep their course and speed to obtain rule-compliant predictions for their future states. These predicted states allow us to evaluate ahead of time if the encounter situation will persist long enough so that the ego vessel has to perform a collision avoidance maneuver. The reaction time t_{react} does not indicate the minimum required reaction time of a human operator but instead specifies how much time the human operator would require to decide if the encounter situation persists. Given a give-way encounter is detected, a rule-compliant collision avoidance maneuver has to be conducted until $\neg$collision_possible evaluates to true (see Table I R_3 - R_5). The time interval for performing a rule-compliant maneuver is $t_{\text{react}} + 2t_{\text{maneuver}}$, where $2t_{\text{maneuver}}$ approximates the time required for the maneuvering.

COLREGS Requirement 2 (Maneuvering priority): Given a rule R_i for $i \in \{3, \ldots, 5\}$ applies, rules R_j for $j \neq i \wedge j \in \{3, \ldots, 5\}$ are not applied until $\neg$collision_possible is true.

B. Emergency Rule Predicates

We use the predicate collision_possible to determine if two vessels are on a collision course for rules R_3 - R_6. Because the rules R_3 - R_6 assume a constant velocity, we use the velocity obstacle concept [61] for this predicate. However, the velocity obstacle concept is not sufficient for detecting imminent risk as necessary for R_1. Thus, we present four predicates in this section that are relevant for our formalization of rule R_1.

First, we define an auxiliary position predicate determining if vessel m is in a relative orientation sector of vessel l:

$$\begin{aligned}&\text{in\_sector}(\mathbf{s}_l, \mathbf{s}_m, \underline{\beta}, \overline{\beta}) \iff\\ &\quad \mathbf{h}_{\underline{\beta}}^T\mathtt{proj}_{\mathbf{p}}(\mathbf{s}_m) - b_{l,\underline{\beta}} \leq 0 \wedge\\ &\quad \mathbf{h}_{\overline{\beta}}^T\mathtt{proj}_{\mathbf{p}}(\mathbf{s}_m) - b_{l,\overline{\beta}} > 0,\end{aligned}$$

where the lower relative orientation is $\underline{\beta}$ and the upper relative orientation is $\overline{\beta}$ relative to the orientation of vessel l. The normal vector $\mathbf{h}_i$ is the unit vector in the direction $i - \pi/2$ and $b_{l,i}$ is the offset to the origin for a line through the position of vessel l in the direction i. We illustrate the sector predicate with two specific usages in Fig. 4.

Second, we use set-based prediction for rule R_1 to detect potential collisions in the near future. In particular, we predict the future occupancy of the obstacle vessel until the time horizon t_{pred} as described in (3) and that of the ego vessel as in (4), for the control sequence $\mathbf{u}_{\text{keep}}(t) = [0\ \text{m/s}^2\ , 0\,\text{rad/s}]$ to keep course and speed as demanded for stand-on vessels. If the ego occupancy and the predicted occupancy of the obstacle vessel intersect, the ego vessel is in an emergency situation:

$$\begin{aligned}&\text{is\_emergency}(\mathbf{s}_{\text{ego}}, \mathbf{s}_{\text{obs}}, \mathcal{V}_{\text{ego}}, \mathcal{V}_{\text{obs}}, t_{\text{pred}}, \mathbf{u}_{\text{keep}}(t)) \iff\\ &\quad \exists t \in [t_0, t_0 + t_{\text{pred}}] : \mathcal{O}_{\text{pm}}(\mathbf{s}_0, \Omega_{\text{pm}}, t, \mathcal{V}_{\text{obs}}) \cap\\ &\quad \mathcal{O}_{\text{traj}}(\mathbf{s}_{\text{ego}}, \Omega_{\text{yc}}, t, \mathcal{V}_{\text{ego}}, \mathbf{u}_{\text{keep}}(t)) \neq \emptyset,\end{aligned}$$

where t_0 is the current time.

Third, the predicate emergency_maneuver describes a maneuver that minimizes the risk of collision for the specific traffic situation. We detail our interpretation of emergency_maneuver in Section V-A.

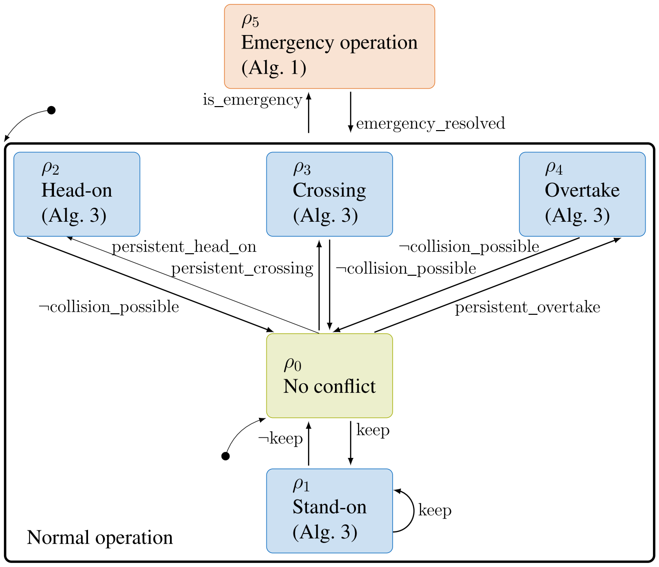


Fig. 3. Statechart Γ modeling the legal safety specification with predicates at the transitions. The states for the regular collision avoidance rules R_3 - R_6 are depicted in blue and the emergency operation state for rule R_1 in red. For safety verification of actions, the algorithms identifying the set of rule-compliant actions (indicated in brackets) are employed given the current state ρ_i of the statechart.

Fourth, an emergency situation is resolved when the obstacle vessel is behind the ego vessel, is moving away from the ego vessel, and the Euclidean distance between both is larger than a specified minimum distance d_{resolved}:

$$\begin{aligned}&\text{is\_emergency\_resolved}(\mathbf{s}_{\text{ego}}, \mathbf{s}_{\text{obs}}, d_{\text{resolved}}) \iff\\ &\underbrace{\text{in\_sector}(\mathbf{s}_{\text{ego}}, \mathbf{s}_{\text{obs}}, 3\pi/2, \pi/2)}_{\textbf{obstacle is behind}} \wedge\\ &\underbrace{\texttt{unit\_v}(\mathbf{s}_{\text{obs}})^T\, \texttt{unit\_v}(\mathbf{s}_{\text{ego}}) \leq 0}_{\textbf{moving away}} \wedge\\ &\underbrace{\|\texttt{proj}_{\mathbf{p}}(\mathbf{s}_{\text{obs}}) - \texttt{proj}_{\mathbf{p}}(\mathbf{s}_{\text{ego}})\|_2 \leq d_{\text{resolved}}}_{\textbf{distance between vessels is large enough}}\end{aligned}$$

where the unit orientation vector of a state is $\texttt{unit\_v}(\mathbf{s}) = [\cos(\texttt{proj}_\theta(\mathbf{s})), \sin(\texttt{proj}_\theta(\mathbf{s}))]$.

C. Specification-Compliant Statechart

The overall rule specification is modeled by the statechart Γ in Fig. 3. Due to assumption 5), the initial state in every traffic situation is the state ρ_0. There are two main states for normal operation and emergency operation. During normal operation, whenever the predicate collision_possible is true, the corresponding maneuver state for R_3 - R_6 (see blue states in Fig. 3) is entered and the collision avoidance maneuver is started.

Proposition 1: For the states ρ_i, $\forall i \in \{1,\ldots,4\}$, the predicate collision_possible is true.

Proof: This follows directly from the definition of the predicates keep, head_on, crossing, and overtake (see Appendix A), which are true for the states of the statechart $\rho_1 - \rho_4$, respectively. ■

Lemma 1: For two specific vessels, at most one of the predicates keep, head_on, crossing, or overtake can be true at the same time.

Proof: The predicates keep, head_on, crossing, and overtake cannot apply at the same time due to their mutually exclusive specification. The detailed proof is in Appendix B. ■

If an emergency situation is detected, the statechart transitions to the emergency operation state until the emergency situation is resolved.

Theorem 1: It holds that $\Gamma, \rho_0 \models \langle \Phi, \leq \rangle$ for the statechart Γ, its initial state ρ_0, and the rulebook $\langle \Phi, \leq \rangle$.

Proof: The initial state ρ_0 fulfills the rulebook by assumption 5) (see Section III-C). We continue proving the compliance with each rule:

(I) R_1: If is_emergency is true, R_1 applies and R_3–R_6 do not (see COLREGS Requirement 1), which is realized by transitioning to ρ_5 (see Fig. 3). The state ρ_5 can only be exited iff is_emergency_resolved evaluates to true. Thus, the transition to and from ρ_5 directly represents R_1.

If collision_possible $\wedge$ ¬is_emergency is true, then Γ has to represent rules R_3 - R_6. Whenever collision_possible becomes true, it can be deduced from Lemma 1 and Preposition 1 that the statechart transitions to a state ρ_i, $i \in \{1,\ldots,4\}$.

(II) R_3–R_5: Based on COLREGS Requirement 2, once a rule R_3–R_5 applies, i.e., the statechart is in either of the states ρ_i, $i \in \{2,\ldots,4\}$, the respective avoidance maneuver has to be conducted until ¬collision_possible $\vee$ is_emergency is true. For is_emergency, we showed in case (I) of this proof that Γ models $\langle \Phi, \leq \rangle$. For ¬collision_possible, the statechart Γ transitions to ρ_0.

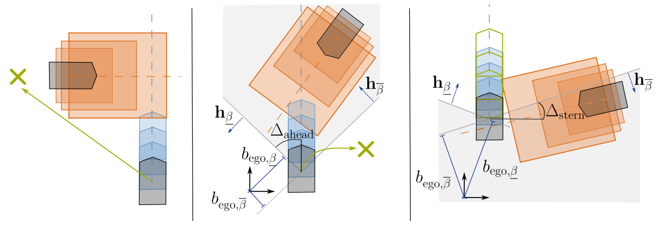
(III) R_6: Once rule R_6 applies, i.e., keep is true, the statechart transitions to ρ_1 and stays there until ¬keep $\vee$ ¬collision_possible $\vee$ is_emergency. If ¬keep $\wedge$ collision_possible, an encounter of higher priority is present (see COLREGS Requirement 1) and R_3 - R_5 apply. In this situation, the statechart transitions to the states ρ_i for $i \in \{2,\ldots,4\}$ and the remaining proof steps are stated in case (III). Identically to case (III), if ¬collision_possible is true, the statechart Γ transitions to ρ_0 and if is_emergency the statechart transitions to ρ_5. ■

V. Rule-Compliant Maneuver Synthesis

Given our specification-compliant statechart Γ, we need to identify rule-compliant actions for the individual states ρ_i of the statechart. Trivially, for the state ρ_0 all actions are rule-compliant since no rules apply. We introduce the synthesis of emergency maneuvers in Section V-A and of encounter maneuvers in Section V-B. Finally, we detail how we ensure a selection of only safe actions for the RL agent in Section V-C.

A. Emergency Maneuver

Once we detect an emergency situation, i.e., the statechart is in ρ_5, the ego vessel is legally required to evade the obstacle vessel in a manner that minimizes the risk of collision. In similar motion planning applications, such as autonomous driving [10], autonomous aerial traffic [62], or human-robot environments [63], states that are safe for infinite time are used to identify a legally


$\mathbf{h}_\beta$
$\mathbf{h}_{\overline{\beta}}$
Δ_{ahead}
$b_{\text{ego},\beta}$
$b_{\text{ego},\overline{\beta}}$
Δ_{stern}
$b_{\text{ego},\underline{\beta}}$

(a) Base (b) Ahead (c) Stern

Fig. 4. Emergency controller modes with set-based occupancy prediction of obstacle vessel in orange and the occupancy of the ego vessel in blue for several time intervals. The orientation of the ego vessel and the obstacle vessel are indicated with dashed lines and emergency maneuver is depicted by green arrows or occupancies. The green cross indicates the target position for the base and ahead modes. The sectors, for which the predicate in_sector is true, are shown in gray for the ahead and stern mode. The visualization of the sectors includes the arguments of predicate in_sector in dark blue and the point of origin in black.

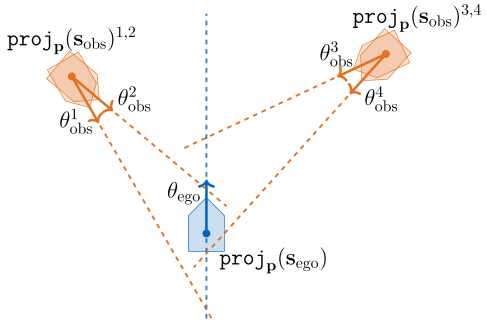

$\texttt{proj}_\mathbf{p}(\mathbf{s}_{\text{obs}})^{1,2}$
$\texttt{proj}_\mathbf{p}(\mathbf{s}_{\text{obs}})^{3,4}$
θ^3_{obs}
θ^2_{obs}
θ^1_{obs}
θ^4_{obs}
θ_{ego}
$\texttt{proj}_\mathbf{p}(\mathbf{s}_{\text{ego}})$

Fig. 5. Visualization of turning direction cases. The obstacle vessel is depicted in orange and the ego vessel in blue. Arrows indicate orientations and positions are marked with dots. The turning direction case is indicated by the superscript. For cases 1 and 3, the ego vessel should turn right and for the cases 2 and 4, the ego vessel should turn left.

safe emergency maneuver. In contrast, the current COLREGS do not state specifically how to interpret “minimize risk” or the characteristics of an invariably safe state. Thus, we cannot provide a formal specification. Consequently, we cannot verify risk minimizing behavior. Nevertheless, we identify three situations in which different emergency maneuvers are appropriate: base mode, ahead mode, and stern mode (see Fig. 4).

In the ahead case (see Fig. 4(b)), the obstacle vessel is in the ahead sector in front of the ego vessel, and the orientation difference between the ego vessel orientation and the reversed orientation of the obstacle vessel is at most Δ_{ahead}. This can be formalized as:

$$\begin{aligned}&\text{ahead\_emergency}(\mathbf{s}_{\text{ego}}, \mathbf{s}_{\text{obs}}, \Delta_{\text{ahead}}) \iff\\ &\quad \texttt{in}(\rho_5) \wedge \neg\text{orientation\_delta}(\mathbf{s}_{\text{ego}}, \mathbf{s}_{\text{obs}}, \Delta_{\text{ahead}}, \pi) \wedge\\ &\quad \text{in\_sector}(\mathbf{s}_{\text{ego}}, \mathbf{s}_{\text{obs}}, -\Delta_{\text{ahead}}, \Delta_{\text{ahead}}),\end{aligned} \tag{6}$$

where the predicate $\texttt{in}(\rho_5)$ evaluates to true if and only if the statechart Γ is in base state ρ_5. In this ahead situation, steering to the stern of the obstacle vessel would lead to an even more critical situation, as both vessels would encounter each other head-on, given the obstacle vessel approximately keeps its speed and course. Thus, we instead require the ego vessel to turn 90°. The direction of turning is determined as presented in Fig. 5. Depending on the situation, turning 90° can be enough to resolve the emergency situations. Yet, if the emergency is not resolved and the traveled distance of the ego vessel from the start of the maneuver is larger than $d_{\text{min,ahead}}$, the emergency controller switches to the base mode (see Fig. 4(a)) and steers the ego vessel behind the stern of the obstacle vessel.

The stern case is necessary for situations where the obstacle vessel is almost astern of the ego vessel and still relatively far away (see Fig. 4(c)):

$$\begin{aligned}&\text{stern\_emergency}(\mathbf{s}_{\text{ego}}, \mathbf{s}_{\text{obs}}, \Delta_{\text{stern}}, \mathbf{u}_{\text{acc}}(t), \mathcal{V}_{\text{ego}}, \mathcal{V}_{\text{obs}},\\ &\qquad\qquad t_{\text{pred}}) \iff\\ &\quad \texttt{in}(\rho_5) \wedge\\ &\quad \text{in\_sector}(\mathbf{s}_{\text{ego}}, \mathbf{s}_{\text{obs}}, 3\pi/2 + \Delta_{\text{stern}}, \pi/2 + \Delta_{\text{stern}}) \wedge\\ &\quad \neg\text{is\_emergency}(\mathbf{s}_{\text{ego}}, \mathbf{s}_{\text{obs}}, \mathcal{V}_{\text{ego}}, \mathcal{V}_{\text{obs}}, t_{\text{pred}}, \mathbf{u}_{\text{acc}}(t)),\end{aligned} \tag{7}$$

with the control sequence $\mathbf{u}_{\text{acc}}(t) = [a_{\text{stern}}, 0\,\text{rad/s}],\ \forall t \leq t_{\text{react}}$ and then $0\ \text{m/s}^2,\ 0\,\text{rad/s}],\ \forall t \leq t_{\text{react}} < t \leq t_{\text{pred}}$. By using the set-based prediction within this predicate, we ensure that we only use this controller mode if it is certain that accelerating would resolve the situation. In such a situation, performing an emergency maneuver that navigates the ego vessel to the stern of the obstacle vessel would be an unnecessarily long detour, given that a short acceleration period would also resolve the emergency situation.

For the base case (see Fig. 4(a)), the emergency situation can be safely resolved by steering to a position behind the stern of the obstacle vessel. The base emergency situation is formalized by:

$$\begin{aligned}&\text{base\_emergency} \iff\\ &\quad \texttt{in}(\rho_5) \wedge \neg\text{ahead\_emergency} \wedge \neg\text{stern\_emergency} \wedge\\ &\quad \neg\text{is\_emergency\_resolved}.\end{aligned}$$

Algorithm 1 summarizes the control mode selection when entering the emergency operation state (see Fig. 3) and is an instantiation of the predicate emergency_maneuver of rule R_1 in Table I for our problem statement. For base and ahead modes, the target positions are depicted in Fig. 4 and obtained with the functions `get_target_ahead` and `get_target_base`, respectively. Given the target position, a reachable desired position given the current state is identified and a control input toward this desired position is generated (for details on the controller design see Appendix C). The controller is abbreviated by the function `tracking_controller`.

B. Encounter Maneuvers

Given a persistent give-way encounter is detected (i.e., the statechart in Fig. 3 transitions to one of the respective blue states $\rho_1, \ldots, \rho_4$), we identify safe actions that result in safe maneuvers resolving the encounter.

Set-based predictions are well suited to verify that no collisions can occur if not all vessels comply with the regular collision avoidance rules R_3 - R_6. Still, for the regular collision avoidance rules, the implicit assumption in the COLREGS is that both vessels comply with them. Thus, for identifying actions

Algorithm 1: $\text{emergency\_maneuver}(\mathbf{s}_{\text{ego}}, \mathbf{s}_{\text{obs}}, *)$.

```
Input: current state of ego vessel $\mathbf{s}_{\text{ego}}$, current state of
  obstacle vessel $\mathbf{s}_{\text{obs}}$, emergency mode $mode$, initial time
  $t_0$, time step size $\Delta t$, acceleration control sequence $\mathbf{u}_{\text{acc}}(t)$
Output: control input $\mathbf{u}(t_i)$
$\mathbf{s}_{\text{ego},0} = \mathtt{proj}_{\mathbf{p}}(\mathbf{s}_{\text{ego}}), \mathbf{s}_{\text{obs},0} = \mathtt{proj}_{\mathbf{p}}(\mathbf{s}_{\text{obs}}), t_i = t_0$
while $\neg$emergency_resolved do
    if $\|\mathtt{proj}_{\mathbf{p}}(\mathbf{s}_{\text{ego},0}) - \mathtt{proj}_{\mathbf{p}}(\mathbf{s}_{\text{ego}})\|_2 > d_{\text{min,ahead}} \wedge$
    $mode = \text{ahead}$ then
        $mode \leftarrow \text{base}$
    end if
    if $mode = \text{stern}$ then
        $\text{a}, \omega \leftarrow \mathbf{u}_{\text{acc}}(t_i)$
    else if $mode = \text{ahead}$ then
        $\mathbf{p}_{\text{target}} \leftarrow \mathtt{get\_target\_ahead}(\mathbf{s}_{\text{ego}}, \mathbf{s}_{\text{ego},0}, \mathbf{s}_{\text{obs},0})$
        $\text{a}, \omega \leftarrow \mathtt{tracking\_controller}(\mathbf{s}_{\text{ego}}, \mathbf{p}_{\text{target}})$
    else
        $\mathbf{p}_{\text{target}} \leftarrow \mathtt{get\_target\_base}(\mathbf{s}_{\text{ego}}, \mathbf{s}_{\text{obs}})$
        $\text{a}, \omega \leftarrow \mathtt{tracking\_controller}(\mathbf{s}_{\text{ego}}, \mathbf{p}_{\text{target}})$
    end if
    return $\mathbf{u}(t_i) = [\text{a}, \omega]$
    $\mathbf{s}_{\text{ego}}, \mathbf{s}_{\text{obs}} \leftarrow \mathtt{step\_environment}(\text{a}, \omega)$
    $t_i \leftarrow t_i + \Delta t$
end while
```

of the ego vessel that are rule-compliant with these rules, we can use a rule-compliant prediction for the obstacle vessel. For the three encounter situations specified (see Fig. 2), we differentiate between the ego vessel being the give-way (R_3–R_5 apply) and stand-on vessel (R_6 applies). First, we detail the verification of actions given the ego vessel is the stand-on vessel, i.e., $\mathtt{in}(\rho_1)$. Then, we describe the more intricate synthesis given that the ego vessel is the give-way vessel (ρ_i where $i \in \{2, \ldots, 4\}$), and finally, summarize our encounter action synthesis.

a) Stand-on maneuver synthesis for ρ_1: The trivial action for the predicate keep is $a_{\text{keep}} = [\text{a} = 0\ \text{m/s}^2, \omega = 0\,\text{rad/s}]$, i.e., keeping course and speed. Note that for this trivial action there is no explicit maneuver time and the action space needs to be restricted to this action until the ego vessel is not the stand-on vessel anymore or an emergency is detected (see Fig. 3).

b) Give-way maneuver synthesis for $\rho_2 - \rho_4$: For all give-way maneuvers, a significant change of orientation (i.e., $\Delta_{\text{large\_turn}}$) is required so that other traffic participants can identify give-way maneuvers (see Fig. 2). For head-on and crossing encounters, the give-way vessel is always obliged to turn toward the right. For the overtake encounter, the suited turning direction depends on the orientation of the obstacle vessel, but this is not further specified in the COLREGS. For our maneuver synthesis, the turning direction is to the left if the orientation of the obstacle vessel is more to the right than the orientation of the ego vessel, and otherwise turning direction is to the right.

Given the turning direction, we identify candidate actions, construct maneuvers based on them, and verify if a maneuver complies with the rules. Candidate actions lead to trajectories that already fulfill the minimal turning requirement within the maneuver segment time t_{m}. A maneuver is verified if the predicate collision_possible is false at the end of the maneuver and the occupancies of both vessels do not intersect during the maneuver:

$$\begin{aligned}
&\text{maneuver\_verified}(\mathbf{u}_{\text{m}}(t), \mathbf{s}_{\text{ego}}, \mathbf{s}_{\text{obs}}, t^{\text{check}}_{\text{horizon}}, \mathbf{u}_{\text{keep}}(t), \\
&\qquad \mathcal{V}_{\text{ego}}, \mathcal{V}_{\text{obs+}}) \iff \\
&\quad \neg\text{collision\_possible}(\mathbf{s}_{\text{ego},t_{\text{end}}}, \mathbf{s}_{\text{obs},t_{\text{end}}}, t^{\text{check}}_{\text{horizon}}) \wedge \\
&\qquad \forall t \in [t_0, t_0 + t_{\text{end}}] : \mathcal{O}_{\text{traj}}(\mathbf{s}_{\text{ego}}, \Omega_{\text{yc}}, t, \mathcal{V}_{\text{ego}}, \mathbf{u}_{\text{m}}(t)) \cap \\
&\qquad \mathcal{O}_{\text{traj}}(\mathbf{s}_{\text{obs}}, \Omega_{\text{yc}}, t, \mathcal{V}_{\text{obs+}}, \mathbf{u}_{\text{keep}}(t)) = \emptyset,
\end{aligned} \tag{8}$$

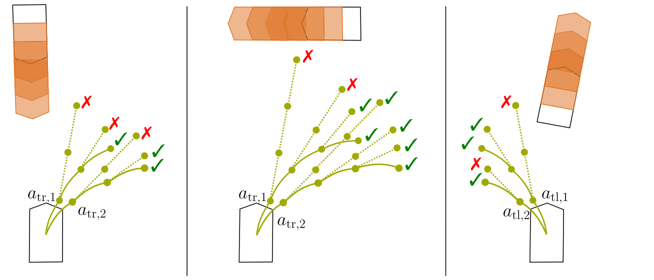
where t_0 is the current time, $t_{\text{end}} \in n\, t_{\text{m}}$ is the time horizon of the maneuver with $n \in \mathbb{N}^+$, $\mathbf{s}_{\text{ego},t_{\text{end}}}$ is the final state of the maneuver, and $\mathbf{u}_{\text{m}}(t)$ is the control sequence for the maneuver trajectory. The predicted obstacle state at t_{end} is $\mathbf{s}_{\text{obs},t_{\text{end}}}$ and the set $\mathcal{V}_{\text{obs+}}$ is the spatial extensions of the obstacle enlarged by the safety factor $d_{\text{obs,safety}}$ for width and length. The occupancy of the obstacle vessel is based on the assumption that the obstacle vessel will keep its speed and course, i.e., the control sequence $\mathbf{u}_{\text{keep}}(t)$. This assumption is compliant with the COLREGS collision avoidance rules for the crossing and overtake encounter. In case of the head-on encounter, the predicted trajectory for the obstacle vessel is a conservative prediction since the obstacle vessel would also need to evade to the right to be rule-compliant. Assuming that the obstacle vessel will keep its course and speed leads to the fact that the ego vessel has to turn more to resolve the encounter situation.

With the turning direction and the maneuver verification predicate defined in (8), we want to determine all actions that lead to verified maneuvers. The generation of maneuvers based on candidate actions is computed by a breadth-first search with rule-compliant pruning. The search algorithm is detailed in Algorithm 2. Note that to obtain a control sequence for multiple actions, we introduce the function $\mathtt{a2u}$. For a maneuver segment trajectory, the control input corresponding to an action, is held constant for a maneuver segment time t_{m} while (1) is forward simulated. We initialize a search tree with a maneuver segment trajectory resulting from the candidate turning action a_{c}. A candidate action a_{c} ensures that the orientation of the ego vessel changes at least $\Delta_{\text{large\_turn}}$ within t_{m}. Potentially, this first maneuver segment trajectory results already in a verifiable maneuver (cf. Algorithm 2, line 2–3). If not, the search tree is extended by (a) a maneuver segment trajectory based on the candidate action a_{c} (cf. Algorithm 2, line 17–18), and (b) with maneuver segment trajectories for each action $a \in \mathcal{A}_{\text{acc}}$, which keep the speed or accelerate the ego vessel (cf. Algorithm 2, line 19–21). If the action of the maneuver segment trajectory that should be extended (obtained with the function $\mathtt{last}$) does not correspond to a_{c}, the maneuver is only extended with the previously used action (cf. Algorithm 2, line 24–25). This has the effect that the vessel does not switch between different accelerations during the maneuver. The expansion of the search tree is stopped (a) if at least one trajectory sequence is verified

Algorithm 2: `build_st`.

```
Input: candidate action $a_\mathrm{c}$, accelerating actions $\mathcal{A}_\mathrm{acc}$, current state of obstacle vessel $\mathbf{s}_\mathrm{obs}$, current state of ego vessel $\mathbf{s}_\mathrm{ego}$, maneuver segment time $t_\mathrm{m}$, maneuver horizon $t_\mathrm{max,m}$, control sequence $\mathbf{u}_\mathrm{keep}(t)$
Output: verified part of search tree $\mathcal{G}$
$t_\mathrm{end} \leftarrow t_\mathrm{m}, \mathcal{G} \leftarrow \{a_\mathrm{c}\}$
$\mathbf{u}_\mathrm{c}(t) = \mathtt{a2u}(a_\mathrm{c})$
if $\mathrm{maneuver\_verified}(\mathbf{u}_\mathrm{c}(t), \dots)$ then
    return $\mathcal{G}$
else
    $\mathcal{U}_\mathrm{m} \leftarrow \{\mathbf{u}_\mathrm{c}(t)\}$
    while $\neg\mathrm{maneuver\_verified}(\mathbf{u}_\mathrm{m}(t), \dots)$ $\forall\, \mathbf{u}_\mathrm{m}(t) \in \mathcal{U}_\mathrm{m}$ do
        $\mathcal{U}_\mathrm{m} \leftarrow \emptyset$
        $\mathcal{G}_\mathrm{temp} \leftarrow \emptyset$
        if $t_\mathrm{end} < t_\mathrm{max,m}$ then
            $t_\mathrm{end} \leftarrow t_\mathrm{end} + t_\mathrm{m}$
        else
            return $\mathcal{G} \leftarrow \emptyset$
        end if
        for $a' \in \mathcal{G}$ do
            if $\mathtt{last}(a') = a_\mathrm{c}$ then
                $\mathbf{u}_\mathrm{m}(t) \leftarrow \mathtt{a2u}(a') + \mathtt{a2u}(a_\mathrm{c})$
                $\mathcal{U}_\mathrm{m} \leftarrow \mathbf{u}_\mathrm{m}(t), \mathcal{G}_\mathrm{temp} \leftarrow [a', a_\mathrm{c}]$
                for $a_\mathrm{acc} \in \mathcal{A}_\mathrm{acc}$ do
                    $\mathbf{u}_\mathrm{m}(t) \leftarrow \mathtt{a2u}(a') + \mathtt{a2u}(a_\mathrm{acc})$
                    $\mathcal{U}_\mathrm{m} \leftarrow \mathbf{u}_\mathrm{m}(t), \mathcal{G}_\mathrm{temp} \leftarrow [a', a_\mathrm{acc}]$
                end for
            else
                $\mathbf{u}_\mathrm{m}(t) \leftarrow \mathtt{a2u}(a') + \mathtt{a2u}(\mathtt{last}(a'))$
                $\mathcal{U}_\mathrm{m} \leftarrow \mathbf{u}_\mathrm{m}(t), \mathcal{G}_\mathrm{temp} \leftarrow [a', \mathtt{last}(a')]$
            end if
        end for
        $\mathcal{G} \leftarrow \mathcal{G}_\mathrm{temp}$
    end while
end if
return $\mathcal{G}$
```

for the current search tree depth, i.e., for time horizon t_end, or (b) if the maneuver horizon $t_\mathrm{max,m}$ is reached. Note that $t_\mathrm{max,m}$ follows from the rule specification and is $t_\mathrm{react} + 2t_\mathrm{maneuver}$. The search tree generation is illustrated in Fig. 6 for three give-way encounters. Due to the rule-compliant pruning, our search algorithm has the time complexity $\mathcal{O}(n\, N_\mathrm{c}\, N_\mathrm{acc})$ for tree generation where $N_\mathrm{c} \in \mathbb{N}^+$ is the number of candidate actions a_c, and $N_\mathrm{acc} \in \mathbb{N}^+$ is the number of actions in $\mathcal{A}_\mathrm{acc}$.



(a) Head-on (b) Crossing (c) Overtake

Fig. 6. Example search trees for the three give-way encounter situations, in which the ego vessel has to give way. The prediction of the obstacle vessel is depicted in orange and the maneuver segment trajectories in green with a dot for the final state. The trajectories based on actions from $\mathcal{A}_\mathrm{acc}$ are displayed as dashed line. Note that we display only one trajectory based on actions from $\mathcal{A}_\mathrm{acc}$ for visualization purposes. The candidate actions initializing the search trees are $a_{d,1}$ and $a_{d,2}$ where d is either tr for turning right and tl for turning left. The mark ✓ indicates that the maneuver is verified for the maneuver_verified predicate and ✗ indicates that the maneuver is not rule-compliant.

Algorithm 3: Encounter Action Verification.

```
Input: stand-on action $a_\mathrm{keep}$, turning to right actions $\mathcal{A}_{tr}$, turning to left actions $\mathcal{A}_{tl}$, accelerating actions $\mathcal{A}_\mathrm{acc}$, current state of obstacle vessel $\mathbf{s}_\mathrm{obs}$, current state of ego vessel $\mathbf{s}_\mathrm{ego}$, encounter predicate $\psi_e$
Output: set of safe actions $\mathcal{A}_s$, verified part of search tree $\mathcal{G}$
$\mathcal{A}_s \leftarrow \emptyset, \mathcal{G} \leftarrow \emptyset$
if $\psi_e = \mathrm{keep}$ then
    $\mathcal{A}_s \leftarrow \{a_\mathrm{keep}\}$
else
    if $\psi_e = \mathrm{head\_on} \vee \psi_e = \mathrm{crossing}$ then
        $\mathcal{A}_\mathrm{temp} \leftarrow \mathcal{A}_{tr}$
    else
        $\mathcal{A}_\mathrm{temp} \leftarrow \mathtt{get\_turning\_act}(\mathbf{s}_\mathrm{ego}, \mathbf{s}_\mathrm{obs}, \mathcal{A}_{tr}, \mathcal{A}_{tl})$
    end if
    for $a \in \mathcal{A}_\mathrm{temp}$ do
        $\mathcal{G}_\mathrm{temp} \leftarrow \mathtt{build\_st}(\mathbf{s}_\mathrm{ego}, \mathbf{s}_\mathrm{obs}, a, \mathcal{A}_\mathrm{acc}, t_\mathrm{m}, t_\mathrm{max,m})$
        if $\mathcal{G}_\mathrm{temp} \neq \emptyset$ then
            $\mathcal{G} \leftarrow \mathcal{G}_\mathrm{temp}, \mathcal{A}_s \leftarrow a$
        end if
    end for
end if
return $\mathcal{A}_s, \mathcal{G}$
```

c) Actions for encounter maneuvers: Algorithm 3 summarizes the action verification to achieve rule-compliant maneuvers for rules R_3 - R_6 given the statechart Γ is in an encounter state (i.e., $\exists i \in \{1, \dots, 4\} : \mathtt{in}(\rho_i)$). We denote the search tree generation with `build_st` (see Algorithm 2) and the detection of actions in the correct turning direction for overtake situations is abbreviated by the function `get_turning_act`. The result of Algorithm 3 is the safe action set $\mathcal{A}_s$ and the verified part of the search tree $\mathcal{G}$.

In an encounter situation, in which the ego vessel has to give way, a maneuver of the verified part of the search tree $\mathcal{G}$ is performed until there is no collision risk with respect to the obstacle vessel. In particular, the actions are conducted for at least the maneuver segment time t_m. At the end of a maneuver segment, the encounter situation is either resolved, or the action selection is constrained to the children of the selected search tree node. If $\mathcal{G}$ is an empty set, the ego vessel is a stand-on vessel and the only selectable action is a_keep.

C. Safe-By-Design Action Selection

We utilize a discrete action space for RL since this realizes efficient online safety verification and makes the encounter action verification feasible. In particular, we define an action set $\mathcal{A}$ of 49 discrete actions. One action represents the emergency action a_{em} and the others result from the combination of turning rates and accelerations:

$$\mathcal{A} = \{a_{\text{em}}, \mathcal{A}_{\text{regular}}\} \quad \text{where}$$
$$\mathcal{A}_{\text{regular}} = \{\text{a} \times \omega \,|\, \text{a} \in \mathcal{A}_{\text{a}}, \omega \in \mathcal{A}_{\omega}\}, \tag{9}$$

where $\mathcal{A}_{\text{a}}$ is the finite set describing the allowed normal accelerations and $\mathcal{A}_{\omega}$ is the finite set describing the allowed turning rates.

In the previous sections, we derived the verification of rule-compliant actions. By constraining the RL agent to these rule-compliant actions, we ensure by design that only safe actions are executed, and consequently only safe trajectories are performed. Theorem 2 states the solution to our problem statement in (5).

Theorem 2: Legal safety specified by $\langle \Phi, \leq \rangle$ can be ensured through constraining the action space of the RL agent to $\mathcal{A}_s(\hat{\rho})$ since all actions in $\mathcal{A}_s(\hat{\rho})$ are specification-compliant actions.

Proof: To prove this statement, we derive the safe action set $\mathcal{A}_s$ for all states of the statechart Γ.

(I) Initial state ρ_0*:* Since no rules apply in this state as proven in Theorem 1, any action is compliant with the specification and $\mathcal{A}_s(\rho_0) = \mathcal{A}_{\text{regular}}$.

(II) Emergency state ρ_5*:* We constrain the actions of the RL agent to the emergency action a_{em} returned by Algorithm 1, i.e., $\mathcal{A}_s(\rho_5) = a_{\text{em}}$.

(III) Encounter states $\rho_1 - \rho_4$*:* Based on Theorem 1 the maneuver predicates for the respective encounter situations must hold in these states to comply with the specification. Algorithm 3 returns the synthesized rule-compliant maneuvers and respective actions $\mathcal{A}_s(\rho_i)$ where $i \in \{1, \ldots, 4\}$.

Given $\mathcal{A}_s(\rho)$, we can constrain the action selection of the RL agent to $\mathcal{A}_s(\rho)$ with standard action masking [8] to obtain the safe policy π_s. Since the safe policy π_s only allows rule-compliant actions from $\mathcal{A}_s$, the trajectories ζ_{π_s} are compliant with the legal safety specification $\langle \Phi, \leq \rangle$. ■

VI. Reinforcement Learning

For the task of autonomous vessel navigation on the open sea, we design a simulation environment based on CommonOcean benchmarks [64] and the yaw-constrained dynamics in (1). The CommonOcean benchmarks contain a planning problem which specifies the goal area and initial state of the ego vessel as well as a scenario which specifies the traffic situation, i.e., for this study the trajectory of the obstacle vessel and the navigational area. At the start of an episode, a CommonOcean benchmark is randomly selected from the training set and the agent is provided with the initial observation. Based on the observation, the agent selects an action from the action set and receives the corresponding reward and next observation of the environment (see Fig. 1). If the safety verification is activated, the agent can only select from the verified safe action set $\mathcal{A}_s$ as derived in Section V. We regard a setting with finite time horizon episodes and terminate the episode in specified situations (see Section VI-A). The observation space, termination conditions, action space, action selection constraints, and reward function are detailed in the following paragraphs.

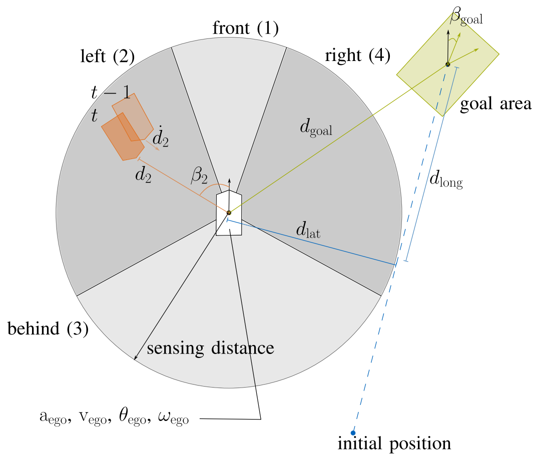


Fig. 7. Illustration of observations with sensing range and four sectors in gray, goal region in green, initial position with direct path to goal region in blue, and obstacle vessel for the previous time step $t-1$ and the current time step t in orange.

A. Observation Space and Termination

The observation space has 27 dimensions. We specify four types of observations: ego vessel observations, goal observations, surrounding traffic observations, and termination observations. Fig. 7 visualizes the ego vessel observations, goal observations, and surrounding traffic observations for the time step t.

The four ego vessel observations are the velocity v_{ego} and orientation θ_{ego} of the ego vessel state s_{ego}, the acceleration a_{ego}, and turning rate ω_{ego} corresponding to the ego vessel control input. The five continuous goal observations are the Euclidean distance to the goal d_{goal}, the remaining time steps until the maximal time step of the episode k_{max}, the orientation difference to the goal orientation range β_{goal}, and the longitudinal d_{long} and lateral d_{lat} position with respect to the line from the initial state to the center of the goal state. The observations d_{long} and d_{lat} are relevant since they indicate the deviation of the ego vessel from the optimal path when no other vessels need to be avoided. Additionally, we provide one Boolean goal observation that evaluates to true whenever $\min(|d_{\text{lat}}|, |d_{\text{long}}|)$ is larger than the distance d_{hull}, i.e., the ego vessel is far away from the path between the initial state and goal area.

The surrounding traffic observations are the distance d_j, angle β_j and distance rate $\dot{d}_j$ for the detected vessel in the sector $j \in \{1, \ldots, J\}$, where J is the number of sectors. The vessels are only detected if the Euclidean distance to the ego vessel is at most the sensing distance d_{sense}. For this study, we align the sectors with the sectors specified for the COLREGS collision avoidance rules. Thus, we obtain the four sectors front, left, right, and behind and twelve observation variables, as depicted in Fig. 7.

The five termination observations are Boolean observations and indicate if

- the maximal time step was reached $\mathbb{1}_{\text{time}} = 1$,
- the vessel is outside of the navigational area $\mathbb{1}_{\text{area}} = 1$,
- the vessel velocity is zero $\mathbb{1}_{\text{stopped}} = 1$,
- the vessel collided $\mathbb{1}_{\text{collision}} = 1$,
- the vessel reached the goal area $\mathbb{1}_{\text{goal}} = 1$.

We terminate the episode when the ego vessel stopped, as reverse driving is not meaningful on the open sea and the termination leads to the agent being reset to a much more meaningful initial state of another CommonOcean benchmark. The termination conditions follow directly form the termination observations, as we terminate the episode if one of these observations is present.

B. Reward

The reward is designed such that the vessel is reinforced in goal reaching behavior and penalized for unsafe or inefficient behavior. In particular, we design a reward function based on sparse and dense components. The sparse rewards are related to termination conditions and using the emergency planner:

$$\begin{aligned} r_{\text{sparse}} = \; & c_{\text{time}} \mathbb{1}_{\text{time}} + c_{\text{area}} \mathbb{1}_{\text{area}} + c_{\text{goal}} \mathbb{1}_{\text{goal}} \\ & + c_{\text{stopped}} \mathbb{1}_{\text{stopped}} + c_{\text{collision}} \mathbb{1}_{\text{collision}} \\ & + c_{\text{emergency}} \mathbb{1}_{\text{emergency}}, \end{aligned}$$

where c_{i} indicate the reward coefficients, which are all negative except for c_{goal}.

Additionally, we define four types of dense rewards for COLREGS compliance, advancing to the goal, keeping the velocity, and deviation from the path between initial state and goal. To incentivise behavior that is compliant with the collision avoidance rules specified in the COLREGS, we utilize a reward component specified in [41, (26)]:

$$r_{\text{colregs}} = -\frac{\alpha}{1 + \exp(\gamma_{\phi,\text{dyn}} |\phi|)} \exp((\zeta_v v_{\text{obs},\phi} - \zeta_{\text{obs,d}}) d_{\text{obs}}).$$

The angle $\phi \in [-\pi, \pi]$ specifies the relative angle between the ego orientation and the orientation toward the obstacle vessel, $v_{\text{obs},\phi}$ specifies the velocity component of the obstacle vessel velocity in the radial direction from the ego vessel to the obstacle vessel, and d_{obs} is the distance observed to the obstacle vessel, i.e., the respective d_j. The parameters $\alpha, \gamma_{\phi,\text{dyn}}, \zeta_v$, and $\zeta_{\text{obs,d}}$ are set to the same values as defined in [41].

Further, we define a reward component that supports the agent in learning how to reach the goal by providing a reward that is proportional to the advance or retreat from the goal since the previous time step:

$$r_{\text{goal}} = c_{\text{reach}} \left(\| \mathbf{p}_{\text{ego},t} - \mathbf{p}_{\text{goal}} \|_2 - \| \mathbf{p}_{\text{ego},t-1} - \mathbf{p}_{\text{goal}} \|_2 \right).$$

The center position of the goal area is $\mathbf{p}_{\text{goal}}$, and $\mathbf{p}_{\text{ego},t}$ is the current ego position, $\mathbf{p}_{\text{ego},t-1}$ is the ego vessel position at the previous time step, and c_{reach} is a scaling coefficient.

On the open sea, vessels typically navigate in a narrow speed range. To enforce this also for the RL agent, the reward component r_{velocity} provides a penalty proportional to the deviation from the desired speed range:

$$r_{\text{velocity}} = \begin{cases} c_v(v_{\text{ego}} - v_{\text{high}}) & \text{if } v_{\text{ego}} > v_{\text{high}} \\ c_v(v_{\text{low}} - v_{\text{ego}}) & \text{if } v_{\text{ego}} < v_{\text{low}} \\ 0 & \text{otherwise.} \end{cases}$$

The parameters v_{low} and v_{high} define the speed range bounds, and c_v is the reward coefficient.

The last reward component informs the agent about its deviation from the direct path between the initial state and the goal area:

$$r_{\text{deviate}} = c_{\text{deviate}} \min(|d_{\text{lat}}|, d_{\text{hull}}),$$

where the coefficient c_{deviate} scales the penalty proportional to the absolute lateral deviation $|d_{\text{lat}}|$, and $c_{\text{deviate}} d_{\text{hull}}$ is the maximum of the reward component r_{deviate}. Finally, the reward function is given by the sum of all components:

$$r = r_{\text{sparse}} + r_{\text{colregs}} + r_{\text{goal}} + r_{\text{velocity}} + r_{\text{deviate}}. \tag{10}$$

VII. Numerical Experiments

Critical encounter situations are rare in maritime traffic data. Thus, this data is not well suited for training RL agents that should learn how to handle encounter situations. Therefore, we construct random CommonOcean benchmarks [64] that represent critical encounters as a foundation of our simulation environment. In particular, we initialize the ego vessel and the other vessel approximately 2000 m–3500 m away from their closest encounter position. The initial velocity range for both vessels is $[3\,\text{m/s}, 7\,\text{m/s}]$. For the obstacle vessel, we generate a trajectory that is close to constant velocity and speed, and disturb the initial orientation and velocity with values sampled uniformly from $[-0.05\,\text{rad}, 0.05\,\text{rad}]$ and $[-0.1\,\text{m/s}, 0.1\,\text{m/s}]$, respectively, to make the trajectory more realistic. The goal area is approximately 4500 m away from the initial position of the ego vessel. The goal area is 400 m long and 60 m wide. The time horizon for the scenario is $k_{\max} = 170$ time steps where the time step size is $\Delta t = 10\,\text{s}$. In total, we constructed 2000 CommonOcean benchmarks [64] and randomly split them in a 70 % training and 30 % testing set. The model of the ego vessel is the yaw-constrained model in (1) and we use the parameters of a container vessel.[2] We reduce the maximum velocity specified in the vessel parameters to 9.5 m/s to better match a realistic velocity range for open sea maneuvering.

Next to the simulation environment, we need to specify values for the parameters of the safety verification approach, ego vessel, and reinforcement learning. Table II summarizes the parameters. Note that the emergency controller can use the full control input space specified for the ego vessel through the intervals $[-\text{a}_{\max}, \text{a}_{\max}]$ and $[-\omega_{\max}, \omega_{\max}]$. For normal operation, we reduce the control input limits to a more reasonable range for open

[2] The container vessel is the vessel type 1 from https://commonocean.cps.cit.tum.de/commonocean-models.

TABLE II
EXPERIMENTAL PARAMETERS

Parameter	**Value**	**Parameter**	**Value**
Safety verification			
Δ_{ahead}	45 deg	Δ_{stern}	20 deg
$v_{\text{pm,max}}$	$10\,\text{m}\,\text{s}^{-1}$	$\mathrm{a}_{\text{pm,max}}$	$0.045\,\text{m}\,\text{s}^{-2}$
d_{resolved}	$2\ l_{\text{ego}}$	$\mathrm{a}_{\text{stern}}$	$0.2\ \mathrm{a}_{\text{max}}$
$d_{\text{obs,safety}}$	$2\ l_{\text{obs}}$	$d_{\text{min,ahead}}$	$3\ l_{\text{obs}}$
$\Delta_{\text{head-on}}$	5 deg	$t_{\text{horizon}}^{\text{check}}$	420 s
$\Delta_{\text{no\_turn}}$	10 deg	t_{maneuver}	70 s
$\Delta_{\text{large\_turn}}$	20 deg	t_{react}	60 s
t_{pred}	180 s	t_{m}	40 s
$t_{\text{max,m}}$	200 s		
Ego vessel			
l_{ego}	175 m	ω_{max}	$0.03\,\text{rad}\,\text{s}^{-1}$
a_{max}	$0.24\,\text{m}\,\text{s}^{-2}$	v_{max}	$9.5\,\text{m}\,\text{s}^{-1}$
Reinforcement learning			
v_{low}	$2.5\,\text{m}\,\text{s}^{-1}$	v_{high}	$8\,\text{m}\,\text{s}^{-1}$
c_{time}	-25	c_{area}	-5
c_{goal}	50	c_{stopped}	-40
$c_{\text{collision}}$	-50	$c_{\text{emergency}}$	-0.5
c_{reach}	1.5	c_v	-2
c_{deviate}	-0.001	d_{sense}	8000 m
d_{hull}	2000 m	J	4
$\mathcal{A}_{\text{a}} = \{-0.048, -0.032, -0.016, 0, 0.016, 0.032, 0.048\}\text{m}\,\text{s}^{-2}$			
$\mathcal{A}_{\omega} = \{-0.018, -0.012, -0.06, 0, 0.06, 0.012, 0.018\}\text{rad}\,\text{s}^{-1}$			

sea maneuvering. This is reflected by the sets of allowable accelerations $\mathcal{A}_{\text{a}}$ and turning rates $\mathcal{A}_{\omega}$ (see Table II). As model-free RL algorithm, we used proximal policy optimization (PPO) [65]. Our implementation is based on stable-baselines3 [66] and the action masking implementation in [8]. The agent networks are multi-layer perceptron networks with two layers and 64 neurons in each layer.

A. Evaluation Concept

To comprehensively evaluate our approach, we introduce two benchmark agents next to our provably safe agent and compare different deployment setups. We train all three agents in our simulation environment, which is based on the training data of critical CommonOcean benchmarks [64]. The trained agents are:

1) the *baseline* agent with the reward function $r = r_{\text{sparse}} + r_{\text{goal}} + r_{\text{velocity}} + r_{\text{deviate}}$, i.e., $r_{\text{colregs}} = 0$ in (10), and no safety verification,
2) the *rule-reward* agent, which is informed by the COLREGS reward r_{colregs}, i.e., reward function (10), and
3) the *safe* agent with safety verification and reward function (10).

The baseline agent represents a straightforward RL implementation for which the agent is informed about unsafe actions only sparsely with a collision penalty. The rule-reward agent models the state-of-the-art for traffic-rule-informed open-sea vessel navigation [6], [7], [41], [42], because the reward function includes a COLREGS reward r_{colregs}. For each agent type, we use ten random seeds and train an agent per seed for three million environment steps.

We evaluate the deployment performance of the trained agents on the testing set of the handcrafted critical scenarios and on scenarios from recorded traffic data.[3] For the rule-reward and baseline agent, we investigate performance without, i.e., as trained, and with safety verification enabled. Including the safety verification after training allows us to evaluate if guaranteeing traffic rule compliance after training is sufficient. Note that the action space of the two benchmark agents is $\mathcal{A}_{\text{regular}}$, except for deployment with safety verification.

We consider critical scenarios from recorded traffic data to examine the generalization of the agents to real-world situations. To this end, we use marine traffic data from three large open-sea areas off the US coast from [15] and extract critical encounters. In particular, we only use scenarios where the distance between two vessels drops to 5000 m or lower. Further, we ensure that the paths of both vessels cross each other. Then, we replace one vessel by an ego vessel to generate the initial state and goal area. The initial state is part of the recorded trajectory and is selected about 2000 m before the closest encounter. The position of the goal area is also part of the recorded trajectory and is about 2000 m after the closest encounter. We use the same shape for the goal area as in our handcrafted scenarios. In total, we identify 49 critical scenarios in the three large open-sea areas off the US coast from traffic data of January 2019 (about 30 GB of raw Automatic Identification System (AIS) data).

We evaluate our agents based on the goal-reaching rate, reward, episode lengths, collisions, emergency controller usage and rule violations. Rule violations reflect how often per episode the regular collision avoidance rules are violated. For that, we count:

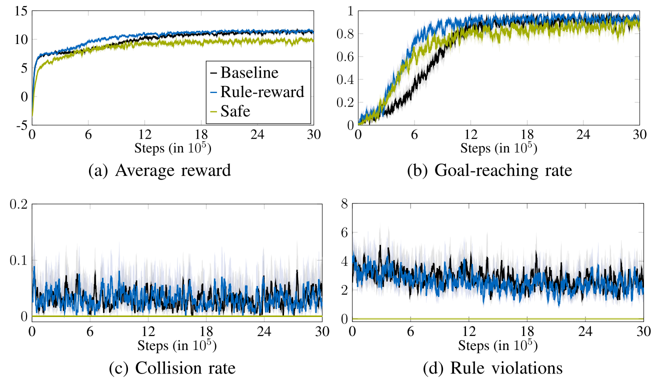
- every time step of violating the stand-on vessel position results;
- every crossing, overtaking and head-on encounter for which no proper collision avoidance maneuver is taken.

B. Results

a) Training evaluation: Fig. 8 shows the training curves for the three agent types. The average reward curves show similar convergence across agent types, although the baseline and rule-reward agents achieve slightly higher rewards after three million training steps. Note that for the displayed reward curves, the emergency penalty and COLREGS reward term r_{colregs} are subtracted for comparability. The goal-reaching rate curves mirror the reward curves and the agents reach goals in about 90 % of all scenarios at the end of the training. We observe that the agent types without safety verification reach the goal slightly more often.

Importantly, there are no collisions and rule violations for the safe agent (see Fig. 8(c) and (d)). For the baseline and rule-reward agent, the collision rate is relatively stable around 5 %

[3] All scenarios are publicly on the https://commonocean.cps.cit.tum.de with benchmark identifiers ZAM_AAA-1_20240121_T-[0,. . .,1999].



(a) Average reward

(b) Goal-reaching rate

(c) Collision rate

(d) Rule violations

Fig. 8. Mean and bootstrapped 95 % confidence interval for training curves for baseline, rule-reward, and safe agents averaged over ten random seeds.

TABLE III
Testing Results on 600 Handcrafted and 49 Recorded Scenarios

Setup		Efficiency		Safety		
Agent	Verify	Goal-reach	Ep. length	Collided	Rules violated	Emerg. steps
Handcrafted testing scenarios						
Base	✗	86.8 %	566 s	3.13 %	2.65	–
RR	✗	**90.7** %	**544** s	2.85 %	2.24	–
Base	✓	44.0 %	678 s	**0.0** %	**0**	10.06 %
RR	✓	47.6 %	702 s	**0.0** %	**0**	9.96 %
Safe	✓	86.3 %	647 s	**0.0** %	**0**	**6.18** %
Recorded maritime traffic scenarios						
Base	✗	83.1 %	563 s	0.41 %	0.75	–
RR	✗	**84.7** %	**550** s	0.41 %	0.82	–
Base	✓	78.3 %	591 s	**0.0** %	**0**	2.35 %
RR	✓	82.4 %	565 s	**0.0** %	**0**	**1.84** %
Safe	✓	78.2 %	630 s	**0.0** %	**0**	2.98 %

Note: The rule-reward and baseline agents are abbreviated with RR and base. Ep. length is the average episode time horizon. Emerg. steps denote the percentage of steps for which the emergency controller intervened.

during the full training time. Rule violations for the baseline and rule-reward agent slightly decrease but never reach zero. This suggests that complying with the COLREGS effectively achieves collision avoidance.

b) Deployment evaluation: The results averaged over ten random seeds for each agent type are summarized in Table III. For the handcrafted scenarios, the rule-reward agents reach the goal for 90.7 % of the scenarios. This is about 5 % higher than for the baseline and safe agent. Yet, only the safe agent achieves zero collisions and no rule violations. The rule-reward agent collides and violates the rules fewer times than the baseline agent. If the safety verification is enabled for the baseline and rule-reward agent, the goal-reaching rate drops significantly by approximately 40 %. Additionally, for the safe agent, the emergency controller intervenes on average in 6 % of the time steps in an episode, whereas for the rule-reward and baseline agents with activated safety verification, the emergency controller is needed in approximately 10 % of the time steps in an episode.

Table III displays the testing results on the 49 critical recorded traffic scenarios for the different agent types. The rule-reward agent reaches the goal most often and exhibits the lowest average episode length. Interestingly, the goal-reaching rate for the baseline and rule-reward agent drops only by about 5 % when activating our safety verification approach. The collision rate and rule violation rate are smaller than for the handcrafted scenarios. With activated safety verification, we observe no collisions and no rule violations. Note that the differences in the reported means for goal-reaching rate and emergency steps between the agents with activated safety verification are statistically insignificant.[4]

C. Discussion

a) Safety in handcrafted scenarios: The safety verification ensures that the encounter traffic rules are never violated and we empirically observe that no collisions occur. However, this results in a lower goal-reaching rate than for the soft-constrained rule-reward agent. One reason for this observation might be that with safety verification, the task is more difficult to solve since the agent is often constrained to avoidance maneuvers before it can maneuver freely again. Thus, the safe agent can explore less freely compared to the baseline and rule-reward agents. The drop in the goal-reaching rate when the safety verification is enabled after training is likely due to the distribution shift, as the baseline and rule-reward agents are probable led to states that they explored less frequently or not at all during training.

b) Safety on recorded scenarios: In contrast, testing the rule-reward and baseline agent with safety verification on the scenarios from recorded traffic data does not lead to such a significant drop. At the same time, the agent setups without safety verification exhibit fewer rule violations and fewer collisions on the recorded maritime traffic scenarios. Both observations indicate that the scenarios based on recorded data are less critical than the handcrafted situations and, thus, easier to solve for the agents that were not constrained to rule-compliant actions during training. Generally, the agents generalize well to the scenarios based on recorded data. Since identifying critical situations in recorded maritime traffic data is computation-heavy and critical situations are very rare, this small gap between realistic recorded and randomly handcrafted situations is compensated by being able to create many scenarios: The 49 critical situations resulted from one month of maritime traffic data at the coast of the US, whereas the 2000 handcrafted critical situations were generated in a matter of minutes. Yet, recorded scenarios are not fully representing the variety of the real world. Thus, future work should investigate if our safe agent also performs well on a real-world test bed.

c) Requirements for multi-vessel traffic situations: Real-world traffic situations can include more than two vessel on a collision course. Our formalized traffic rules can be evaluated for these more complex traffic situations as demonstrated in [15]. Yet, the current version of the COLREGS does not provide a clear collision avoidance specification if more than two vessels are involved. Thus, a formal verification cannot be developed due to the lack of a clear specification. Future work should investigate

[4] The p-values for paired t-tests between the safe agent and the rule-reward and baseline agents for the goal-reaching rate are 0.248 and 0.951, respectively. The p-values for agents with respect to emergency steps are 0.211 and 0.381.

extensions of the COLREGS to fill this specification gap and consequently realize provably rule-compliant motion planning in multi-vessel traffic situations.

d) Action space choice: The discrete action space makes it possible to efficiently identify rule-compliant actions. However, a continuous action space would allow the agent to explore all possible actions. This significantly increases the challenge of identifying safe actions, because there are infinitely many individual continuous actions in a continuous action space. Yet, one approach to investigate in future work could be obtaining rule-compliant state sets as proposed in [67] and correcting actions proposed by the agent to safe actions, e.g., with action projection as in [68].

e) Satisfiablity of rules: The parametrization of the temporal logic rules eases re-adjusting to regulation changes. Yet, these parameters must be manually tuned to ensure that the temporal logic rules are satisfiable. For example, it is important that the detection of an encounter situation happens early enough so that no emergency situation is detected during a give-way maneuver. For instance, theorem provers could help to verify that the chosen rule parameters guarantee that the rules are satisfiable. However, formulating this proof is challenging due to the continuous state and action space, and subject to future work.

VIII. Conclusion

We are the first to propose a provably safe RL approach for autonomous power-driven vessels on the open sea that achieves provable compliance with traffic rules formalized with temporal logic. For that, we introduced an online verification approach based on our formalized rules identifying the set of safe actions. Our formal emergency detection and emergency controller achieves collision avoidance for the regarded traffic situations even if other vessels do not comply with traffic rules. In critical maritime traffic situations, our safe RL agent achieves rule compliance, in contrast to state-of-the art agents that are informed about safety only through the reward. At the same time, all agents achieve a satisfactory goal-reaching performance on critical traffic situations. Our evaluation on recorded traffic situations shows that our safe RL agent generalizes beyond the distribution of training data. This study is a first step toward learning-based motion planning systems complying with traffic rules for autonomous vessel navigation.

Appendix

A. Predicates specified [15]

In Table IV, we briefly recapitulate the predicates specified in [15]. We refer the interested reader to our previous work [15] for detailed explanations. Subsequently, the necessary notation that was not yet introduced in this article is introduced and the re-parametrization of the predicate collision_possible is explained.

The trajectory of vessel i consists of states at discrete time steps and is denoted as $\mathcal{T}_i$. The velocity vector based on the state of the vessel is $\mathbf{v}_i = \texttt{proj}_v(\mathbf{s}_i)\,\texttt{unit\_v}(\mathbf{s}_i)$. We define a clock $\texttt{cl}(\mathcal{T}_i, \mathbf{s}_i)$ that starts at the initial time step of a trajectory and returns the elapsed time for a state $\mathbf{s}_i$. Further, we require a function $\texttt{state}(\mathcal{T}_i, t_k)$ which returns the state of a trajectory at time t_k. The modulo operator $\mathrm{mod}(a, b)$ returns the remainder of a/b for $a, b \in \mathbb{R}$ using floored division. The function $\texttt{t}_s$ returns the time for a predicate trace where the respective predicates changed last from false to true. The collision cone CC' is based on the velocity obstacle concept [61] and the construction is detailed in [15, Fig. 1].

For this work, we made two re-parametrizations of the predicate collision_possible, which determines if two vessels l and m are on a collision course and, thus, could collide within the time t_{horizon}. First, we also want to detect a collision course if the vessels would pass each other with insufficient distance. Thus, we use $r_m = 3\,l_m$ for the collision cone CC' instead of $r_m = l_m$ in [15, Fig. 1]. This results in detecting a collision possibility if the vessels would not keep a safe distance of at least two lengths of the vessel m. Second, we evaluate the set of vessel velocities $\mathcal{V}_l$ with respect to their collision possibility instead of only the current velocity $\mathbf{v}_l$. In particular, we check the collision possibility for

$$\mathcal{V}_l = \{\lambda\,\texttt{unit\_v}(\mathbf{s}_l) | \lambda \in [\texttt{proj}_v(\mathbf{s}_l) - v_\epsilon, \texttt{proj}_v(\mathbf{s}_l) + v_\epsilon]\}.$$

We set the velocity difference v_ϵ to $1\,\mathrm{m/s}$ for our numerical evaluations.

B. Proof of Lemma 1

Proof: To prove that only one predicate of keep, crossing, head_on, and overtake can evaluate to true, we show for each combination that the conjunction is false when evaluated for two vessels l and m. For the combination of crossing and head_on, it directly follows that the predicates cannot be true at the same time from the relative position detected by the respective sector predicates.

(I) crossing ∧ head_on:

$$\begin{aligned} &\text{crossing}(\mathbf{s}_l, \mathbf{s}_m, \cdot) \wedge \text{head\_on}(\mathbf{s}_l, \mathbf{s}_m, \cdot)\\ =\ &(\text{in\_right\_sector}(\mathbf{s}_l, \mathbf{s}_m) \wedge \ldots) \wedge\\ &(\text{in\_front\_sector}(\mathbf{s}_l, \mathbf{s}_m) \wedge \ldots)\\ =\ &\bot \end{aligned}$$

For the combination of crossing and overtake, let us assume that crossing predicate is true. Then, the vessel m is oriented towards left and in the right sector of vessel l (see Fig. 3 and Fig. 4 in [15]). Thus, it is geometrically impossible for vessel l to be in the behind sector of vessel m and overtake cannot be true.

(II) crossing ∧ overtake:

$$\begin{aligned} &\text{crossing}(\mathbf{s}_l, \mathbf{s}_m, \cdot) \wedge \text{overtake}(\mathbf{s}_l, \mathbf{s}_m, \cdot)\\ =\ &(\text{in\_right\_sector}(\mathbf{s}_l, \mathbf{s}_m) \wedge\\ &\text{orientation\_towards\_left}(\mathbf{s}_l, \mathbf{s}_m, \Delta_{\text{head-on}}) \wedge \ldots) \wedge\\ &(\text{in\_behind\_sector}(\mathbf{s}_m, \mathbf{s}_l) \wedge \ldots)\\ =\ &\bot \end{aligned}$$

The predicates head_on and overtake cannot be true simultaneously as the relative positions and orientations contradict each other similar to case (II). In particular, if the vessel m is in the front sector of vessel l and their relative orientation is in $[\pi - \Delta_{\text{head-on}}, \pi + \Delta_{\text{head-on}}]$, then vessel l cannot be in the behind sector of vessel m.

(III) head_on $\wedge$ overtake::

$$\begin{aligned}&\text{head\_on}(\mathbf{s}_l, \mathbf{s}_m, \cdot) \wedge \text{overtake}(\mathbf{s}_l, \mathbf{s}_m, \cdot)\\ =\ &(\text{in\_front\_sector}(\mathbf{s}_l, \mathbf{s}_m)\wedge\\ &\neg\text{orientation\_delta}(\mathbf{s}_l, \mathbf{s}_m, \Delta_{\text{head-on}}, \pi) \wedge \ldots) \wedge\\ &(\text{in\_behind\_sector}(\mathbf{s}_m, \mathbf{s}_l) \wedge \ldots)\\ =\ &\bot\end{aligned}$$

The predicate keep is a disjunction of two cases in which the vessel has to keep its course and speed. Thus, we have to show that for both statements of the disjunction that they evaluate to false. The explanation for the equation steps are marked with small letters in round brackets, e.g., (a), and follow after the respective equations.

(IV) overtake $\wedge$ keep:

$$\begin{aligned}&\text{overtake}(\mathbf{s}_l, \mathbf{s}_m, \cdot) \wedge \text{keep}(\mathbf{s}_l, \mathbf{s}_m, \cdot)\\ \overset{(a)}{=}\ &(\text{overtake}(\mathbf{s}_l, \mathbf{s}_m, \cdot) \wedge (\text{in\_left\_sector}(\mathbf{s}_l, \mathbf{s}_m) \wedge \ldots)) \vee\\ &(\text{overtake}(\mathbf{s}_l, \mathbf{s}_m, \cdot) \wedge \text{overtake}(\mathbf{s}_m, \mathbf{s}_l, \cdot))\\ \overset{(b)}{=}\ &((\text{in\_behind\_sector}(\mathbf{s}_l, \mathbf{s}_m) \wedge \ldots) \wedge\\ &(\text{in\_left\_sector}(\mathbf{s}_l, \mathbf{s}_m) \wedge \ldots)) \vee\\ &(\text{overtake}(\mathbf{s}_l, \mathbf{s}_m, \cdot) \wedge \text{overtake}(\mathbf{s}_m, \mathbf{s}_l, \cdot))\\ \overset{(c)}{=}\ &\bot \vee \bot\\ =\ &\bot\end{aligned}$$

(a) We distribute the disjunction in keep over the conjunction with overtake.

(b) We insert the relevant parts of the predicates (see Table IV).

(c) For the first part of the disjunction, the vessels cannot be simultaneously in two sectors as in case (I). For the second part of the disjunction, the two overtake predicates cannot be true at the same time, as both vessels cannot overtake each other at the same time.

(V) crossing $\wedge$ keep:

$$\begin{aligned}&\text{crossing}(\mathbf{s}_l, \mathbf{s}_m, \cdot) \wedge \text{keep}(\mathbf{s}_l, \mathbf{s}_m, \cdot)\\ \overset{(a)}{=}\ &(\text{crossing}(\mathbf{s}_l, \mathbf{s}_m, \cdot) \wedge (\text{in\_left\_sector}(\mathbf{s}_l, \mathbf{s}_m) \wedge \ldots)) \vee\\ &(\text{crossing}(\mathbf{s}_l, \mathbf{s}_m, \cdot) \wedge \text{overtake}(\mathbf{s}_m, \mathbf{s}_l, \cdot))\\ \overset{(b)}{=}\ &((\text{in\_right\_sector}(\mathbf{s}_l, \mathbf{s}_m) \wedge \ldots) \wedge\\ &(\text{in\_left\_sector}(\mathbf{s}_l, \mathbf{s}_m) \wedge \ldots)) \vee\\ &((\text{in\_right\_sector}(\mathbf{s}_l, \mathbf{s}_m) \ \wedge \ldots) \wedge\\ &(\text{in\_behind\_sector}(\mathbf{s}_l, \mathbf{s}_m) \wedge \ldots))\\ \overset{(c)}{=}\ &\bot \vee \bot\\ =\ &\bot\end{aligned}$$

(a) We distribute the disjunction in keep over the conjunction with crossing.

(b) We insert the relevant parts of the predicates (see Table IV).

(c) For both parts of the disjunction, the vessels cannot be simultaneously in two sectors as in case (I).

(VI) head_on $\wedge$ keep::

$$\begin{aligned}&\text{head\_on}(\mathbf{s}_l, \mathbf{s}_m, \cdot) \wedge \text{keep}(\mathbf{s}_l, \mathbf{s}_m, \cdot)\\ \overset{(a)}{=}\ &(\text{head\_on}(\mathbf{s}_l, \mathbf{s}_m, \cdot) \wedge (\text{in\_left\_sector}(\mathbf{s}_l, \mathbf{s}_m) \wedge \ldots)) \vee\\ &(\text{head\_on}(\mathbf{s}_l, \mathbf{s}_m, \cdot) \wedge \text{overtake}(\mathbf{s}_m, \mathbf{s}_l, \cdot))\\ \overset{(b)}{=}\ &((\text{in\_front\_sector}(\mathbf{s}_l, \mathbf{s}_m) \wedge \ldots) \wedge\\ &(\text{in\_left\_sector}(\mathbf{s}_l, \mathbf{s}_m) \wedge \ldots)) \vee\\ &((\text{in\_front\_sector}(\mathbf{s}_l, \mathbf{s}_m) \ \wedge \ldots) \wedge\\ &(\text{in\_behind\_sector}(\mathbf{s}_l, \mathbf{s}_m) \wedge \ldots))\\ \overset{(c)}{=}\ &\bot \vee \bot\\ =\ &\bot\end{aligned}$$

(a) We distribute the disjunction in keep over the conjunction with head_on.

(b) We insert the relevant parts of the predicates (see Table IV).

(c) For both parts of the disjunction, the vessels cannot be simultaneously in two sectors as in case (I). ■

C. Position Tracking Controller

We design a Lyapunov controller to track a desired position $\mathbf{p}_{\text{des}}$ to realize the emergency maneuver control. This desired position is either the target position $\mathbf{p}_{\text{target}}$ to be reached or generated based on the current position $\mathbf{p}_t$ and the desired position so that the vessel approximately maintains the desired velocity v_{desired}. The Lyapunov function for turning rate V_ω and acceleration V_{a} are:

$$\begin{aligned}V_\omega &= 1 - ([\cos(\theta_t), \sin(\theta_t)]\mathbf{w}_{\text{des}}^T)^2,\\ V_{\text{a}} &= 0.5\,(\mathbf{p}_{\text{des}} - \mathbf{p}_t)(\mathbf{p}_{\text{des}} - \mathbf{p}_t)^T,\end{aligned}$$

with the desired orientation vector

$$\mathbf{w}_{\text{des}} = (\mathbf{p}_{\text{des}} - \mathbf{p}_t)/\| \mathbf{p}_{\text{des}} - \mathbf{p}_t \|_2.$$

With these Lyapunov functions, we obtain the control:

$$\begin{aligned}\omega &= -\lambda_1 \frac{V_\omega}{-2\,([-\sin(\theta), \cos(\theta)]\mathbf{w}_{\text{des}}^T)\,([\cos(\theta), \sin(\theta)]\mathbf{w}_{\text{des}}^T)},\\ \text{a} &= -\lambda_2 \frac{V_{\text{a}}}{-(\mathbf{p}_{\text{des}} - \mathbf{p}_t)[v_t\cos(\theta_t), v_t\sin(\theta_t)]^T}.\end{aligned}$$

If V_ω is larger than a threshold Δ_{V_ω}, then the acceleration control is set to zero so that the vessel only turns. For our numerical evaluations, we use the following parameter values: $v_{\text{desired}} = 6\,\text{m/s}$, $\lambda_1 = 4$, $\lambda_2 = 0.04$, and $\Delta_{V_\omega} = 0.3$.

TABLE IV
Predicates for Traffic Rule Specifications from [15] With Adaptions for This Work

Predicate	**Arguments**	**Definition**	**Detects ...**
Position and orientation predicates			
in_front_sector	$\mathbf{s}_l, \mathbf{s}_m$	in_sector($\mathbf{s}_l, \mathbf{s}_m, -\Delta_{\text{head-on}}, \Delta_{\text{head-on}}$)	relative position in front sector
in_left_sector	$\mathbf{s}_l, \mathbf{s}_m$	in_sector($\mathbf{s}_l, \mathbf{s}_m, -112.5^\circ, -\Delta_{\text{head-on}}$)	relative position in left sector
in_right_sector	$\mathbf{s}_l, \mathbf{s}_m$	in_sector($\mathbf{s}_l, \mathbf{s}_m, \Delta_{\text{head-on}}, 112.5^\circ$)	relative position in right sector
in_behind_sector	$\mathbf{s}_l, \mathbf{s}_m$	in_sector($\mathbf{s}_l, \mathbf{s}_m, 112.5^\circ, 247.5^\circ$)	relative position in behind sector
orientation_delta	$\mathbf{s}_l, \mathbf{s}_m,$ $\Delta_{\text{orient}}, c_{\text{o}}$	$\text{mod}(\text{proj}_\theta(\mathbf{s}_m) - \text{proj}_\theta(\mathbf{s}_l) + c_{\text{o}}, 2\pi)$ $\in [\Delta_{\text{orient}}, 2\pi - \Delta_{\text{orient}}]$	if relative orientation is in defined range
orientation_towards_right	$\mathbf{s}_l, \mathbf{s}_m,$ $\Delta_{\text{head-on}}$	$\text{mod}(\text{proj}_\theta(\mathbf{s}_m) - \text{proj}_\theta(\mathbf{s}_l), 2\pi)$ $\in [-\pi + \Delta_{\text{head-on}}, -\Delta_{\text{head-on}}]$	if relative orientation of vessel m is toward right
orientation_towards_left	$\mathbf{s}_l, \mathbf{s}_m,$ $\Delta_{\text{head-on}}$	$\text{mod}(\text{proj}_\theta(\mathbf{s}_m) - \text{proj}_\theta(\mathbf{s}_l), 2\pi)$ $\in [\Delta_{\text{head-on}}, \pi - \Delta_{\text{head-on}}]$	if relative orientation of vessel m is toward right
Velocity predicates			
drives_faster	$\mathbf{s}_l, \mathbf{s}_m$	$\mathtt{proj}_v(\mathbf{s}_l) > \mathtt{proj}_v(\mathbf{s}_m)$	if vessel l is faster than vessel m
safe_speed	$\mathbf{s}_l, v_{\max}$	$0 \leq \mathtt{proj}_v(\mathbf{s}_l) \leq v_{\max}$	safe speed of vessel l
General predicates			
collision_possible	$\mathbf{s}_l, \mathbf{s}_m, t_{\text{horizon}}$	$\mathcal{V}_l \in CC'(\mathbf{s}_l, \mathbf{s}_m) \wedge$ $\lVert \mathbf{v}_l - \mathbf{v}_m \rVert_2 \geq \lVert \mathtt{proj}_{\mathbf{p}}(\mathbf{s}_l) - \mathtt{proj}_{\mathbf{p}}(\mathbf{s}_m) \rVert_2 / t_{\text{horizon}}$	if vessels l and m are on a collision course
change_course	$\mathbf{s}_l, \mathcal{T}_l,$ $t_{\text{start}}, \Delta_{\text{course}}$	$\lvert \sum_{t_i = t_{\text{start}}}^{\mathtt{cl}(\mathcal{T}_l, \mathbf{s}_l)} \mathtt{proj}_\omega(\mathtt{state}(\mathcal{T}_l, t_i))\, \Delta t \rvert \geq \Delta_{\text{course}}$	if course has changed significant since t_{start}
turning_to_starbord	$\mathbf{s}_l, \mathcal{T}_l, t_{\text{start}}$	$\text{mod}\big(\mathtt{proj}_\theta(\mathtt{state}(\mathcal{T}_l, \mathtt{cl}(\mathcal{T}_l, \mathbf{s}_l))) -$ $\text{proj}_\theta(\mathtt{state}(\mathcal{T}_l, t_{\text{start}})), 2\pi\big) \in (\pi, 2\pi)$	if course has changed to starboard since t_{start}
overtake	$\mathbf{s}_l, \mathbf{s}_m, t_{\text{horizon}}^{\text{check}}$	collision_possible($\mathbf{s}_l, \mathbf{s}_m, t_{\text{horizon}}^{\text{check}}$) $\wedge$ in_behind_sector($\mathbf{s}_m, \mathbf{s}_l$) $\wedge$ drives_faster($\mathbf{s}_l, \mathbf{s}_m$) $\wedge$ $\neg$orientation_delta($\mathbf{s}_l, \mathbf{s}_m, 67.5^\circ, 0$)	give-way vessel of overtaking encounter situation
maneuver_overtake	$\mathbf{s}_l, \mathbf{s}_m,$ $\mathcal{T}_l, t_{\text{horizon}}^{\text{check}},$ $\Delta_{\text{large\_turn}}$	change_course($\mathbf{s}_l, \mathcal{T}_n, \mathtt{t}_s$(overtake), $\Delta_{\text{large\_turn}}$)	correct maneuver of give-way vessel in overtaking encounter situation
head_on	$\mathbf{s}_l, \mathbf{s}_m, t_{\text{horizon}}^{\text{check}},$ $\Delta_{\text{head-on}}$	collision_possible($\mathbf{s}_l, \mathbf{s}_m, t_{\text{horizon}}^{\text{check}}$) $\wedge$ in_front_sector($\mathbf{s}_l, \mathbf{s}_m$) $\wedge$ $\neg$orientation_delta($\mathbf{s}_l, \mathbf{s}_m, \Delta_{\text{head-on}}, \pi$)	give-way vessel of head-on encounter situation
maneuver_head_on	$\mathbf{s}_l, \mathbf{s}_m,$ $\mathcal{T}_l, t_{\text{horizon}}^{\text{check}},$ $\Delta_{\text{large\_turn}}, \Delta_{\text{head-on}}$	change_course($\mathbf{s}_l, \mathcal{T}_n, \mathtt{t}_s$(head_on), $\Delta_{\text{large\_turn}}$)$\wedge$ turning_to_starboard($\mathbf{s}_l, \mathcal{T}_n, \mathtt{t}_s$(head_on))	correct maneuver of give-way vessel in head-on encounter situation
crossing	$\mathbf{s}_l, \mathbf{s}_m, t_{\text{horizon}}^{\text{check}},$ $\Delta_{\text{head-on}}$	collision_possible($\mathbf{s}_l, \mathbf{s}_m, t_{\text{horizon}}^{\text{check}}$)$\wedge$ in_right_sector($\mathbf{s}_l, \mathbf{s}_m$)$\wedge$ orientation_towards_left($\mathbf{s}_l, \mathbf{s}_m, \Delta_{\text{head-on}}$)	give-way vessel of crossing encounter situation
maneuver_crossing	$\mathbf{s}_l, \mathbf{s}_m,$ $\mathcal{T}_l, t_{\text{horizon}}^{\text{check}},$ $\Delta_{\text{large\_turn}}, \Delta_{\text{head-on}}$	change_course($\mathbf{s}_l, \mathcal{T}_n, \mathtt{t}_s$(crossing), $\Delta_{\text{large\_turn}}$)$\wedge$ turning_to_starboard($\mathbf{s}_l, \mathcal{T}_n, \mathtt{t}_s$(crossing))	correct maneuver of give-way vessel in crossing encounter situation
keep	$\mathbf{s}_l, \mathbf{s}_m, t_{\text{horizon}}^{\text{check}},$ $\Delta_{\text{head-on}}$	(collision_possible($\mathbf{s}_l, \mathbf{s}_m, t_{\text{horizon}}^{\text{check}}$)$\wedge$ in_left_sector($\mathbf{s}_l, \mathbf{s}_m$)$\wedge$ orientation_towards_right($\mathbf{s}_l, \mathbf{s}_m, \Delta_{\text{head-on}}$))$\vee$ overtake($\mathbf{s}_m, \mathbf{s}_l, t_{\text{horizon}}^{\text{check}}$)	stand-on vessel
no_turning	$\mathbf{s}_l, \mathcal{T}_l, \Delta_{\text{no\_turn}}$	$\neg$change_course($x_n, \mathcal{T}_n, \mathtt{t}_s$(keep), $\Delta_{\text{no\_turn}}$)	correct stand-on maneuver

References

[1] B. R. Kiran et al., “Deep reinforcement learning for autonomous driving: A survey,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 4909–4926, Jun. 2022.

[2] F. Ye, S. Zhang, P. Wang, and C.-Y. Chan, “A survey of deep reinforcement learning algorithms for motion planning and control of autonomous vehicles,” in *Proc. IEEE Intell. Veh. Symp.*, 2021, pp. 1073–1080.

[3] M. El-Shamouty, X. Wu, S. Yang, M. Albus, and M. F. Huber, “Towards safe human-robot collaboration using deep reinforcement learning,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 4899–4905.

[4] D. Han, B. Mulyana, V. Stankovic, and S. Cheng, “A survey on deep reinforcement learning algorithms for robotic manipulation,” *Sensors*, vol. 23, no. 7, 2023, Art. no. 3762.

[5] P. Sarhadi, W. Naeem, and N. Athanasopoulos, “A survey of recent machine learning solutions for ship collision avoidance and mission planning,” *IFAC-PapersOnLine*, vol. 55, no. 31, pp. 257–268, 2022.

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

[7] X. Xu, P. Cai, Z. Ahmed, V. S. Yellapu, and W. Zhang, “Path planning and dynamic collision avoidance algorithm under COLREGs via deep reinforcement learning,” *Neurocomputing*, vol. 468, pp. 181–197, 2022.

[8] H. Krasowski, J. Thumm, M. Müller, L. Schäfer, X. Wang, and M. Althoff, “Provably safe reinforcement learning: Conceptual analysis, survey, and benchmarking,” *Trans. Mach. Learn. Res.*, 2023. [Online]. Available: https://openreview.net/forum?id=mcN0ezbnzO

[9] B. Vanholme, D. Gruyer, B. Lusetti, S. Glaser, and S. Mammar, “Highly automated driving on highways based on legal safety,” *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 1, pp. 333–347, Mar. 2013.

[10] N. Mehdipour, M. Althoff, R. D. Tebbens, and C. Belta, “Formal methods to comply with rules of the road in autonomous driving: State of the art and grand challenges,” *Automatica*, vol. 152, 2023, Art. no. 110692.

[11] C. E. Tuncali, G. Fainekos, H. Ito, and J. Kapinski, “Simulation-based adversarial test generation for autonomous vehicles with machine learning components,” in *Proc. IEEE Intell. Veh. Symp.*, 2018, pp. 1555–1562.

[12] C.-I. Vasile, J. Tumova, S. Karaman, C. Belta, and D. Rus, “Minimum-violation scLTL motion planning for mobility-on-demand,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 1481–1488.

[13] S. Maierhofer, A.-K. Rettinger, E. C. Mayer, and M. Althoff, “Formalization of interstate traffic rules in temporal logic,” in *Proc. IEEE Intell. Veh. Symp.*, 2020, pp. 752–759.

[14] K. Esterle, L. Gressenbuch, and A. Knoll, “Formalizing traffic rules for machine interpretability,” in *Proc. IEEE 3rd Connected Automated Veh. Symp.*, 2020, pp. 1–7.

[15] H. Krasowski and M. Althoff, “Temporal logic formalization of marine traffic rules,” in *Proc. IEEE Intell. Veh. Symp.*, 2021, pp. 186–192.

[16] X. Zhang, C. Wang, L. Jiang, L. An, and R. Yang, “Collision-avoidance navigation systems for maritime autonomous surface ships: A state of the art survey,” *Ocean Eng.*, vol. 235, 2021, Art. no. 109380.

[17] “COLREGs: Convention on the international regulations for preventing collisions at Sea,” International Maritime Organization (IMO), 1972.

[18] Y. Kuwata, M. T. Wolf, D. Zarzhitsky, and T. L. Huntsberger, “Safe maritime autonomous navigation with COLREGS, using velocity obstacles,” *IEEE J. Ocean. Eng.*, vol. 39, no. 1, pp. 110–119, Jan. 2014.

[19] L. Zhao and M. I. Roh, “COLREGs-compliant multiship collision avoidance based on deep reinforcement learning,” *Ocean Eng.*, vol. 191, pp. 106436–106450, 2019.

[20] S. Guo, X. Zhang, Y. Zheng, and Y. Du, “An autonomous path planning model for unmanned ships based on deep reinforcement learning,” *Sensors*, vol. 20, no. 2, 2020.

[21] X. Zhang, C. Wang, Y. Liu, and X. Chen, “Decision-making for the autonomous navigation of maritime autonomous surface ships based on scene division and deep reinforcement learning,” *Sensors*, vol. 19, no. 18, 2019.

[22] M. Junmin et al., “Mechanism of dynamic automatic collision avoidance and the optimal route in multi-ship encounter situations,” *J. Mar. Sci. Technol.*, vol. 26, pp. 141–158, 2021.

[23] Y. He, Y. Jin, L. Huang, Y. Xiong, P. Chen, and J. Mou, “Quantitative analysis of COLREG rules and seamanship for autonomous collision avoidance at open sea,” *Ocean Eng.*, vol. 140, pp. 281–291, 2017.

[24] H.-T. L. Chiang and L. Tapia, “COLREG-RRT: An RRT-based COLREGS-compliant motion planner for surface vehicle navigation,” *IEEE Robot. Automat. Lett.*, vol. 3, no. 3, pp. 2024–2031, Jul. 2018.

[25] M. R. Benjamin and J. A. Curcio, “COLREGS-based navigation of autonomous marine vehicles,” in *Proc. IEEE/OES Auton. Underwater Veh.*, 2004, pp. 32–39.

[26] T. A. Johansen, T. Perez, and A. Cristofaro, “Ship collision avoidance and COLREGS compliance using simulation-based control behavior selection with predictive hazard assessment,” *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 12, pp. 3407–3422, Dec. 2016.

[27] B. O. H. Eriksen, M. Breivik, E. F. Wilthil, A. L. Flåten, and E. F. Brekke, “The branching-course model predictive control algorithm for maritime collision avoidance,” *J. Field Robot.*, vol. 36, no. 7, pp. 1222–1249, 2019.

[28] D. K. M. Kufoalor, E. Wilthil, I. B. Hagen, E. F. Brekke, and T. A. Johansen, “Autonomous COLREGs-compliant decision making using maritime radar tracking and model predictive control,” in *Proc. 18th Eur. Control Conf.*, 2019, pp. 2536–2542.

[29] P. Stankiewicz and M. Kobilarov, “A primitive-based approach to good seamanship path planning for autonomous surface vessels,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 7767–7773.

[30] T. R. Torben, J. A. Glomsrud, T. A. Pedersen, I. B. Utne, and A. J. Sørensen, “Automatic simulation-based testing of autonomous ships using gaussian processes and temporal logic,” *Proc. Inst. Mech. Engineers, Part O: J. Risk Rel.*, vol. 237, no. 2, pp. 293–313, 2023.

[31] Y. Liu and R. Bucknall, “A survey of formation control and motion planning of multiple unmanned vehicles,” *Robotica*, vol. 36, no. 7, pp. 1019–1047, 2018.

[32] W. Wu, Z. Peng, L. Liu, and D. Wang, “A general safety-certified cooperative control architecture for interconnected intelligent surface vehicles with applications to vessel train,” *IEEE Trans. Intell. Veh.*, vol. 7, no. 3, pp. 627–637, Sep. 2022.

[33] W. Wu and S. Tong, “Collision-free finite-time adaptive fuzzy output-feedback formation control for unmanned surface vehicle systems,” *IEEE Trans. Intell. Veh.*, vol. 9, no. 1, pp. 1094–1103, Jan. 2024.

[34] Y. Zhao, Y. Ma, and S. Hu, “USV formation and path-following control via deep reinforcement learning with random braking,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 12, pp. 5468–5478, Dec. 2021.

[35] Y. Zhang, W. Wu, and W. Zhang, “Noncooperative game-based cooperative maneuvering of intelligent surface vehicles via accelerated learning-based neural predictors,” *IEEE Trans. Intell. Veh.*, vol. 8, no. 3, pp. 2212–2221, Mar. 2023.

[36] G. Wen, X. Fang, J. Zhou, and J. Zhou, “Robust formation tracking of multiple autonomous surface vessels with individual objectives: A noncooperative game-based approach,” *Control Eng. Pract.*, vol. 119, 2022.

[37] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley & Sons, Ltd, 2011.

[38] J. Zhang, H. Zhang, J. Liu, D. Wu, and C. G. Soares, “A two-stage path planning algorithm based on rapid-exploring random tree for ships navigating in multi-obstacle water areas considering COLREGs,” *J. Mar. Sci. Eng.*, vol. 10, no. 10, 2022, Art. no. 1441.

[39] T. T. Enevoldsen, C. Reinartz, and R. Galeazzi, “COLREGs-informed RRT* for collision avoidance of marine crafts,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 8083–8089.

[40] A. Tsolakis, D. Benders, O. De Groot, R. R. Negenborn, V. Reppa, and L. Ferranti, “COLREGs-aware trajectory optimization for autonomous surface vessels,” *IFAC-PapersOnLine*, vol. 55, no. 31, pp. 269–274, 2022.

[41] E. Meyer, A. Heiberg, A. Rasheed, and O. San, “COLREG-compliant collision avoidance for unmanned surface vehicle using deep reinforcement learning,” *IEEE Access*, vol. 8, pp. 165344–165364, 2020.

[42] D.-H. Chun, M.-I. Roh, H.-W. Lee, J. Ha, and D. Yu, “Deep reinforcement learning-based collision avoidance for an autonomous ship,” *Ocean Eng.*, vol. 234, 2021, Art. no. 109216.

[43] W. Xie, L. Gang, M. Zhang, T. Liu, and Z. Lan, “Optimizing multi-vessel collision avoidance decision making for autonomous surface vessels: A colregs-compliant deep reinforcement learning approach,” *J. Mar. Sci. Eng.*, vol. 12, no. 3, 2024.

[44] Y. Fan, Z. Sun, and G. Wang, “A novel intelligent collision avoidance algorithm based on deep reinforcement learning approach for USV,” *Ocean Eng.*, vol. 287, 2023, Art. no. 115649.

[45] N. Fulton and A. Platzer, “Safe reinforcement learning via formal methods: Toward safe control through proof and learning,” in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 6485–6492.

[46] N. Fulton and A. Platzer, “Verifiably safe off-model reinforcement learning,” in *Proc. Int. Conf. Tools Algorithms Construction Anal. Syst.*, 2019, pp. 413–430.

[47] B. Mirchevska, C. Pek, M. Werling, M. Althoff, and J. Boedecker, “High-level decision making for safe and reasonable autonomous lane changing using reinforcement learning,” in *Proc. IEEE 21st Int. Intell. Transp. Syst. Conf.*, 2018, pp. 2156–2162.

[48] H. Krasowski, X. Wang, and M. Althoff, “Safe reinforcement learning for autonomous lane changing using set-based prediction,” in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst.*, 2020, pp. 1–7.

[49] M. Brosowsky, F. Keck, J. Ketterer, S. Isele, D. Slieter, and M. Zöllner, “Safe deep reinforcement learning for adaptive cruise control by imposing state-specific safe sets,” in *Proc. IEEE Intell. Veh. Symp.*, 2021, pp. 488–495.

[50] H. Krasowski, Y. Zhang, and M. Althoff, “Safe reinforcement learning for urban driving using invariably safe braking sets,” in *Proc. IEEE 25th Int. Conf. Intell. Transp. Syst.*, 2022, pp. 2407–2414.

[51] D. Tabas and B. Zhang, “Computationally efficient safe reinforcement learning for power systems,” in *Proc. IEEE Amer. Control Conf.*, 2022, pp. 3303–3310.

[52] M. Alshiekh, R. Bloem, R. Ehlers, B. Könighofer, S. Niekum, and U. Topcu, “Safe reinforcement learning via shielding,” in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 2669–2678.

[53] B. Könighofer, F. Lorber, N. Jansen, and R. Bloem, “Shield synthesis for reinforcement learning,” in *Proc. 9th Int. Symp. Leveraging Appl. Formal Methods, Verification Validation: Verification Princ.*, 2020, pp. 290–306.

[54] X. Li, Z. Serlin, G. Yang, and C. Belta, “A formal methods approach to interpretable reinforcement learning for robotic planning,” *Sci. Robot.*, vol. 4, no. 37, 2019, Art. no. eaay6276.

[55] A. Censi et al., “Liability, ethics, and culture-aware behavior specification using rulebooks,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 8536–8542.

[56] M. Koschi and M. Althoff, “Set-based prediction of traffic participants considering occlusions and traffic rules,” *IEEE Trans. Intell. Veh.*, vol. 6, no. 2, pp. 249–265, Jun. 2021.

[57] H. Roehm, J. Oehlerking, M. Woehrle, and M. Althoff, “Model conformance for cyber-physical systems: A survey,” *ACM Trans. Cyber- Phys. Syst.*, vol. 3, no. 3, pp. 1–26, 2019.

[58] M. Althoff, “Reachability analysis and its application to the safety assessment of autonomous cars,” Dissertation, Technische Univ. München, Munich, Germany, 2010.

[59] M. Wetzlinger, N. Kochdumper, S. Bak, and M. Althoff, “Fully automated verification of linear systems using inner and outer approximations of reachable sets,” *IEEE Trans. Autom. Control*, vol. 68, no. 12, pp. 7771–7786, Dec. 2023.

[60] S. Magdici and M. Althoff, “Fail-safe motion planning of autonomous vehicles,” in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst.*, 2016, pp. 452–458.

[61] P. Fiorini and Z. Shiller, “Motion planning in dynamic environments using velocity obstacles,” *Int. J. Robot. Res.*, vol. 17, no. 7, pp. 760–772, 1998.

[62] T. Schouwenaars, J. How, and E. Feron, “Decentralized cooperative trajectory planning of multiple aircraft with hard safety guarantees,” in *Proc. AIAA Guid., Navigation, Control Conf. Exhibit*, 2004, Art. no. 5141.

[63] S. Bouraine, T. Fraichard, and H. Salhi, “Provably safe navigation for mobile robots with limited field-of-views in dynamic environments,” *Auton. Robots*, vol. 32, no. 3, pp. 267–283, 2012.

[64] H. Krasowski and M. Althoff, “CommonOcean: Composable benchmarks for motion planning on oceans,” in *Proc. IEEE 25th Int. Conf. Intell. Transp. Syst.*, 2022, pp. 1676–1682.

[65] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” 2017, arXiv:1707.06347.

[66] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, “Stable-baselines3: Reliable reinforcement learning implementations,” *J. Mach. Learn. Res.*, vol. 22, no. 268, pp. 1–8, 2021.

[67] E. Irani Liu and M. Althoff, “Specification-compliant driving corridors for motion planning of automated vehicles,” *IEEE Trans. Intell. Veh.*, vol. 8, no. 9, pp. 4180–4197, Sep. 2023.

[68] N. Kochdumper, H. Krasowski, X. Wang, S. Bak, and M. Althoff, “Provably safe reinforcement learning via action projection using reachability analysis and polynomial zonotopes,” *IEEE Open J. Control Syst.*, vol. 2, pp. 79–92, 2023.

Hanna Krasowski (Graduate Student Member, IEEE) received the B.Sc. degree in mechanical engineering from the Technical University of Darmstadt, Darmstadt, Germany, in 2017, and the M.Sc. degree in robotics, cognition, and intelligence in 2020 from the Technical University of Munich, Munich, Germany, where she is currently working toward the Ph.D. degree. Her research interests include provably safe reinforcement learning and motion planning for cyber-physical systems.

Matthias Althoff (Member, IEEE) received the Diploma Engineering degree in mechanical engineering and the Ph.D. degree in electrical engineering from the Technical University of Munich, Munich, Germany, in 2005 and 2010, respectively. From 2010 to 2012 he was a Postdoctoral Researcher with Carnegie Mellon University, Pittsburgh, PA, USA. From 2012 to 2013, he was an Assistant Professor with Technische Universität Ilmenau, Ilmenau, Germany. He is currently an Associate Professor of computer science with the Technical University of Munich. His research interests include formal verification of continuous and hybrid systems, reachability analysis, planning algorithms, nonlinear control, automated vehicles, and power systems.