Received 18 December 2023; revised 19 May 2024; accepted 2 July 2024. Date of publication 5 July 2024; date of current version 30 July 2024.

Digital Object Identifier 10.1109/OJITS.2024.3424587

Global and Local Awareness: Combine Reinforcement Learning and Model-Based Control for Collision Avoidance

LUMAN ZHAO (Member, IEEE), GUOYUAN LI (Senior Member, IEEE), AND HOUXIANG ZHANG (Senior Member, IEEE)

Department of Ocean Operations and Civil Engineering, Norwegian University of Science and Technology, 6009 Ålesund, Norway

CORRESPONDING AUTHOR: L. ZHAO (e-mail: Luman.Zhao@dnv.com)

This work was supported by the Research Council of Norway through the Knowledge-Building Project for Industry
"Digital Twins for Vessel Life Cycle Service" under Project 280703.

ABSTRACT In this research, we focus on developing an autonomous system for multiship collision avoidance. The proposed approach combines global path planning based on deep reinforcement learning (DRL) and local motion control to improve computational efficiency and alleviate the sensitivity to heading angle changes. To achieve this, firstly, DRL is used to learn a policy that maps observable states of target ships to a sequence of predicted waypoints. This learning task aims to generate a specific trajectory while avoiding collision with target ships complying with the international regulations for preventing collisions at sea (COLREGs). The learned policy is used as a global path planner during navigation. Secondly, the line-of-sight (LOS) guidance system is applied to calculate the desired course command based on the collision-free trajectory generated according to the policy. Lastly, a model-based control strategy is implemented to control the ship to the specific goal in collision-free space while satisfying the desired commands. We demonstrate the performance of the approach using an example of an autonomous surface vehicle. In comparison to other methods, our proposed control can provide a more stable and smoother maneuvering effect.

INDEX TERMS Control application of autonomous systems, deep reinforcement learning, multi-ship collision avoidance, online path following, the international regulations for preventing collisions at sea (COLREGs).

I. INTRODUCTION

AN AUTONOMOUS system for ships, i.e., the ability of an intelligent agent to move towards a specific goal smoothly without collision, has been attracting a significant amount of research attention recently. With the rapid development of technology, the world's first autonomous zero-emission container ship, Yara Birkeland has been delivered successfully and will be put into operation in 2022 [1]. The technology related to autonomous systems lies in the intersection of many fundamental research areas: motion prediction, path planning with collision avoidance, motion control, etc.

Collision avoidance for ships in an unpredictable environment is a challenging task. For example, a collision in congested waters can be catastrophic with increasing traffic densities and the average cruising speed. The outcome of such a collision may lead to a "pile-up" like on the motorway. It can be seen in the context of managing the risks that lack awareness of the other vessel, poor lookout, and insufficient assessment of situation account for 60%. Human error is the most common cause of maritime collisions [2]. In conclusion, the underlying human errors are a lack of experience and correct application of the International Regulations for the Prevention of Collisions

The review of this article was arranged by Associate Editor Abel C. H. Chen.

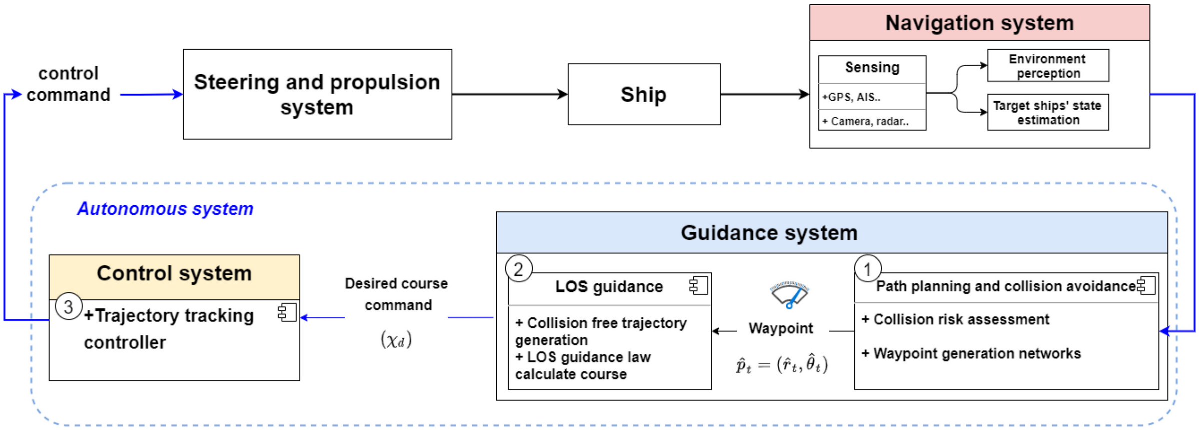

control
command
Steering and propulsion system
Ship
Navigation system
Sensing
+GPS, AIS..
+ Camera, radar..
Environment perception
Target ships' state estimation
Autonomous system
Guidance system
Control system
3
+Trajectory tracking controller
Desired course command
(χ_d)
2
LOS guidance
+ Collision free trajectory generation
+ LOS guidance law calculate course
Waypoint
$\hat{p}_t = (\hat{r}_t, \hat{\theta}_t)$
1
Path planning and collision avoidance
+ Collision risk assessment
+ Waypoint generation networks

FIGURE 1. The framework of the proposed autonomous system.

at Sea (COLREGs) [3]. The regulation is used to give guidance and help the crew to avoid collisions at sea. The officer on watch (OOW) in the busy waters is more easily misinterpreted or ignored. By eliminating human error from the equation, the development of autonomous system for collision avoidance is expected to enhance situation awareness, to improve collision risk prediction, and to automate the decision-making process.

Model-based methods are a popular paradigm for ship autonomous systems because they can leverage a given dynamics model to control it to follow a robust collision-free trajectory efficiently. According to a specific task for motion control for obstacle avoidance, environmental perception is first processed to build a global description. Meanwhile, the collision risks with target ships can be confirmed; Then, it generates collision-free trajectories comprising a sequence of waypoints. Finally, a local tracking controller is responsible for controlling the ship to the closest waypoint. This category of methods is usually used for a given encounter scenario. That is to say, the model-based methods usually consider specific dynamic models and environmental disturbances to conduct a more reliable result for a specific ship [4]. Accordingly, the challenges of model-based methods mainly come from the uncertainties of the model, limited knowledge of the environmental changes, and more complex encounter situations that lack description from COLREGs. A widely used way is to incorporate as many factors as possible into collision avoidance algorithms, such as COLREGs regulations, environmental disturbance, and the motion of target ships, to make the result more reliable and robust. The more involved factors are, the higher the computation power cost.

Data-driven methods such as deep reinforcement learning (DRL) applied to autonomous systems, from autonomous robots and vehicles to aircraft and ships. Numerous studies have demonstrated that utilizing DRL can effectively solve the problem of motion planning for ships. For example, the end-to-end DRL algorithms train an autonomous agent through a learning process that maps directly from the state estimation input to motion commands. Then the autonomous agent can successively make decisions depending on its current situation. The learned policy has been demonstrated to be potentially powerful and capable in an unknown environment through the trial-and-error training process. However, the policy in simulation brings up the problem of transferring such policies to the real world.

This article adopts a hybrid method to address the efficiency and feasibility needs for autonomous system development, combining global path planning based on DRL and local motion control. The framework of this system is proposed based on the GNC system [5], which consisting of the navigation system, the guidance system, and the control system (as illustrated in Fig. 1). The navigation system contains sensors that allow the ship to locate itself, sense its surrounding, and communicate with the incoming target ships. The guidance system is responsible for making decisions and generating collision-free trajectories to lead the ship to its target location based on the information from the navigation system. It is built by adopting a DRL model, which generates the desired course for the control system. The control system takes the desired signal as input and outputs relevant commands to the steering and propulsion system.

The learning process and control procedure are designated to achieve specific subgoals instead of the entire navigation stack. This combination emphasizes the benefit of deploying DRL in global planning, i.e., providing a collision-free trajectory with online planning capabilities. Furthermore, the involved model-based control strategy can provide an accurate control command to actuators, greatly offloading the learning process's computational load. It raises the potential to implement it in the real world.

Given the above, main contributions of this paper are summarized as follows:

- We introduce an innovative autonomous system architecture that integrates Deep Reinforcement Learning (DRL) for global path planning with model-based techniques for local motion control. This hybrid approach provides a comprehensive solution for autonomous maritime navigation.

- The proposed DRL-trained policy enables real-time computation of collision-free trajectories, ensuring dynamic adaptability and enhanced safety in complex maritime environments. This capability significantly reduces the risk of collisions by predicting and responding to potential hazards in real time.
- The computed trajectories not only avoid collisions but also strictly adhere to the COLREGs. Our method considers the movements of surrounding vessels, ensuring both legal compliance and operational safety in multi-ship scenarios.
- The proposed approach effectively manages multi-ship interactions in compliance with COLREGs within a simulation environment. We demonstrate that our system can handle complex navigational scenarios involving multiple vessels, efficiently coordinating their movements to prevent collisions while adhering to maritime regulations.

The rest of this article is organized as follows. In Section II, we provide a comparative review of work related to our method. In Section III, we cover preliminaries about the COLREGs regulations and ship modeling of the study. Section IV describes the proposed method of integrating global planning and control for the ship collision avoidance problem. To analyze the benefits of our method, we perform extensive experiments and comparative analysis with other techniques in Section V. Finally, Section VI concludes this paper.

II. RELATED WORKS

An extensive body of techniques has been working on developing autonomous systems for ships. In the scope of the GNC system, path planning and collision avoidance are two fundamental topics that are attacking research attention [4]. Reference [6] indicated that current studies adopted a hierarchical paradigm: global path planning combined with local motion control. It mainly focused on the robot's global path planning and motion control by comparing the learning-based methods with the classical methods. This well-categorized study can give a helpful guide in the field of marine vehicles. Compared with the global path planning and motion control approaches applied to aerial vehicle [7], researchers are more concentrated on solving the challenges from the COLREGs regulations, environmental disturbance, and validation issues. A wide variety of approaches have been proposed to overcome the aforementioned problems.

Model-Based Method: Autonomous system design has gravitated toward model-based methods because of their capability to deploy such systems in the real world reliably. The family of model predictive control (MPC) methods have been widely used for path planning and collision avoidance. For example, [8], [9] used the MPC method to calculate collision-free trajectories as the optimal solution by minimizing a cost function incorporate optimization into multi-ship collision avoidance. Reference [10] calculated the desired course and speed offsets by formulating the objective function, associating with the dynamic model of ships, COLREGs regulations, and environmental disturbance. These studies solved path planning and collision avoidance problems by formulating an objective function. If the conditions are used for long-range multi-ship maneuvering, the optimization process in MPC would be time-consuming for real-time control. The shortest computational latency is highly desirable for real-time implementation. Reference [11] integrated the dynamic model with a planning method, rapidly exploring random tree, to compute the collision-free trajectory, where comparisons with the MPC method provide further insight into the performance and capabilities of the approach. Reference [12] proposed a hybrid method, fast marching square and velocity obstacle (VO), for global path planning to generate the optimal trajectory. In conclusion, one of the challenges of the model-based methods is the convergence and computational complexity of numerical optimization. It will also increase the difficulty if the surrounding environment is a prior unknown.

End-to-End Learning-Based Method: To address the requirements to build an accurate dynamic model, and the problem of high computation cost with multiple target ships in a long-range encounter situation, and exploit in an unknown environment, recently, lots of machine learning techniques have already demonstrated remarkable potential promising results. Reference [6] highlighted the learning-based methods from different dimensions, including end-to-end learning and sub-task learning. For end-to-end learning, the system is directly trained holistically towards the final goal by the overall objective function at each learning step. An example of this end-to-end approach in this category is proposed by [13], which demonstrated a DRL-based algorithm could avoid both static and dynamic obstacles by combining two well-defined reward functions. In addition, DRL had been showing a great advantage in solving the end-to-end autopilot problems [14]. It can learn optimal steering policies in an unknown environment, that map observations directly to the ship's actions, thus enabling path following and collision avoidance during the navigation. For instance, [15] used a decentralized multi-agent reinforcement learning (MARL) framework, enabling autopilot vehicles to learn human-drive vehicle behavior for optimized social utility. Reference [16] showed that reinforcement learning could effectively organize control transitions in mixed autonomy systems, significantly reducing traffic disruption compared to traditional methods.

In some cases, these DRL policies learned from end-to-end strategy lack long-term planning capabilities based on possibly sparse and delayed rewards. Moreover, compared with the model-based method, the learning-based method typically cannot provide explicit assurance of safety [6].

Combining Model-Based Control and DRL: As a result, a combined approach had been proposed to improve learning efficiency and showed the potential to implement it in the real world. It can benefit from the advantages of model-based control and DRL in a way that addresses

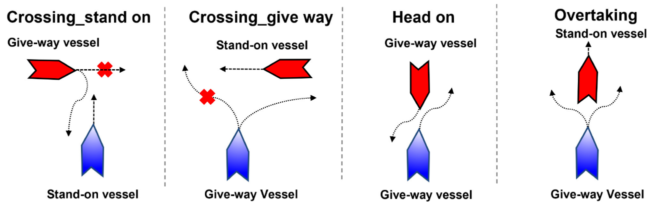


FIGURE 2. Illustration of the COLREGs interpretation.

their limitations [17]. Several papers take their effort to combine the learning with classical control methods for ground vehicle navigation and manipulation [18]. One of the hierarchical solutions, where the trained DRL policy is used for local control, sampling-based planning, such as RRT [19], Probabilistic Roadmaps [20] are used for global planning. These sampling-based algorithms are suitable for global planning tasks. For example, [20] combined DRL and sampling-based planner for long-range navigation tasks, where the sampling-based planner provided roadmaps, and DRL agent was used to controlling the robot under the direction of planning. Reference [19] used DRL to learn to propose a motion planning method that combining a sampling-based planner with DRL. In contrast, other works such as [21], used learning to predict waypoints that were used with maneuver control. The DRL policy generated the waypoints, and the model-based control planner generated collision-free control behavior. This study uses the DRL to predict the next waypoint, which is ensured to generate a collision-free trajectory comply with COLREGs, then use a model-based method to calculate the action command.

III. PRELIMINARIES

A. COLREGS

Maritime vessels are forced to comply with collision regulations, COLREGs, which describe collision avoidance patterns. Three typical patterns, crossing, head-on, and overtaking, are presented in Fig. 2. It illustrates that target ships (TSs) are in different relative bearing regions of the own ship (OS). Each vessel treats itself as the OS from the first-person perspective; the other vessels are the TSs for ease of expression. For example, if two vessels meet head-on or nearly head-on, and there is a risk of collision, both vessels shall alter course to starboard. Rule 14 and Rule 15 deal with two vessels approaching from about 3 degrees off either bow, to 22.5 degree about either beam. A vessel that has another on her starboard side shall actively avoid the other. This vessel is termed as the give-way vessel; the other one is termed as the stand-on vessel [3].

Algorithm 1 Combining DRL and Low-Level Control for Goal-Oriented Navigation

```
Require: $P^G = (x^G, y^G)$
for t = 1 to T do
    ▷ Measure TSs states, relative goal state, relative pose state, and previous action state
    $s_t = (s^r, s^g, s^p, s^l)_t$
    for $H$ time steps do
        ▷ Predict the next waypoint based on the policy
        $\hat{p}_t = \mathrm{DRL}(s_t, P^G_t)$
        ▷ Calculate the desired course angle command based on LOS guidance law
        $\{\chi_d\}_t = \mathrm{LOS}(\hat{p}_t, p_t)$
        ▷ Trajectory tracking controller
        $\{X_\tau, Y_\tau, N_\tau\}_t = \mathrm{FLC}(\{\chi_d\}_t)$
    end for
end for
```

B. DYNAMIC MODELING OF A SHIP

To formulate the dynamic model of an autonomous surface vessel (ASV), we define the generalised coordinate position vector (position-orientation vector) $\boldsymbol{\eta} = [x, y, \psi]^T$, and the generalised velocity vector (linear-angular velocity vector) $\nu = [u, v, r]^T$. Based on the above definitions, the ship kinematic model follows in Equation (1). [5]

$$\dot{\boldsymbol{\eta}} = \mathbf{R}(\psi)\nu \tag{1}$$

where x and y are the positions in north and east, respectively; ψ refers to the heading angle relative to north. ν consists of surge velocity u, sway velocity v, and yaw rate r; $\mathbf{R}(\psi)$ refers to a rotation matrix.

$$\mathbf{R}(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

The general 3-DOF equation of the ship manoeuvring model can be written as follows [5]:

$$m\Big(\dot{u} - vr - x_G r^2\Big) = X_H + X_\tau \tag{3a}$$

$$m(\dot{v} + ur + x_G \dot{r}) = Y_H + Y_\tau \tag{3b}$$

$$I_{zz}\dot{r} - m x_G(\dot{v} + ur) = N_H + N_\tau \tag{3c}$$

where the first terms X_H, Y_H, and N_H above represent the hydrodynamic forces; The second terms X_τ, Y_τ, and N_τ represent the actuator forces, where $X_\tau \propto n\cos\delta$, and $Y_\tau \propto n\sin\delta$, given by the propeller shaft speed n and rudder deflection angle δ; $N_\tau \propto l_r * Y_\tau$ is proportional to Y_τ, along with the rudder length l_r [22].

IV. METHOD OF AUTONOMOUS SYSTEM DESIGN

A. OVERVIEW

The whole procedure of the autonomous system is summarized in Algorithm 1. The proposed guidance system in this study consists of a collision avoidance module and a line-of-sight (LOS) guidance module (refer to Fig. 1). The collision avoidance module is used to predict the next waypoint based on the OS observation, which contains the information about the encountered TSs, the OS's state relative to the goal, as well as its relative pose state. The predicted waypoint will guide the OS towards the direction of avoiding the

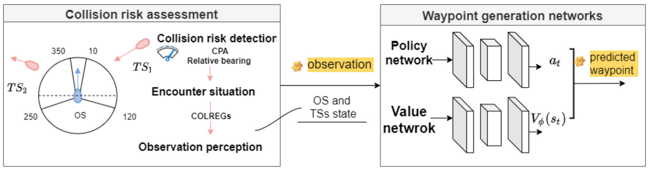


FIGURE 3. Block diagram of the collision avoidance module.

TSs with COLREGs compliance. The predicted waypoint $\hat{p}_t$ and the current vessel state p_t are the input to the LOS guidance module that provides the necessary course angle χ_d to achieve. The trajectory tracking controller module then calculates the steering and propulsion control commands to determine the necessary control forces and moments based on the course angle, that can drive the OS to the predicted next waypoint smoothly. It will repeat the process until the OS arrives at the destination.

B. COLLISION AVOIDANCE MODULE

As shown in Fig. 3, we develop the collision avoidance module to generate a sequence of waypoints by considering the collision risk with TSs and COLREGs compliance. More specifically, at time step t, the OS measures an observation s_t, and the goal location, P_t^G, which are specified in the OS's body-fixed coordinate frame, calculates a desired waypoint $\hat{p}_t = \text{DRL}(s_t, P_t^G)$ in the polar coordinate of the OS. (Line 6 in Algorithm 1). The collision avoidance module makes use of two submodules: collision risk assessment, and waypoint generation networks.

1) COLLISION RISK ASSESSMENT

In congested waters, where ships are navigating with frequent trajectory changes, a safe situation can suddenly become critical. Under these situations, the ships should pay careful attention to the encountered ships with high collision risk. As radar is being used as the primary tool for collision avoidance, the geometry of the encounters is defined by the safe passing distance, the CPA. CPA is termed as the closest point of approach, which indicates the minimum distance value between the OS and the approaching TS. Therefore, to take appropriate action early, we should evaluate the CPA-based collision risk in real-time. Assuming that the position vector from TS to OS is given as $\vec{P}_{\text{OT}} = x_{\text{OT}}\vec{i} + y_{\text{OT}}\vec{j}$; The velocity vector V_{OT} can be derived as $\frac{\mathrm{d}\vec{P}_{\text{OT}}}{\mathrm{d}t} = \vec{V}_{\text{OT}}$.

To calculate the minimum relative distance between TS and OS (DCPA), we reconstruct it into a minimization problem of finding the $\min \|\vec{P}_{\text{OT}}\|$ when $t \geq 0$. Correspondingly, the time t here can be regarded as the TCPA, time to the closest point of approach, only if the time is greater than zero.

$$\text{TCPA} = -\frac{\vec{P}_{\text{OT}}|_{t=0} \cdot \vec{V}_{\text{OT}}}{\vec{V}_{\text{OT}} \cdot \vec{V}_{\text{OT}}} \tag{4a}$$

$$\text{DCPA} = \vec{P}_{\text{OT}}|_{t=0} + \vec{V}_{\text{OT}} \cdot \text{TCPA} \tag{4b}$$

TABLE 1. COLREGs situation category.

COLREGs situations		Conditions
Head-on		$\beta_{\text{OS}} \in [0, 5.7°] \cup [354.3°, 360°)$ $\beta_{\text{TS}} \in [0, 5.7°] \cup [354.3°, 360°)$
Crossing	Give way	$\beta_{\text{OS}} \in [0, 112.5°]$
	Stand on	$\beta_{\text{OS}} \in [247.5°, 360°]$
Overtaking	Give way	$\beta_{\text{OS}} \in [0, 90°] \cup [270°, 360°)$ $\beta_{\text{TS}} \in (112.5°, 247.5°)$
	Stand on	$\beta_{\text{OS}} \in (112.5°, 247.5°)$ $\beta_{\text{TS}} \in [0, 90°] \cup [270°, 360°)$

As aforementioned, Fig. 2 illustrates the four main types of encounter situations that OS may confront: Crossing_stand on, Crossing_give way, Head-on, and Overtaking. They depend on both the relative bearing of the TS $\beta_{\text{OS}} = \text{atan2}(y_{\text{OT}}, x_{\text{OT}}) - \psi_T$ from the OS, and the relative bearing of OS β_{TS} from the TS [23]. When the DCPA and TCPA of two encountered ships trigger the collision risk condition, the relative bearing can be used to decide the type of the COLREGs situations. The TS is then categorized based on its instantaneous position and the heading ψ_T, as described in Table 1

2) WAYPOINT GENERATION NETWORKS

An essential capability for autonomous agents is to plan a collision-free trajectory in real-time when detecting the obstacles along the predefined path. The more encountered TSs are detected, the higher computation cost for real-time trajectory generation is needed. As a result, a learning-based method is proposed to meet real-time planning and collision avoidance demands. This learning procedure allows a more guided and efficient exploration in unknown environments based on the ship's prior experience with various environments. Using the learning-based method guarantees the trajectory planning in real-time but can significantly decrease the computational time by deploying the pre-trained policy [24].

We formulate collision avoidance as a sequential decision-making problem in a DRL framework, where an agent learns an optimal policy from interaction with the environment. Refer to Fig. 3, the agent measures states from the environment when collision hazard occurs, then transit them into a neural network architecture to calculate an action, the predicted waypoint. The agent can acquires experience and adapt to an unknown environment during the training process. The detailed definition of the observation, action, and reward function are presented as follows. The randomly chosen screenshot of the simulation environment is illustrated in Fig. 4.

Observation: The observation $\mathbf{s}_t$ are constructed by concatenating four features, which are listed in Table 2. $\mathbf{s}^r$ is the measurements of the last three consecutive frames from a 360 degrees distance sensor with a maximum range of L meters and 2 degrees interval. It will offers 180 distance values every data frame (i.e., $\mathbf{s}_t^r \in \mathbb{R}^{3\times180}$). The overall idea

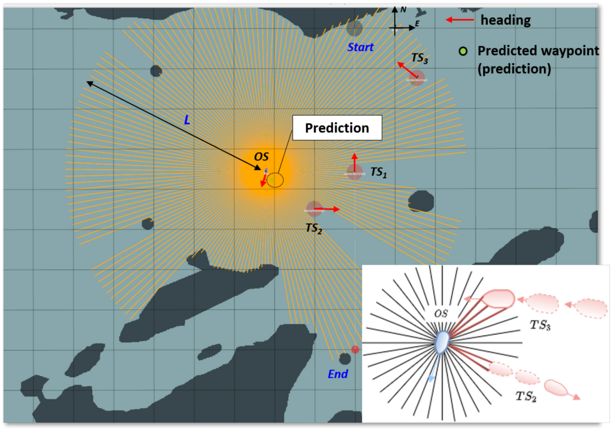


FIGURE 4. Snapshot of the training environment where the OS is approaching to its destination with multiple TSs.

TABLE 2. The observation space defined in the DRL algorithm.

Feature	State	Description
s^r	d_i	Sensor range at each degree, where $i = 0, 1, \ldots, 179$
s^g	$\|\psi_G - \psi\|$	Relative heading with the angle to the destination
	$\|P^G - p\|_2$	Distance to the destination from the current position
	P^G	Goal location in OS's body-fixed coordination frame
s^p	y_e	Cross-track error
	$\psi_e = \psi_d - \psi$	Heading angle error
	u	Surge velocity
s^l	a_{t-1}	The previous action at timestep t

is illustrated in Fig. 4, where an agent in OS scans for a complete 360-degree sector. The agent will take the vector of distance measurement d_i three times to build an accurate map of the environment, further to determine either the TS toward or away from the OS.

The relative goal state $\mathbf{s}^g$ represents the goal concerning the OS's current position in polar coordinates with distance and angle. It is a 2D vector. We define the relative pose state $\mathbf{s}^p$, including the cross-track error y_e and the heading error ψ_e, as the change in heading and cross distance needed for the OS to navigate straight toward the look-ahead point from the current position and heading. In addition, we consider the previous action $\mathbf{s}^l$ as the current observation space. At last, the observation space is normalized using the statistics aggregated over the entire training process.

Action Space: Based on the observation, the waypoint generation network outputs a predicted 2D waypoint $\hat{p}_t = (\hat{r}_t, \hat{\theta}_t)$ in the polar coordinate of the OS. Where $\hat{r}$ is defined as 5 times of ship's length, $\hat{\theta}$ is the turning angle of 10 degrees based on its current heading angle. The prediction generated by the training network is executed over a time horizon of H seconds, which satisfies COLREGs and collision-free constraints. Consequently, a new waypoint is generated until the OS arrives at the destination.

Reward Function: Reward functions are used to shape the agent's behavior by interacting with the environment. By getting positive or negative rewards for taking certain actions that completes or fails at a task in the environment, the agent should learn a policy resulting in good actions by trying to maximize its reward. The reward r at timestep t is designed as a sum of four terms, ${}^g r$, ${}^{ca} r$, ${}^{co} r$ and ${}^p r$, is presented in Eq. (5):

$$r_t = \left({}^g r\right)_t + \left({}^{ca} r\right)_t + \left({}^{co} r\right)_t + \left({}^p r\right)_t. \tag{5}$$

The objective of the agent is to reach the destination while avoiding collisions on the way. As a result, constraining the agent to reach the goal is intuitive. First, the agent is awarded by ${}^g r^t$ for achieving its goal, where P_t^G and p_t are the goal position and position of the vessel at time step t respectively:

$$\left({}^g r\right)_t = \begin{cases} r_{arrival} & \text{if } \|p_t - P_t^G\| < 1.0 \\ k_g\left(\|p_{t-1} - P_t^G\| - \|p_t - P_t^G\|\right) & \text{otherwise.} \end{cases} \tag{6}$$

where the r_{arrival} takes the value of 10.0 and k_g is set to -0.1. To learn an optimal policy for collision avoidance, there should be a significant negative reward for being involved in a collision. When the OS collides with the approaching TSs, it will be penalized by $({}^{ca} r)_t$:

$$\left({}^{ca} r\right)_t = \begin{cases} r_{collision} & \text{if } \|\mathbf{p}_{OS}^t - \mathbf{p}_{TS}^t\| < 2R \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

where the R takes the summation of the length between OS and TS. A third reward component is caused by non-compliance with COLREGs. When the OS comply with COLREGs, it will get the positive reward. Otherwise, it will be punished. The COLREGs reward $r_{colregs}$ has the maximum value 1 if the OS avoids a collision in those encounter situations.

$$\left({}^{co} r\right)_t = \begin{cases} r_{colregs} & \text{if comply with COLREGs} \\ -r_{colregs} & \text{otherwise.} \end{cases} \tag{8}$$

To ensure the vessel can smoothly converge to the predefined path, the path following reward component $({}^p r)_t = [({}^h r)_t, ({}^c r)_t]$ dependent on the cross-track error y_e and course angle error ψ_e are selected as the performance measure for the path following task.

$$\left({}^h r\right)_t = \begin{cases} C_{\psi_e} e^{-k_h\left((\psi_e)^2 + (\dot{\psi}_e)^2\right)} & \text{if } \psi_e < \psi \\ -r_{heading} & \text{otherwise.} \end{cases} \tag{9}$$

$$\left({}^c r\right)_t = \begin{cases} C_{y_e} e^{-k_c\left((y_e)^2 + (\dot{y}_e)^2\right)} & \text{if } y_e < y \\ -r_{cross} & \text{otherwise.} \end{cases} \tag{10}$$

Training Network: The basic architecture of the developed waypoint generation network is illustrated in Fig. 5. We use the Convolutional Neural Networks (CNN) to map the observation vector $\mathbf{s}^r$ from the distance sensor to the desired next waypoint $\hat{p}_t = (\hat{r}_t, \hat{\theta}_t)$. The first hidden layer convolves 16 one-dimensional filters with kernel size equals three over the three input scans and applies ReLU non-linearity. The second hidden layer convolves 16 one-dimensional filters

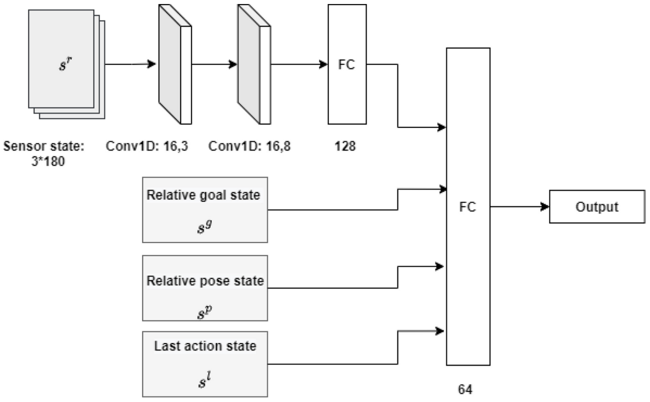


FIGURE 5. Structure of waypoint generation networks.

with kernel size equals eight and uses ReLU. The third hidden layer is a fully connected (FC) layer with 128 rectifier units. The output of the third hidden layer is concatenated with the other three feature inputs and then fed into the last FC layer with 64 rectifier units. Consequently, the network outputs an action from a Gaussian distribution.

Training Algorithm: We train our waypoint generation networks using Proximal Policy Optimization (PPO), in an actor-critic framework, to get an optimal policy $\pi_\theta(a_t|s_t)$, and a value function $\hat{A}_t$. PPO updates policies via

$$\theta_{t+1} = \theta_t + \alpha \bigtriangleup_\theta L^{PPO}(\theta) \tag{11}$$

Here, $L^{PPO}(\theta)$ is given by Equation. 12, in which ϵ is a clipping hyperparameter which roughly indicates how far away the new policy is allowed to go from the old. The value of epsilon is set to 0.2 in the paper. $\hat{E}$ denotes the empirical expectation over time steps. $\hat{A}_t$ is an estimate of the advantage function. The advantage function [25] represents how good a state action pair is compared with the average value of current state. $r_t(\theta) = \frac{\pi_\theta(a_i|s_i)}{\pi_{\theta_{\text{old}}}(a_i|s_i)}$ denotes the probability ratio between the updated and the previous policies. Correspondingly, the clip function will constraint the value of $r_t(\theta)$ between $1-\epsilon$ and $1+\epsilon$.

$$L^{PPO}(\theta) = \hat{E}\Big[\min\Big(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\Big)\Big] \tag{12}$$

Algorithm 2 describes the waypoint prediction procedure that runs on an agent. At each iteration, the agent collects T time steps of data, and implements the policy for T time steps. First, define the temporal difference residual $\delta_t = r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t)$, and compute the Generalized Advantage Estimator $\hat{A}_t$ based on the current state value function $V_\phi(s_t)$, which is approximated with a neural network with parameters ϕ. Second, we update the policy by maximizing the surrogate objective function $L^{PPO}(\theta)$ on these sampled trajectories. It is via gradient ascent with the Adam optimizer for E_π epochs. Thirdly, we construct the mean squared error loss $L^V(\phi)$ for $V_\phi(s_t)$, and optimize it with Adam optimizer for E_v epochs. $\pi_\theta(a_t|s_t)$ and $V_\phi(s_t)$ are updated independently since their parameters are not shared with each other. Finally, the algorithm for predicting the waypoints is given below:

Algorithm 2 Training the Waypoint Generation Network With the PPO Algorithm [26]

```
1: Initialize policy network $\pi_\theta$ and value network $V_\phi(s_t)$ using hyper-parameters in Table. 3.
2: for iteration $= 1, 2, \ldots,$ do
3:     Run policy $\pi_\theta$ for $T$ time steps, collecting $\{\mathbf{s}_t, r_t, a_t\}$, where $a_t = \hat{p}_t, t \in [0, T]$
4:     Estimate advantages using GAE [25], $\hat{A}_t = \sum_{l=0}^{T}(\gamma\lambda)^l\delta_t$, where $\delta_t = r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t)$
5:     break, if $T > T_{max}$
6:     $\pi_{old} \leftarrow \pi_\theta$
7:     for $j = 1, \ldots, E_\pi$ do
8:         $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{old}(a_t|s_t)}$
9:         $L^{PPO}(\theta) = \sum_{t=1}^{T_{max}} \mathbf{min}(r_t(\theta)\hat{A}_t, \mathtt{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)$
10:        Update $\theta$ with $lr_\theta$ by Adam [27] w.r.t $L^{PPO}(\theta)$
11:    end for
12:    for $k = 1, \ldots, E_V$ do
13:        $L^V(\phi) = -\sum_{t=1}^{T}(\sum_{t'>t}\gamma^{t'-t}r^{t'} - V_\phi(s_t))^2$
14:        Update $\phi$ with $lr_\phi$ by Adam w.r.t $L^V(\phi)$
15:    end for
16: end for
```

The hyper-parameters during the training process are tuned in Table 3.

TABLE 3. The hyper-parameters of our training algorithm described in Algorithm 2.

Parameter	Value
λ in line 4	0.95
γ in line 4 and 13	0.99
T_{max} in line 5	8000
E_π in line 7	20
lr_θ in line 10	5e5 (first stage), 2e5 (second stage)
E_V in line 12	10
lr_ϕ in line 14	1e3

C. LOS GUIDANCE MODULE

Given the predicted waypoint, a LOS guidance module provides its desired heading to the waypoint. The LOS guidance principle is illustrated in Fig. 6. The along-track x_e and cross-track error y_e for the current position of the OS $p(x, y)$ are given by [28]. Only the cross-track error is relevant for the path following purpose since $y_e = 0$ indicates that the OS has converged to the straight line. The line connects the previous waypoint $\hat{p}_{k-1}$ and the predicted waypoint $\hat{p}_k$.

$$\begin{bmatrix} x_e \\ y_e \end{bmatrix} = \mathbf{R}\big(\psi_p\big)\begin{bmatrix} x - \hat{x}_{k-1} \\ y - \hat{y}_{k-1} \end{bmatrix} \tag{13}$$

where $\mathbf{R}(\psi_p)$ is the rotation matrix from the initial frame to the path-fixed reference frame; $\hat{p}_k = (\hat{x}_k, \hat{y}_k)$ is the position of the k-th predicted waypoint represented in North-East-Down

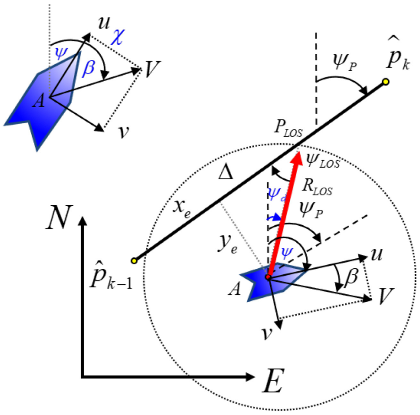


FIGURE 6. Line-of-sight guidance geometry for straight lines. The heading angle and sideslip angle are ψ and β, respectively.

TABLE 4. Simulation setup: case 2 and case 3 illustrate the multi-ship collision avoidance scenarios.

	Case 2		Case 3			
	Scenario 1		Scenario 1		Scenario 2	
OS	start point	[0,0]	start point	[0,0]	start point	[0,-80]
	end point	[0,600]	end point	[0,-500]	end point	[0,-650]
TSs	headon_1	[-50,600]	headon	[0,-500]	headon	[0,-750]
	[TS1]	[0,0]	[TS1]	[0,0]	[TS1]	[0,-150]
	headon_2	[50,600]	crossing starboard	[-300,-200]	crossing starboard_1	[-300,-350]
	[TS2]	[-30,0]	[TS2]	[100,-400]	[TS2]	[100,-550]
	headon_3	[0,-500]			crossing starboard_2	[-150,-300]
	[TS3]	[10,0]			[TS3]	[50,-300]
					crossing port	[200,-400]
					[TS4]	[-200,-400]

coordinate frame, and ψ_p is the horizontal path-tangential angle

$$\psi_p = \text{atan2}\big(\hat{y}_k - \hat{y}_{k-1}, \hat{x}_k - \hat{x}_{k-1}\big) \tag{14}$$

In marine guidance applications, the LOS vector starts at the OS's current position and passes through a point p_{LOS}, which is located on the path-tangential line at a lookahead distance $\triangle > 0$ ahead of the direct projection of the ship's position $p(x, y)$ on to the path [28]. The look-ahead distance value is chosen as constant value in this study.

In the presence of external disturbances β, the desired heading angle is derived in Eq. (15), based on the lookahead-based guidance law (Line 8 in Algorithm 1).

$$\psi_d = \psi_p + \arctan\left(\frac{-y_e}{\triangle}\right) - \beta \tag{15}$$

D. TRAJECTORY TRACKING CONTROLLER MODULE

This section focuses on tracking desired trajectories for the nonlinear system using feedback linearization control method [22] (Line 10 in Algorithm 1). Two feedback linearizing controllers, which act as the speed and the yaw rate controller; and One conventional PD controller, acting as the heading controller are implemented in the simulator. As presented in Eq. (16), the speed controller X_τ is on the form

$$\begin{aligned} F_x = -\Big(mvr + Y_{\dot{v}}vr + Y_{\dot{r}}r^2\Big) - (X_u + X_{|u|u}|u| \\ + X_{uuu}u^2)u + K_{p,u}m(u_d - u) \end{aligned} \tag{16}$$

the yaw rate controller Y_τ is shows in Eq. (17)

$$\begin{aligned} F_y = (mur - X_{\dot{u}}ur) - (Y_v v + Y_r r + Y_{|v|v}|v|v \\ + Y_{vvv}v^3) + K_{p,r}I_z(r_d - r)/l_r \end{aligned} \tag{17}$$

the heading PD-controller is in Eq. (18)

$$F_y = K_{p,\psi}I_z\big((\psi_d - \psi) - K_{d,\psi}r\big)/l_r \tag{18}$$

where F_x, F_y are the control forces on the forward direction and the yaw rate; u, v, r are the forward speed, lateral speed, and the yaw rate respectively; m is the inertia of the vessel; X, Y with different subscripts are the maneuvering coefficients of the vessel; $K_{p,u}$, $K_{p,r}$, and $K_{d,\psi}$ are the control parameters in this study.

V. EXPERIMENTAL RESULTS

A. SIMULATION SETUP

To validate the proposed approach for solving the multi-ship collision avoidance problem, we consider the autonomous navigation for an ASV transiting through a strait. The parameters of the ship model are referred to in the study [29]. Meanwhile, multiple TSs are approaching from its front, port, and starboard sides. The simulation setting is summarized in Table 4, from static obstacle to dynamic multiple TSs collision avoidances that can demonstrate the scalability of the proposed method. In all cases, the same hyperparameter settings of the training are used.

The OS is approaching the destination while avoiding collisions with encountered TSs. To achieve this, we trained our agent using the PPO algorithm as presented in the collision avoidance module. The time step size is set to $\Delta t = 0.1s$. For each training iteration, the agent exploits the policy to generate trajectories until the maximum of $T_{max} = 8000$-time steps. We select samples randomly from the trajectories, including the state, action, and reward. The selected mini-batch is used to compute a loss function that combines the policy surrogate and a value function error term. The parameters of the function are updated by the Adam optimizer. The procedure will repeat until completing the given training iteration.

B. CASE 1: COLLISION AVOIDANCE WITH STATIC OBSTACLES

Case 1 is shown in Fig. 7, the OS can converge to its desired path and toward the destination in a narrow channel. Before entering the channel, it starts to avoid static obstacles by maneuvering to the starboard. The right side of the picture shows a snapshot of the whole scenario. In stage 1, the OS starts to alter its course to the destination as its initial heading is −90 degrees. In stage 2, before arriving at the

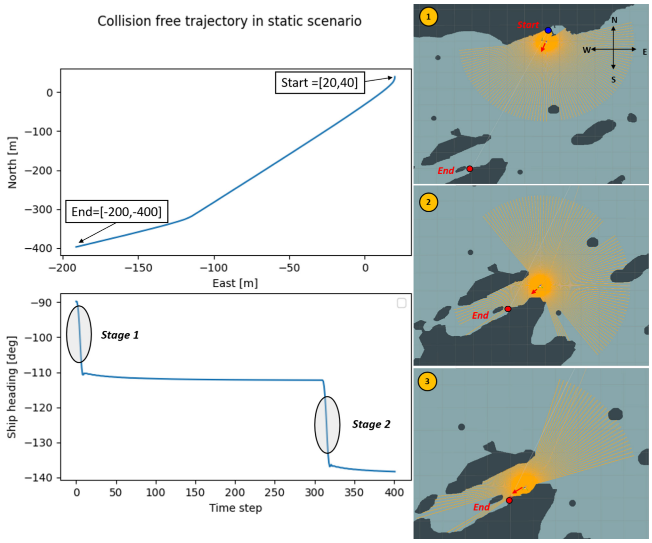


FIGURE 7. Collision avoidance with static obstacles: the desired trajectory passes through a narrow channel.

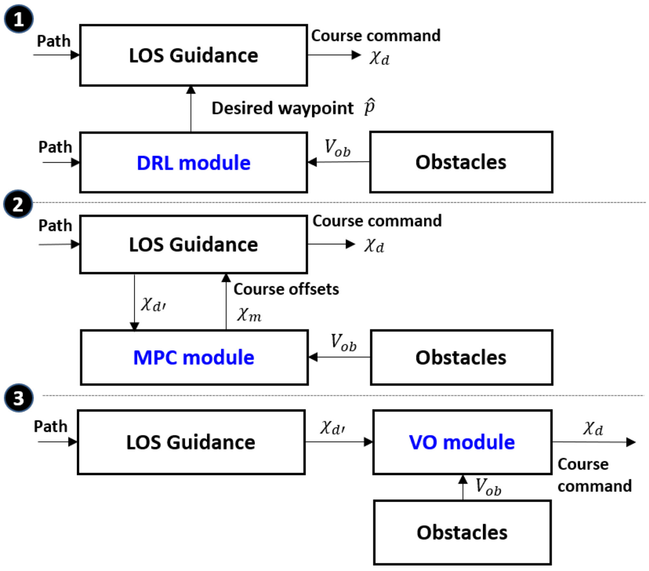


FIGURE 8. Guidance system design based on the three methods.

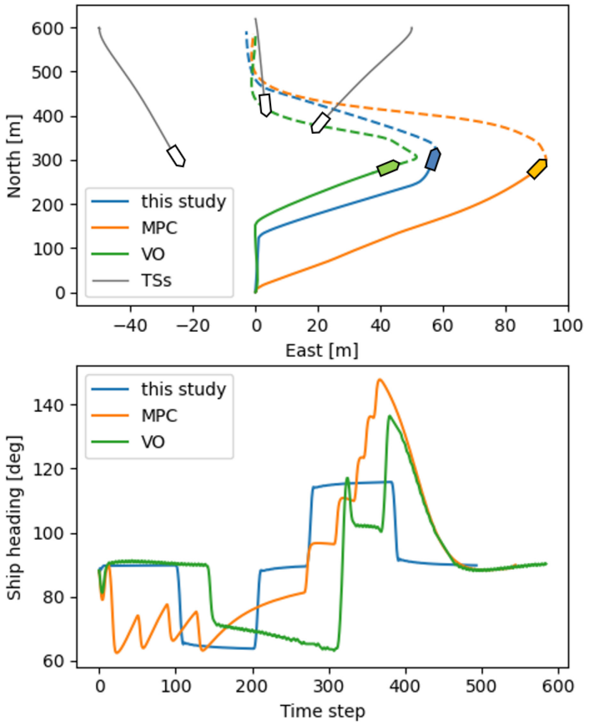


FIGURE 9. Head_on in case 2: based on the three methods, the OS successfully complete its avoidance maneuver for the TS approaching from its head-on side.

channel entrance, the OS changes its course again by having an acceptable distance to the entrance, which resides slightly outside the channel. Subsequently, the OS converges to the destination by avoiding static obstacles.

C. CASE 2: COMPARISON WITH MPC AND VO METHODS

When avoiding moving obstacles, the OS assumes to move straight from south to north when three TSs appear in front of it in constant heading and speed. Fig. 9 shows the resulting trajectory of the proposed method, with trajectories under MPC and VO as comparisons. All three guidance system designs have the similar ability to avoid collision, react to surrounding TSs, and return to course, while in compliance with COLREGs. The proposed method is therefore considered on par with the other two widely used methods. As shown in Fig. 8, a LOS guidance module is used to calculate a course command that guides the ship converging the straight line connecting the previous and the current waypoints. The MPC module provides a course angle offset χ_m such that the actual course command is $\chi_d = \chi_{d'} + \chi_m$; While the VO-based guidance system simply reacts and replans using the latest information of the surrounding environment.

For the case considered, the MPC [10] and VO [22] methods solve the problem in the condition of the information of pre-known of surrounding information, including the dynamics of the ship, the dynamics of the steering and propulsion system, and any number of TSs. While in this study, with no a priori knowledge of the environment except for the waypoints of its desired path, the agent of OS makes decisions based on a range sensor measuring the distance to surrounding obstacles. In comparison to the two methods Fig. 9, the heading angle in this study presents a more stable and smooth maneuvering effect.

D. CASE 3: COMPARISON WITH END-TO-END LEARNING APPROACH

For end-to-end DRL, the system is directly trained holistically towards the final goal by the overall objective function at each learning step. In the study of [14], the end-to-end DRL directly maps the states of TSs to an OS's control commands in terms of rudder angle. The oscillation of the control command is inevitable. The proposed method is compared to the end-to-end DRL method by planning a collision-free trajectory in case 3. It shows that the proposed

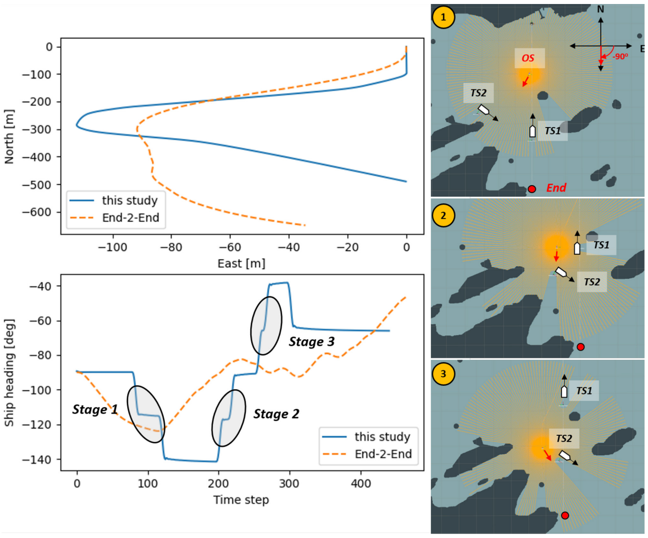


FIGURE 10. Comparison with MPC and VO methods.

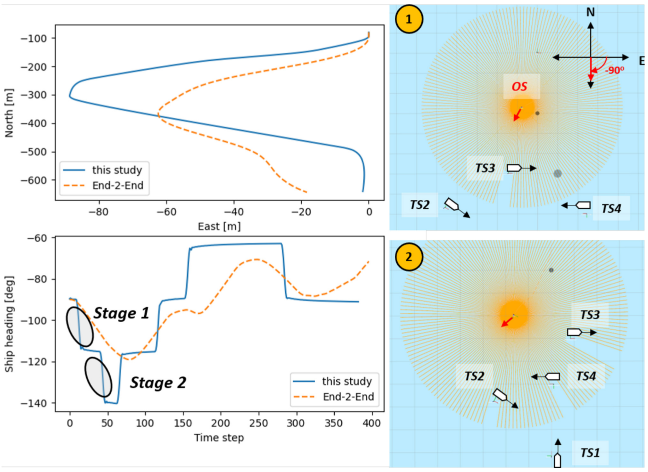


FIGURE 11. Comparison with end-to-end learning approach.

method can achieve a relatively stable effect better than the end-to-end DRL method.

The trajectory of the OS and corresponding heading are illustrated in Fig. 10 and Fig. 11. The considered scenario 1 is characterized by the OS that is transiting strait while simultaneously is facing a starboard crossing and head-on coming. The OS starts to alter its course when detecting the collision risk with a margin of $L = 300m$. The optimal control behavior corresponds to a course change toward starboard side until all the TSs are passed at a safe distance on OS's port side.

Fig.11 illustrates a similar situation: while the OS is transiting strait, two vessels TS2 and TS3 are approaching from the starboard side, TS4 is approaching from starboard side, and TS1 is coming from head-on side. At the beginning, TS3 appears in the OS's detection field. The optimal control strategy of OS is to make a change in course towards starboard as shown in stage 1. Later on, another TS2 comes in from the starboard side such that the OS needs to alter its course more to the starboard side. When the OS passes at a safe distance in front of TS 2 and 3, it starts to converge to its destination.

VI. CONCLUSION AND FUTURE WORKS

In this paper, we have designed and implemented an autonomous system, enabling the collision avoidance of multiple ships. Due to the difficulties arising from uncertainties of the dynamic model and the extremely high cost of exploring and sampling computations. A hierarchical method, combines global path planning based on DRL and local motion control together to naturally handle the real-time path planning and efficiency of both calculation and sampling problems. The learning process and control procedure are designated to achieve specific subgoals, instead of the entire navigation stack. This combination emphasizes the benefit of deploying DRL in global planning, i.e., providing a collision-free trajectory that has online planning capabilities, and alleviating the sensitivity to the heading angle changes. Furthermore, the model-based control strategy can provide an accurate control command to actuators, greatly offloading the computational load from the learning process. Moreover, it is expected to implement into the real application better than the end-to-end reinforcement learning method. The simulation result demonstrated that this proposed method can successfully be applied to the autopilot task with relatively smooth trajectories. For future work, we will deploy a more accurate tracking controller, further validate the proposed algorithm in real-world experiments.

REFERENCES

[1] *Yara Birkeland Press Kit*, Yara Int., Oslo, Norway, 2019.

[2] H. Gale and D. Patraiko, "Improving navigational safety," presented at Seaways, 2007.

[3] *Convention on the International Regulations for Preventing Collisions at Sea, 1972 (Colregs)*, Int. Maritime Organ., London, U.K., 1972.

[4] Y. Huang, L. Chen, P. Chen, R. R. Negenborn, and P. van Gelder, "Ship collision avoidance methods: State-of-the-art," *Saf. Sci.*, vol. 121, pp. 451–473, Jan. 2020.

[5] T. I. Fossen, "*Marine Control Systems: Guidance. Navigation, and Control of Ships, Rigs and Underwater Vehicles*. Trondheim, Norway: Marine Cybern., 2002. [Online]. Available: https://www.marinecybernetics.com

[6] X. Xiao, B. Liu, G. Warnell, and P. Stone, "Motion control for mobile robot navigation using machine learning: A survey," 2020, *arXiv:2011.13112*,

[7] J. Hu, X. Yang, W. Wang, P. Wei, L. Ying, and Y. Liu, "Obstacle avoidance for UAS in continuous action space using deep reinforcement learning," *IEEE Access*, vol. 10, pp. 90623–90634, 2022.

[8] 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.

[9] 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.

[10] I. B. Hagen, D. K. M. Kufoalor, E. F. Brekke, and T. A. Johansen, "MPC-based collision avoidance strategy for existing marine vessel guidance systems," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2018, pp. 7618–7623.

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

[12] P. Chen, Y. Huang, E. Papadimitriou, J. Mou, and P. van Gelder, "Global path planning for autonomous ship: A hybrid approach of fast marching square and velocity obstacles methods," *Ocean Eng.*, vol. 214, Oct. 2020, Art. no. 107793.

[13] 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.

[14] L. Zhao and M.-I. Roh, "COLREGS-compliant multiship collision avoidance based on deep reinforcement learning," *Ocean Eng.*, vol. 191, Nov. 2019, Art. no. 106436.

[15] R. Valiente, B. Toghi, R. Pedarsani, and Y. P. Fallah, "Robustness and adaptability of reinforcement learning-based cooperative autonomous driving in mixed-autonomy traffic," *IEEE Open J. Intell. Transp. Syst.*, vol. 3, pp. 397–410, 2022.

[16] R. Alms, A. Noulis, E. Mintsis, L. Lücken, and P. Wagner, "Reinforcement learning-based traffic control: Mitigating the adverse impacts of control transitions," *IEEE Open J. Intell. Transp. Syst.*, vol. 3, pp. 187–198, 2022.

[17] Y. Cui, S. Osaki, and T. Matsubara, "Reinforcement learning boat autopilot: A sample-efficient and model predictive control based approach," in *Proc. IROS*, 2019, pp. 2868–2875.

[18] F. Xia, C. Li, R. Martín-Martín, O. Litany, A. Toshev, and S. Savarese, "ReLMoGen: Leveraging motion generation in reinforcement learning for mobile manipulation," 2020, *arXiv:2008.07792*.

[19] H.-T. L. Chiang, J. Hsu, M. Fiser, L. Tapia, and A. Faust, "RL-RRT: Kinodynamic motion planning via learning reachability estimators from RL policies," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 4298–4305, Oct. 2019.

[20] A. Faust et al., "PRM-RL: Long-range robotic navigation tasks by combining reinforcement learning and sampling-based planning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2018, pp. 5113–5120.

[21] S. Bansal, V. Tolani, S. Gupta, J. Malik, and C. Tomlin, "Combining optimal control and learning for visual navigation in novel environments," in *Proc. Conf. Robot Learn.*, 2020, pp. 420–429.

[22] T. Stenersen, "Guidance system for autonomous surface vehicles," M. S. thesis, Dept. Eng. Cybern., Norwegian Univ. Sci. Technol., Trondheim, Norway, 2015.

[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, Aug. 2017.

[24] T. T. Nguyen, N. D. Nguyen, and S. Nahavandi, "Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications," *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 3826–3839, Sep. 2020.

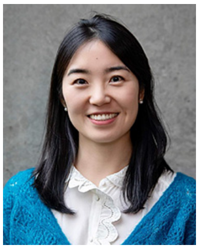
[25] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," 2015, *arXiv:1506.02438*.

[26] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.

[27] D. P. Kingma and J. A. Ba, "A method for stochastic optimization," 2014, *arXiv:1412.6980.*

[28] M. Breivik and T. I. Fossen, "Guidance laws for autonomous underwater vehicles," in *Underwater Vehicles*, vol. 4. London, U.K.: IntechOpen, 2009, pp. 51–76.

[29] Ø. A. G. Loe, "Collision avoidance for unmanned surface vehicles," M.S. thesis, Institutt for teknisk kybernetikk, Norwegian Univ. Sci. Technol., Trondheim, Norway, 2008.



LUMAN ZHAO (Member, IEEE) received the Ph.D. degree in naval architecture and ocean engineering from Seoul National University, South Korea, in 2019.

From 2019 to 2022, she was a Postdoctoral Research Associate with the Intelligent Systems Laboratory, Department of Ocean Operations and Civil Engineering, Norwegian University of Science and Technology, Norway. She is currently a Senior Researcher with the Group Research and Development, DNV. Her research interests include intelligent system development, automatic traffic scenarios generation for system testing, and trustworthy artificial intelligence applications.

GUOYUAN LI (Senior Member, IEEE) received the Ph.D. degree in computer science from the Department of Informatics, Institute of Technical Aspects of Multimodal Systems, University of Hamburg, Hamburg, Germany, in 2013.

In 2014, he joined the Department of Ocean Operations and Civil Engineering, Intelligent Systems Laboratory, Norwegian University of Science and Technology, Ålesund, Norway, where he is currently a Professor of Ship Intelligence. His research interests include modeling and simulation of ship motion, autonomous navigation, intelligent control, optimization algorithms, and locomotion control of bio-inspired robots. He has published more than 70 articles in these areas.

HOUXIANG ZHANG (Senior Member, IEEE) received the Ph.D. degree in mechanical and electronic engineering in 2003.

He is a Full Professor with the Department of Ocean Operations and Civil Engineering, Faculty of Engineering, Norwegian University of Science and Technology (NTNU), Norway. In 2004, he worked as a Postdoctoral Fellow and a Senior Researcher with the Institute of Technical Aspects of Multimodal Systems, Department of Informatics, Faculty of Mathematics, Informatics and Natural Sciences, University of Hamburg, Germany. In February 2011, he finished the Habilitation on Informatics with the University of Hamburg. In April 2011, he joined NTNU, where he is a Professor of Mechatronics. From 2011 to 2016, he also hold a Norwegian National GIFT Professorship on product and system design funded by the Norwegian Maritime Centre of Expertise. He has engaged in two main research areas, including control, optimization, and AI application, especially on autonomous vehicles; and marine automation, digitalization, and ship intelligence. He has applied for and coordinated more than 30 projects supported by the Norwegian Research Council, German Research Council (DFG), EU, and industry. In these areas, he has published over 200 journal and conference papers as an author or a co-author.

Dr. Zhang has received four best paper awards, and five finalist awards for Best Conference Paper at the International conference on Robotics and Automation. He was elected to be a member of the Norwegian Academy of Technological Sciences in 2019.