Title	Collision probability reduction method for tracking control in automatic docking/berthing using reinforcement learning
Author(s)	Wakita, Kouki; Akimoto, Youhei; Rachman, Dimas M. et al.
Citation	Journal of Marine Science and Technology (Japan). 2023, 28, p. 844-861
Version Type	VoR
URL	https://hdl.handle.net/11094/93224
rights	This article is licensed under a Creative Commons Attribution 4.0 International License.
Note	

Collision probability reduction method for tracking control in automatic docking/berthing using reinforcement learning

Kouki Wakita[1] · Youhei Akimoto[2,3] · Dimas M. Rachman[1] · Yoshiki Miyauchi[1] · Atsuo Maki[1]

Abstract
Automation of berthing maneuvers in shipping is a pressing issue as the berthing maneuver is one of the most stressful tasks seafarers undertake. Berthing control problems are often tackled by tracking a predefined trajectory or path. Maintaining a tracking error of zero under an uncertain environment is impossible; the tracking controller is nonetheless required to bring vessels close to desired berths. The tracking controller must prioritize the avoidance of tracking errors that may cause collisions with obstacles. This paper proposes a training method based on reinforcement learning for a trajectory tracking controller that reduces the probability of collisions with static obstacles. Via numerical simulations, we show that the proposed method reduces the probability of collisions during berthing maneuvers. Furthermore, this paper shows the tracking performance in a model experiment.

Keywords Berthing control · Trajectory tracking · VecTwin rudder · Reinforcement learning · TD3

1 Introduction

Autonomous vessels are becoming increasingly important due to crew shortages, cost reduction, and safety considerations. In particular, the automation and autonomy of berthing maneuvers are significant issues because berthing maneuvering is one of the most stressful maneuvers seafarers undertake.

Numerous studies have been conducted on the automation of docking and berthing operations for surface vessels [1–13]. For a solution to the berthing control problem to be useful in practical situations, the solution must be generated in real time. This problem is challenging due to the computational constraints and the complexity and uncertainty of maneuverability. Several studies have tackled this problem in two stages: planning and tracking/following. For instance, Shouji et al. [1, 2] discussed the optimal control problem for automatic berthing and proposed a numerical solution. Additionally, they proposed a feedback controller for tracking the solution. Ahmed et al. [3] proposed an artificial neural network controller for berthing. Their research incorporated the utilization of a reference line, referred to as an imaginary line, to plan the berthing maneuver. Bitar et al. [4] proposed an optimization-based trajectory planner for docking operation and used a dynamic positioning controller for trajectory tracking, and Martinsen et al. [5] addressed the presence of unmapped objects by using range sensors. Sawada et al. [6] proposed a berthing system that applied a path-following and path-planning algorithm.

We focus on the approach using *trajectory planning* and *trajectory tracking*. Here, in this paper, a *trajectory* is defined as a time series of state variables, which are subject to spatial and temporal constraints that take into account the vessel dynamics. The generation of a trajectory is called *trajectory planning,* and the tracking of a given trajectory is called *trajectory tracking*. In particular, the desired trajectory of berthing maneuvering is called the *berthing trajectory*. By preparing the berthing trajectory in advance, it is possible to solve the berthing control problem in real time only via tracking the prepared trajectory.

✉ Kouki Wakita
kouki_wakita@naoe.eng.osaka-u.ac.jp

Atsuo Maki
maki@naoe.eng.osaka-u.ac.jp

1 Osaka University, 2-1 Yamadaoka, Suita, Osaka, Japan

2 Faculty of Engineering, Information and Systems, University of Tsukuba, 1-1-1 Tennodai, Tsukuba, Ibaraki 305-8573, Japan

3 RIKEN Center for Advanced Intelligence Project, 1-4-1 Nihonbashi, Chuo-ku, Tokyo 103-0027, Japan

However, tracking the predefined berthing trajectory is not straightforward because the vessel speed changes significantly, the effect of wind disturbance increases at low-speed, and complex maneuvers, including backward and crabbing motion, are frequently required. Therefore, an essential task is to develop a tracking controller that can safely track a given berthing trajectory.

1.1 Related research

Considerable research has been conducted on trajectory tracking control of surface vessels [14–21]. For instance, there exists much research work on the application of nonlinear model predictive control (NMPC) to trajectory tracking control [17, 20], and the development of tracking control based on backstepping control methods for robust control in situations where uncertainties due to disturbances exist [16, 18, 19]. Studies have also been undertaken on dynamic positioning control, focusing on situations where the vessel speed is near zero [15].

Research on adopting reinforcement learning (RL) to surface vessel control has recently increased [22–28]. RL is an area of machine learning and can be used to find the optimal action maximizing a cumulative reward obtained in an environment containing uncertainty [29]. Although RL requires training, RL does not require real-time optimization computation because it uses function approximation via a neural network (NN). Martinsen et al. implemented an algorithm based on the deep deterministic policy gradient method to obtain a controller that minimizes the tracking errors in straight and curved paths in situations involving unknown currents [23, 24]. Subsequently, Martinsen et al. proposed an RL-based control scheme for trajectory tracking control that achieves both DP control and path following [25]. Moreover, Martinsen et al. extended the scheme to improve the tracking performance using NMPC [28].

However, berthing trajectories typically consist of complex motions such as turning, stopping the ship, backward motion, spinning, and crabbing motion. Therefore, training for trajectories that consist of any combination of motions is required.

Additionally, in berthing maneuvers, the avoidance of collisions is also required. This is because a small tracking error can cause a collision in berthing maneuvers because the purpose of the berthing is to bring the vessel close to the desired berth, which in itself represents an obstacle. Moreover, it is impossible to keep tracking errors equal to zero in an uncertain environment.

Some studies on collision avoidance of surface vessels have already been undertaken [26, 27]. However, those studies treated path following and collision avoidance as dual objective problems. In this case, path following and collision avoidance are in a trade-off relationship. The vessel may not be able

to reach the berthing point due to avoiding a collision with the berth. Therefore, in berthing maneuvers, it is effective to avoid tracking errors that increase the probability of collisions.

1.2 Scope of this study

The purpose of this study is to develop a methodology that avoids the tracking errors that may cause collisions with static obstacles. This paper proposes a training method for a trajectory tracking controller that reduces the probability of collision with static obstacles. The main contributions of this paper are as follows:

1. The generation of random trajectories generated from maneuvering simulation was introduced to permit trajectory tracking controllers to track complex trajectories.
2. The development of a method to generate static pseudo-obstacles depending on the desired trajectory was proposed for training on the avoidance of collisions with obstacles.
3. A measure representing the distance between a static obstacle and the vessel position was introduced to give the geometric information describing the obstacle to the controller.
4. A new reward function was proposed to train the trajectory tracking controller to prioritize the avoidance of the variety of tracking errors that may cause a collision.
5. The potential of the proposed method to reduce the probability of collisions in berthing maneuvering was demonstrated via numerical simulations. Furthermore, the results of the model experiments conducted outdoors are shown.

The proposed method requires a training environment including a maneuvering model and response characteristics of steering systems. However, it is unnecessary to prepare a large number of berthing trajectories and obstacle information for training; only berthing trajectory and obstacle information in the target harbor is required.

The remainder of the paper is organized as follows: Sect. 2 describes the training method of the trajectory tracking controller. Section 3 describes how the trajectory tracking controller can be applied to berthing maneuvers. Section 4 reports and analyzes the numerical simulation and model experiment results of berthing maneuvering. Finally, Sect. 5 presents the conclusions of the study.

2 Methods

The subject ship is a 3-meter model ship shown in Fig. 1. L and B denote the length and breadth of this model ship and their values listed in Table 1. This ship is equipped with

Fig. 1 Subject ship in Inukai pond

Table 1 Particular of the model ship

| Length: L | 3.000 (m) |
| Breadth: B | 0.484 (m) |

Table 2 Limits of the actuator state variables

Item	Lower limit	Upper limit
δ_p	−105 (deg.)	35 (deg.)
δ_s	−35 (deg.)	105 (deg.)
n_p	10 (rps)	10 (rps)
n_{bt}	−30 (rps)	30 (rps)

a single propeller, a VecTwin rudder, and a bow thruster. VecTwin rudder system [30] is a rudder system that has a pair of fishtail rudders and enables various maneuvers. The VecTwin rudder system significantly enhances ship maneuverability [31] and enables the development of low-speed positioning systems [32]. In this paper, the rudders of the VecTwin rudder system are called as VecTwin rudder.

The actuator states is defined as $\boldsymbol{u} \equiv \left(\delta_p, \delta_s, n_p, n_{bt}\right)^\top \in \mathbb{R}^4$, where δ_p represents the rudder angle on the port side, δ_s represents the rudder angle on the starboard side, n_p is the propeller revolution number, and n_{bt} is the bow thruster revolution number. In this study, the response characteristics of the steering rudder are taken into account. The control commands are defined as $\boldsymbol{u}_{cmd} \equiv \left(\delta_{p,cmd}, \delta_{s,cmd}, n_{p,cmd}, n_{bt,cmd}\right)^\top \in \mathbb{R}^4$. The upper and lower limits of the actuator state variables are defined in Table 2. Note that the propeller revolution number is constant at 10 rps because the VecTwin rudder system enables various motions by changing the rudder angle while maintaining a constant propeller revolution number. This revolution number 10 rps allows the vessel to reach the typical vessel speed of harbor maneuvers on a real scale.

In this study, the motion of the vessel in the harbor is modeled by a surge–sway–yaw three-degrees-freedom equation of motion. The coordinate systems are earth-fixed coordinates, represented by $o_0 − x_0 y_0 z_0$ and ship-fixed coordinates, represented by $o − xyz$, which has its origin on midship. The coordinate systems are shown in Fig. 2. ψ, u, v_m, and r denotes the heading angle, the surge velocity, the sway velocity of
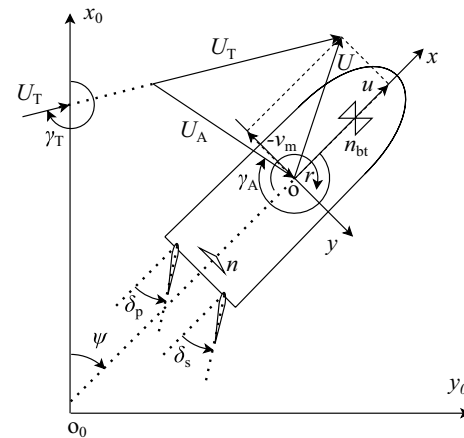

Fig. 2 The coordinates systems used in this work

the midship, and the yaw angular velocity, respectively. Pose vector of the vessel is defined as $\boldsymbol{\eta} \equiv \left(x_0, y_0, \psi\right)^\top \in \mathbb{R}^3$, and velocity vector is defined as $\boldsymbol{v} \equiv \left(u, v_m, r\right)^\top \in \mathbb{R}^3$. The vessel state vector is defined as $\boldsymbol{x} \equiv \left(\boldsymbol{\eta}^\top, \boldsymbol{v}^\top\right)^\top \in \mathbb{R}^6$. The observed vessel state vector is defined as $\hat{\boldsymbol{x}} \in \mathbb{R}^6$. The vector of true wind speed and direction is defined as $\boldsymbol{w} \equiv \left(U_T, \gamma_T\right)^\top \in \mathbb{R}^2$, where U_T represents the true wind speed, and γ_T is the true wind direction. The true wind speed and direction are assumed to depend on time but not space.

2.1 Optimization of the trajectory tracking controller

In this study, the trajectory tracking problem of minimizing the tracking error is formulated as a maximization problem related to a reward. This section describes the formulation used here.

This paper focuses on the trajectory tracking of the pose vector, $\boldsymbol{\eta}$. Here, the desired pose vector is defined as $\boldsymbol{r} \in \mathbb{R}^3$. The time series of desired pose vector can be represented as follows:

$$\mathscr{R} = \left\{\boldsymbol{r}_k\right\}_{k=0,\ldots,N-1} , \tag{1}$$

where k denotes discrete-time steps, and N is the total number of steps in the desired trajectory. The time of k-th step is defined as t_k, and the time step between t_k and t_{k+1} is defined as Δt. The time step between successive decisions in the RL algorithm is also equal to Δt.

In this study, the geometric information related to static obstacles is assumed to be given. The static obstacles are represented by multiple polygons which can be defined,

$$\mathscr{O} = \bigcup_{k=1}^{N_{obs}} S_i , \tag{2}$$

where S_i represents the i-th obstacle polygon, and N_{obs} represents the total number of obstacle polygons.

In RL, the state variable for decision-making is often represented by s and the action variables by a; these conventions is adopted in this work. The trajectory tracking controller makes decisions depending on the observed vessel state, $\hat{x}$, the actuator state, u, the desired trajectory, $\mathscr{R}$, and the static obstacle, $\mathscr{O}$. Thus, the state variable and the action variables at k-th step are defined as,

$$
\begin{cases}
s_k = f_s(\hat{x}_k, u_k, \mathscr{R}, \mathscr{O}) \\
a_k = u_{\mathrm{cmd},k} \ ,
\end{cases}
\tag{3}
$$

where f_s is a function that generates the input of the controller; this function is described in Sect. 2.3. The action can then be determined as $a_k = \mu(s_k)$, where μ is the policy function. This function is the trajectory tracking controller and is represented by a NN.

This study considers trajectory tracking as an episodic task in RL. A time series of the state and action variables throughout the episode is defined as $\tau = (s_0, a_0, \ldots, s_{N-1}, a_{N-1}, s_N)$. Let's assume that τ is distributed according to a probability distribution ρ^μ. The distribution ρ^μ is defined as follows:

$$
\rho^\mu = p(s_0) \prod_{k=0}^{T-1} p(s_{k+1}|s_k, a_k) \mu(s_k) \ ,
\tag{4}
$$

where $p(s_0)$ represents the probability distribution of the initial state, s_0, observed from the environment; $p(s_{k+1}|s_k, a_k)$ represents the state transition probability distribution. These distributions were implemented in the numerical simulations presented in this work and are described in detail in Sect. 2.2. The objective function described in Eq. 5 represents the expected cumulative reward obtained in an uncertain environment. Then, an objective function with a finite-time horizon can be defined as,

$$
J(\mu) = E_{\tau \sim \rho^\mu} \left[\sum_{k=0}^{N-1} \gamma^k R(t_k, s_k, a_k) \right] \ ,
\tag{5}
$$

where $E_{\tau \sim \rho^\mu}[\cdot]$ represents the expectation value inside the bracket when τ is generated according to the distribution ρ^μ, γ is the discount rate, and $R(\cdot)$ is the reward function. The discount rate is a hyper-parameter that implies the idea of discounting the reward to be obtained in the distant future. The reward function $R(\cdot)$ is described in detail in Sect. 2.4.

This objective function means the cumulative rewards obtained in an episode of the training environment. The trajectory tracking controller that maximizes this objective function is obtained by RL.

In the field of artificial intelligence, various RL algorithms have been developed [29]. In the optimization problem of this paper, the action space is continuous, and deterministic policy functions are used. Therefore, the RL algorithms called deep deterministic policy gradient (DDPG) [33, 34] and twin-delayed deep deterministic policy gradient (TD3) [35] can be applied; TD3 is an algorithm that improves on DDPG and was used in this study.

2.2 Training environment

This section describes the training environment used in this work. In training, the desired trajectory and the static obstacle are generated for each episode. After that, maneuvering simulations were conducted. The methods used for the generation of the desired trajectory and static obstacles are described in Sect. 2.2.1. The environment of maneuvering simulation is described in detail in Sect. 2.2.2.

2.2.1 Desired trajectory and pseudo-obstacle

In training, the desired trajectories, $\mathscr{R}$, are generated stochastically for each episode, and the static pseudo-obstacles, $\mathscr{O}$, are generated depending on the desired trajectories. This section describes the generation methods used for these variables.

The manner in which the desired trajectories were generated is described here. The target of this research is to train the trajectory tracking controller for berthing maneuvers. It is reasonable to train on a set of desired berthing trajectories in the target harbor. However, collecting many berthing trajectories is a time-consuming task. Moreover, training based on a small number of specific trajectories is undesirable because it may be necessary to change the desired trajectory due to practical considerations. Therefore, in this paper, trajectories obtained from a maneuvering simulation subject to random control input are introduced.

The trajectories generated for training are required to include complex motions such as turning and backward motion. In this maneuvering simulation, control inputs were determined randomly at each time step of the simulation. The distribution followed by control inputs was determined so that the time average of the thrusts generated from actuators was around zero. Thus, the bow thruster revolution number was determined based on the uniform distribution over the interval listed in Table 2. The port and starboard side rudder angles were determined based on the normal distributions of $\mathcal{N}(75, 30^2)$ and $\mathcal{N}(-75, 30^2)$, respectively. Here, the mean values of these distributions were set to take the value at which the net forces and the moment approached nearly zero.

The initial state of the maneuvering simulation was determined randomly. The initial vessel velocity was determined based on the uniform distribution, whose interval is listed

**Table 3** Initial velocity interval of the uniform distribution used to generate the desired trajectory

Item	Lower limit	Upper limit
u	−0.072 (m/s)	0.437 (m/s)
v_m	−0.07 (m/s)	0.07 (m/s)
r	−0.1 (deg./s)	0.1 (deg./s)
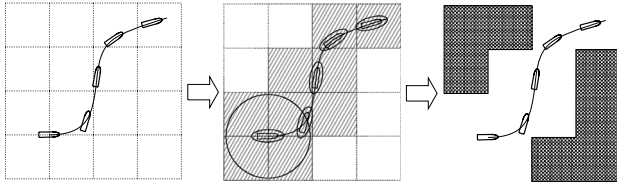
Fig. 3 The generation method of static pseudo-obstacles used in this work

in Table 3. The selection of this interval was based on the typical vessel speed of harbor maneuvers on a real scale. The initial vessel pose was set to zero. In this maneuvering simulation, wind disturbance was neglected.

The generation method of the static obstacle is described. The static obstacles used in the training procedure were automatically generated according to the desired trajectory. Here, we impose that the desired trajectory does not contact any obstacle. We also defined that the generated obstacles consist of line segments that are all longer than the vessel length. A schematic of the proposed scheme is presented in Fig. 3 and can be summarized in the following steps:

1. A grid is set based on $x_0 y_0$ plane. Here, one of the grid points is set to coincide with the origin of the coordinate system used. The grid size of both the x-axis and y-axis is set to be $2L$, and the number of grid elements was set such that it covered the region where the desired trajectory exists. The distance of the generated pseudo-obstacle from the desired trajectory can be controlled by the grid size. If the grid size is small, the distance between the obstacle and trajectory is close. Therefore, the grid size $2L$ was determined so that the environment is not too severe.

2. The area where obstacles must not exist is then determined according to the desired trajectory. It is assumed that no obstacles are generated in the area through which the vessel passes when the motion of the desired trajectory is performed. Here, the vessel shape was represented by an ellipse whose semi-major axis was $0.75L$ and semi-minor axis was B. Additionally, no obstacles would be generated in a circular area of radius $1.9L$ centered at the origin of the coordinate system. The circular area is introduced to avoid collisions arising from initial
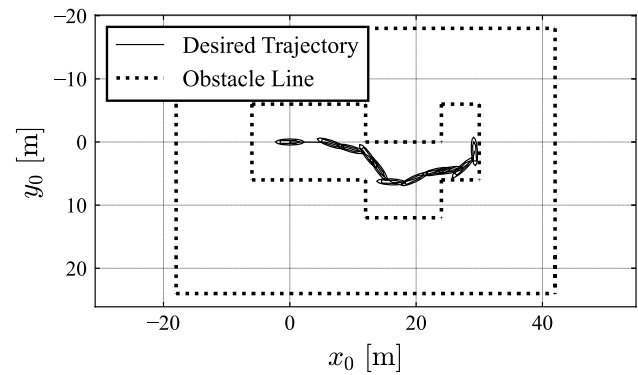
Fig. 4 An example of a desired trajectory with the static pseudo-obstacles used in training

Table 4 Uniform distribution intervals of the initial tracking error from the berthing trajectory

Item	Lower limit	Upper limit
x_0	$-L$ (m)	L (m)
u	−0.036 (m/s)	0.036 (m/s)
y_0	$-L$ (m)	L (m)
v_m	−0.007 (m/s)	0.007 (m/s)
ψ	−10 (deg.)	10 (deg.)
r	−0.1 (deg./s)	0.1 (deg./s)

tracking errors. Considering the initial tracking errors listed in Table 4, no obstacle should be generated in the four grids tangent to the origin of the coordinate system. The radius of the circular area was determined to be $1.9L$ so that no obstructions are generated in the four grids.

3. All grids through which the vessel does not pass are defined as static obstacles.

An example of the generated trajectory and pseudo-obstacles is shown in Fig. 4.

2.2.2 Maneuvering simulation

The simulation environment was implemented considering a low-speed maneuvering model, steering response characteristics, wind disturbance, and artificial noise; the inclusion of all these elements led to a realistic environment. This section describes the models used to generate each of these elements in detail.

First, the maneuvering model used in this study is described. The mathematical maneuvering group (MMG) model [36] for low-speed maneuvering was used. The submodels that constitute the used MMG model are described in the following. Yoshimura's unified model [37] was used for hull hydrodynamic forces. The resistance coefficient

and linear hydrodynamic derivatives were determined through captive model tests, while the remaining coefficients were estimated using empirical formulae [37]. For rudder and propeller forces, Kang's model [38] was used. The thrust coefficients of the propeller were obtained from the propeller open test, and for the remaining coefficients, the VLCC's coefficients given by Kang [38] were applied. The thruster force was modeled based on Kobayashi's model [39]. The thrust coefficient was obtained experimentally. The relation between the reduction of thruster force and ship speed was represented by a second-order polynomial function with respect to the Froude number, and the coefficients were determined through captive model tests. For wind forces, Fujiwara's regression model [40] was used. The coefficients of this model were determined from the hull and superstructure geometry of the subject ship. The used MMG model is denoted as follows:

$$\dot{x} = f_{\mathrm{MMG}}(x, u, w) \ , \tag{6}$$

where the overdot, $\dot{\ }$, denotes the derivative with respect to time, t.

Second, the response characteristics of the actuator state, u, are described. The rudder steering system with which the subject model ship was equipped has the characteristic of approaching a command value at a constant speed. Therefore, the response characteristics of rudder steering were modeled using the following expressions,

$$\begin{cases} \dot{\delta}_{\mathrm{p}} = K \cdot \mathrm{sign}\left(\delta_{\mathrm{p,cmd}} - \delta_{\mathrm{p}}\right) \\ \dot{\delta}_{\mathrm{s}} = K \cdot \mathrm{sign}\left(\delta_{\mathrm{s,cmd}} - \delta_{\mathrm{s}}\right) \ , \end{cases} \tag{7}$$

where K is the rudder steering speed and is determined to be 20 deg./s. This value is taken based on measurements of the actual model ship system. The response characteristics of the bow thruster revolution numbers were neglected. The bow thruster revolution numbers are given as $n_{\mathrm{bt}} = n_{\mathrm{bt,cmd}}$.

The wind process was generated using the method proposed by Maki et al. [41]. In this method, one-dimensional filter equation using the wind speed spectrum approximated via Hino's spectrum [42] is numerically solved using Euler-Maruyama's method. This method can generate the wind process if the time-averaged true wind speed and direction are given. Here, the time-averaged true wind speed and direction are defined as $\bar{U}_{\mathrm{T}}$ and γ_{T}, respectively. These variables are stochastically determined for each episode.

The maneuvering model described in Eq. 6 was solved numerically using the Runge–Kutta method, and the response characteristics described in Eq. 7 were solved analytically. Here, the time step of the numerical solutions is defined as Δt_{sim}; this time step was set to be 0.1 s, which is shorter than the time step Δt. Here, the initial value of

the vessel state is given such that the tracking error with the desired trajectory follows a uniform distribution over the interval listed in Table 4. The initial value of the actuator state is given as (0.0, 0.0, 0.0, 0.0).

Furthermore, process noise was added to the MMG model, and observation noise was added to the parameter describing the vessel state. The process noise and the observation noise were defined to follow a normal distribution, and their covariance matrices are represented by $\boldsymbol{\Sigma}_{\mathrm{sys}} \in \mathbb{R}^{6\times6}$ and $\boldsymbol{\Sigma}_{\mathrm{obs}} \in \mathbb{R}^{6\times6}$, respectively. The covariance matrix of the observed noise was determined by the nominal observation accuracy, which was described in the equipment specifications.

2.3 Input of the controller

This section describes the function f_s used in Eq. 3. The trajectory tracking controller makes decisions depending on the vessel state, tracking error, and geometric relationship between the vessel and the obstacles. In the following, these elements determining the state, s, are described.

The tracking error is defined as the error between the vessel pose vector, $\boldsymbol{\eta}$, and the desired state, r; this value is based on ship-fixed coordinate system $\mathrm{o} - xy$, and is represented as follows:

$$\boldsymbol{e}_{i,j} = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} (\boldsymbol{r}_i - \boldsymbol{\eta}_j) \ , \tag{8}$$

where i and j represent the time step of the desired state and the vessel pose, respectively. The geometric relationship between vessel and obstacle is defined as the error between vessel positions and the obstacle point nearest to the desired positions; this parameter can be written as,

$$\tilde{\boldsymbol{e}}_{i,j} = \begin{bmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{bmatrix} \left(\boldsymbol{o}_{\mathrm{near},i} - \tilde{\boldsymbol{\eta}}_j\right) \ , \tag{9}$$

where $\tilde{\boldsymbol{\eta}}_j$ is defined as the position component of $\boldsymbol{\eta}_j$ and $\boldsymbol{o}_{\mathrm{near},i}$ is defined as,

$$\boldsymbol{o}_{\mathrm{near},i} = \underset{\boldsymbol{o}\in\mathcal{O}}{\mathrm{argmin}} \|\boldsymbol{o} - \tilde{\boldsymbol{r}}_i\| \ , \tag{10}$$

where $\tilde{\boldsymbol{r}}_i$ denotes the position component of $\boldsymbol{r}_i$.

In this study, s_k is defined to consist of the tracking error, $\boldsymbol{e}_{i,j}$, the distance to the static obstacle, $\tilde{\boldsymbol{e}}_{i,j}$, the vessel velocity, $\boldsymbol{v}_k$ and the actuator state, $\boldsymbol{u}_k$. s_k is thus expressed as,

$$\begin{aligned} s_k = \Big(&\boldsymbol{e}_{k,k}^{\mathsf{T}}, \ \boldsymbol{e}_{k+\frac{T_1}{\Delta t},k}^{\mathsf{T}}, \ \boldsymbol{e}_{k+\frac{T_2}{\Delta t},k}^{\mathsf{T}}, \ \cdots, \\ &\tilde{\boldsymbol{e}}_{k,k}^{\mathsf{T}}, \ \tilde{\boldsymbol{e}}_{k+\frac{T_1}{\Delta t},k}^{\mathsf{T}}, \ \tilde{\boldsymbol{e}}_{k+\frac{T_2}{\Delta t},k}^{\mathsf{T}}, \ \cdots, \ \boldsymbol{v}_k^{\mathsf{T}}, \ \boldsymbol{u}_k^{\mathsf{T}} \Big)^{\mathsf{T}} \ , \end{aligned} \tag{11}$$

where $T_1, T_2, \ldots$ represent arbitrary time steps that are assumed to be positive multiples of Δt. Thus, this study also considers the error between the current vessel state and the future desired state as an input of the controller. Therefore, the function $\boldsymbol{f}_s$ is defined via the use of Eqs. 8–11

2.4 Reward function

This section describes the reward function used in this study. The reward function was designed to obtain a trajectory tracking controller that minimizes tracking errors and controls effort. Furthermore, we propose a method to preferentially minimize tracking errors that may lead to a collision with static obstacles.

One measure of tracking error is the error norm of the vessel pose state, $\boldsymbol{\eta}$, and desired state, $\boldsymbol{r}$. However, this measure requires the time-consuming task of adjusting various weights due to the units being different for the information regarding the position and heading angles. Therefore, the measures of the tracking error were defined based on the error of bow and stern positions that can be expressed as,

$$\begin{cases} e_{\text{bow},k} = \|\boldsymbol{g}_{\text{bow}}(\boldsymbol{\eta}_k) - \boldsymbol{g}_{\text{bow}}(\boldsymbol{r}_k)\| \\ e_{\text{stern},k} = \|\boldsymbol{g}_{\text{stern}}(\boldsymbol{\eta}_k) - \boldsymbol{g}_{\text{stern}}(\boldsymbol{r}_k)\| \end{cases}, \tag{12}$$

where $\boldsymbol{g}_{\text{bow}} \in \mathbb{R}^2$ and $\boldsymbol{g}_{\text{stern}} \in \mathbb{R}^2$ denote the bow and stern positions, respectively; these variables are defined as,

$$\begin{cases} \boldsymbol{g}_{\text{bow}}(\boldsymbol{\eta}) = \left(x_0 + \frac{1}{2}L\cos\psi, \ y_0 + \frac{1}{2}L\sin\psi \right)^\top \\ \boldsymbol{g}_{\text{stern}}(\boldsymbol{\eta}) = \left(x_0 - \frac{1}{2}L\cos\psi, \ y_0 - \frac{1}{2}L\sin\psi \right)^\top \end{cases}. \tag{13}$$

We also define the measure of the distance between the vessel and the static obstacles. Since the purpose of berthing is to get the vessel close to the berth, a measure that prevents the vessel from approaching the berthing point is undesirable. For this reason, a measure that has no trade-off between reducing tracking errors and avoiding the collision with the obstacles is defined here.

In this study, we introduced an imaginary obstacle line shown in Fig. 5; this line passes through $\boldsymbol{o}_{\text{near},k}$, and the angle of inclination of this line is equal to the desired heading angle. The normal vector on the $x_0 y_0$ plane orthogonal to the imaginary obstacle line was also introduced. This normal vector is defined as $\boldsymbol{n}_k \in \mathbb{R}^2$, and can be calculated as according to,

$$\begin{bmatrix} \boldsymbol{n}_k \\ 0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{g}_{\text{bow}}(\boldsymbol{r}_k) - \boldsymbol{g}_{\text{stern}}(\boldsymbol{r}_k) \\ 0 \end{bmatrix} \times \boldsymbol{e}_z, \tag{14}$$
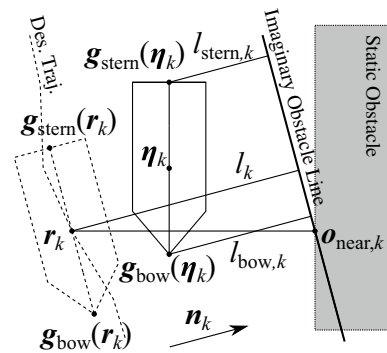
Fig. 5 Illustration of the relationship between the vessel and static obstacle

where $\boldsymbol{e}_z$ is a unit vector orthogonal to the $x_0 y_0$ plane, and this vector was $(0, 0, 1)^\top$. Here, the distance between the position of the desired state and the imaginary obstacle line was expressed as,

$$l_k = \left| \frac{\boldsymbol{n}_k \cdot (\tilde{\boldsymbol{r}}_k - \boldsymbol{o}_{\text{near},k})}{\|\boldsymbol{n}_k\|} \right|. \tag{15}$$

Besides, the distances from the actual bow and stern positions to the imaginary obstacle line were expressed as follows:

$$\begin{cases} l_{\text{bow},k} = \left| \dfrac{\boldsymbol{n}_k \cdot (\boldsymbol{g}_{\text{bow}}(\boldsymbol{\eta}_k) - \boldsymbol{o}_{\text{near},k})}{\|\boldsymbol{n}_k\|} \right| \\ l_{\text{stern},k} = \left| \dfrac{\boldsymbol{n}_k \cdot (\boldsymbol{g}_{\text{stern}}(\boldsymbol{\eta}_k) - \boldsymbol{o}_{\text{near},k})}{\|\boldsymbol{n}_k\|} \right| \end{cases}, \tag{16}$$

Then, the indicators that increase as the vessel approaches the obstacles compared to the desired position can be expressed as $l_k - l_{\text{bow},k}$ and $l_k - l_{\text{stern},k}$.

Thus, the measures of nearness between the vessel and the static obstacles based on the desired position were defined as follows:

$$\begin{cases} c_{\text{bow},k} = \max\left\{ 0, \dfrac{l_k - l_{\text{bow},k}}{l_k} \right\} \\ c_{\text{stern},k} = \max\left\{ 0, \dfrac{l_k - l_{\text{stern},k}}{l_k} \right\} \end{cases}. \tag{17}$$

Note that the cases where the vessel position is farther from the obstacles than the desired position were ignored since we focused on only reducing the tracking errors that increase the probability of collisions.

Moreover, the tolerances of the measures described in Eqs. 12 and 17 were introduced. These tolerances are defined as,

$$\begin{cases} e_{\mathrm{tol},k} = (e_0 - e_\infty) \exp(-b_\mathrm{e} t_k) + e_\infty \\ c_{\mathrm{tol},k} = (c_0 - c_\infty) \exp(-b_\mathrm{c} t_k) + c_\infty \end{cases}, \tag{18}$$

where e_0 and c_0 are the initial tolerances, e_∞ and c_∞ are the tolerances when the time of the episode elapses infinitely, and b_e and b_c are the attenuation factor of the tolerances.

If the measures given in Eqs. 12 and 17 exceed these tolerances, no further rewards are given and the episode is stopped. In other words, the episode is terminated if one of the following inequalities is not satisfied:

$$\begin{cases} e_{\mathrm{bow},k} < e_{\mathrm{tol},k} \\ e_{\mathrm{stern},k} < e_{\mathrm{tol},k} \\ c_{\mathrm{bow},k} < c_{\mathrm{tol},k} \\ c_{\mathrm{stern},k} < c_{\mathrm{tol},k} \end{cases}. \tag{19}$$

If a collision occurs, the episode is terminated. The collision detection area was considered to be equivalent to the area defined in Sect. 2.2.1. If this area contacts an obstacle, the episode is terminated.

The reward function can thus be defined as follows:

$$\begin{aligned} R(t_k, s_k, a_k, \mathscr{R}, \mathscr{O}) &= \frac{e_{\mathrm{tol},k} - e_{\mathrm{bow},k}}{e_{\mathrm{tol},k}} + \frac{e_{\mathrm{tol},k} - e_{\mathrm{stern},k}}{e_{\mathrm{tol},k}} \\ &+ \frac{c_{\mathrm{tol},k} - c_{\mathrm{bow},k}}{c_{\mathrm{tol},k}} + \frac{c_{\mathrm{tol},k} - c_{\mathrm{stern},k}}{c_{\mathrm{tol},k}} - \lambda \sum_{i=1}^{4} \left(\frac{u_{\mathrm{cmd},i} - u_{\mathrm{c},i}}{u_{\mathrm{std},i}} \right)^2, \end{aligned} \tag{20}$$

where λ is a positive constant value, $u_{\mathrm{c},i}$ represents the control input with the lowest control effort, and $u_{\mathrm{std},i}$ is a constant that normalizes each variable. Here, if λ is large, the reward becomes negative. The controller may obtain more rewards when the episode is terminated early in such a case. For this reason, λ is assumed to be a small value, and we defined as $\lambda = 1/300$ in numerical experiments.

3 Application to berthing maneuvers

In this paper, the proposed methods were evaluated in terms of tracking the berthing trajectory. This section describes the generation method of the berthing trajectories and the development of tracking control methods for the berthing trajectories. The target harbor of this study is Inukai pond, which exists at Osaka University. This port geometry was shown in Fig. 6.

3.1 Trajectory planning

In this study, the desired trajectories for berthing maneuvers were generated according to the trajectory planning method proposed by Miyauchi et al. [10]. In this algorithm, the trajectory planning of the berthing was modeled as a time-minimizing problem, and the spatial constraint and the terminal condition are embedded in the objective function. This problem is solved by the framework of optimal control theory [43, 44] with the use of a covariance matrix adaptation evolution strategy (CMA-ES) (see, for example, Ref. [45]).

Here, the obtained trajectory is defined as,

$$\mathscr{R}' = \{\boldsymbol{r}'_{k'}\}_{k'=0,\dots,N'-1}, \tag{21}$$

where k' is discrete-time steps, and N' is the total number of time steps. Note that the new subscript k' is introduced since the time step of the obtained trajectory does not necessarily equal that of the desired trajectory generated in training. The former is determined equally with the decision-making time step of the RL algorithm, while the latter is determined by trajectory planning. The time at which the k'-th step occurs is defined as $t_{k'}$, and the time step between $t_{k'}$ and $t_{k'+1}$ is defined as $\Delta t'$.

We prepared 44 trajectories with 4 different terminal conditions and 11 different initial conditions; these conditions are listed in Table 5. The target poses of terminal conditions were defined at a point $1.5B$ away from the berth, and have different heading angles. The vessel pose of the initial condition was set at a distance of about $8L$ from the target poses. The vessel speed of the initial condition was determined as the speed which can be reduced sufficiently before reaching the target point. The tolerances of the terminal condition are determined based on the previous paper [10] In the trajectory planning, the limits of actuator commands were changed from those listed in Table 2 to generate the trajectory with enough margins of control forces. The limit of the actuator states used is
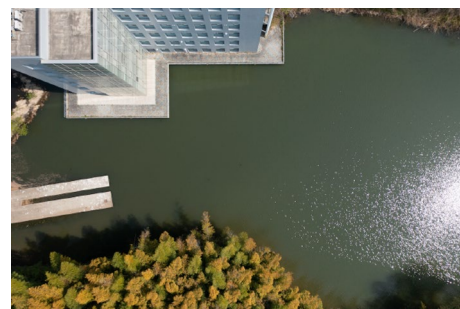

Fig. 6 Port geometry of experimental pond

Table 5 Initial and terminal conditions, and tolerances used in trajectory planning

$\boldsymbol{\eta}_{\mathrm{init},i}$	$(8L \text{ (m)}, i - 5.0 \text{ (m)}, \pi \text{ (rad)}), \quad i = 0, 1, \dots, 10$
$\boldsymbol{v}_{\mathrm{init}}$	$(0.291 \text{ (m/s)}, 0.00 \text{ (m/s)}, 0.00 \text{ (rad/s)})$
$\boldsymbol{\eta}_{\mathrm{des},1}$	$(-3.00 \text{ (m)}, -1.5B \text{ (m)}, \pi \text{ (rad)})$
$\boldsymbol{\eta}_{\mathrm{des},2}$	$(-3.00 \text{ (m)}, -1.5B \text{ (m)}, 0.0 \text{ (rad)})$
$\boldsymbol{\eta}_{\mathrm{des},3}$	$\left(1.5B \text{ (m)}, 3.00 \text{ (m)}, \frac{3\pi}{2} \text{ (rad)}\right)$
$\boldsymbol{\eta}_{\mathrm{des},4}$	$\left(1.5B \text{ (m)}, 3.00 \text{ (m)}, \frac{\pi}{2} \text{ (rad)}\right)$
$\boldsymbol{v}_{\mathrm{des}}$	$(0.0 \text{ (m/s)}, 0.0 \text{ (m/s)}, 0.0 \text{ (rad/s)})$
$\boldsymbol{\eta}_{\mathrm{tol}}$	$(0.02 \text{ (m)}, 0.02 \text{ (m)}, 0.0175 \text{ (rad)})$
$\boldsymbol{v}_{\mathrm{tol}}$	$(0.0141 \text{ (m/s)}, 0.0141 \text{ (m/s)}, 0.0094 \text{ (rad/s)})$

Table 6 Limit of actuator state variables used in trajectory planning

Item	Lower limit	Upper limit
δ_{p}	−105 (deg.)	−45 (deg.)
δ_{s}	45 (deg.)	105 (deg.)
n_{p}	10 (rps)	10 (rps)
n_{bt}	−15 (rps)	15 (rps)

shown in Table 6. This idea has been proposed by Kose et al. [46]. The obtained trajectories are shown in Fig. 7.

3.2 Selection of the desired state

We describe a selection method of the desired pose vector from the desired trajectory. In trajectory tracking, the desired state is selected according to the desired trajectory and does not depend on the current state. When there is a considerable error between the current state and the desired state, the velocity of the vessel deviates significantly from the velocity assumed on the desired trajectory in order to reduce the error. Consequently, this may lead to an undesired increase in vessel speed. In a berthing maneuver, the tracking control must not exceed the desired vessel speed. This is because the vessel may be unable to stop at the berthing point from a higher vessel speed.

Therefore, the desired pose vector was chosen from the desired trajectory according to the vessel position. The distance between the current state $\boldsymbol{\eta}_k$ and a desired state $\boldsymbol{r}'_{k'}$ are calculated as follows:

$$L(k', k) = \left(\boldsymbol{r}'_{k'} - \boldsymbol{\eta}_k\right)^{\top} W\left(\boldsymbol{r}'_{k'} - \boldsymbol{\eta}_k\right) , \tag{22}$$
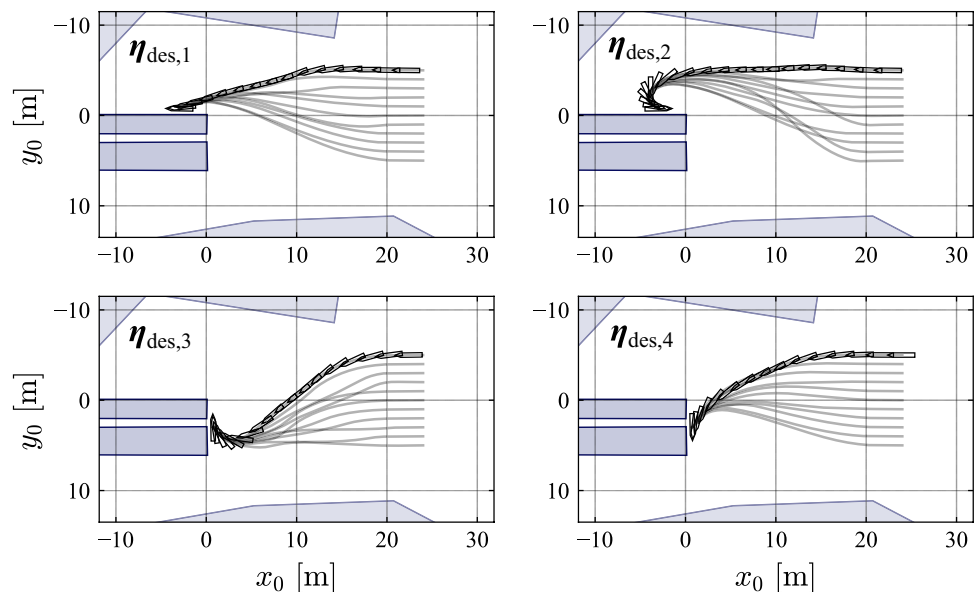
where $W = \mathrm{diag}(1, 1, 0)$. Thus, the error of the heading angle is ignored. Let's assume that $\boldsymbol{r}'_{i_k}$ denotes the nearest state in the desired trajectory from the current state $\boldsymbol{\eta}_k$. Then, i_k is selected by

$$i_k = \operatorname*{argmin}_{k' \in K_k} \left\{L(k', k)\right\} , \tag{23}$$

where K_k denotes the search range. This range is determined as follows:

$$K_k \equiv \begin{cases} \{0, 1, \dots, N' - 1\} & (k = 0) \\ \{i_{k-1} + 1, \dots, i_{k-1} + I\} & (k > 0) \end{cases} , \tag{24}$$

where I denotes a constant and should be defined to avoid shortcuts when desired trajectories intersect. In summary, the desired pose vectors of the control input are selected as follows:

Fig. 7 Desired trajectories generated by the trajectory planning procedure

$$
\begin{cases}
\boldsymbol{r}_k = \boldsymbol{r}'_{i_k} \\
\boldsymbol{r}_{k+\frac{T_1}{\Delta t}} = \boldsymbol{r}'_{i_k+\frac{T_1}{\Delta t'}} \\
\boldsymbol{r}_{k+\frac{T_2}{\Delta t}} = \boldsymbol{r}'_{i_k+\frac{T_2}{\Delta t'}} \\
\quad\vdots
\end{cases}
\tag{25}
$$

Note that $T_1, T_2, \dots$ must also be positive multiples of $\Delta t'$.

3.3 Pseudo-obstacles in the target harbor

Tracking performance is degraded in situations that are different from those experienced in training due to the characteristics of NNs. In other words, if the input of controller $\boldsymbol{s}_k$ that has not occurred in training is entered into the policy function represented by the NNs, the output of the policy function is unpredictable.

In the target harbor, a situation may be encountered that did not occur in the training. For instance, in the target harbor, the value of $\tilde{\boldsymbol{e}}_{k,k}, \tilde{\boldsymbol{e}}_{k+\frac{T_1}{\Delta t},k}, \tilde{\boldsymbol{e}}_{k+\frac{T_2}{\Delta t},k}, \dots$ included in Eq. 11 is larger than that in training. This is because the obstacles of the target harbor have a larger space than the pseudo-obstacles generated in training. Therefore, a pseudo-obstacle was also generated in the target harbor to prevent tracking performance degradation.

The method described in Sect. 2.2.1 would lead to the generation of pseudo-obstacles that would significantly affect berthing maneuvering. Therefore, the ellipse area that was used to represent the vessel shape was changed to a circular area whose radius is $1.9L$. Since generated pseudo-obstacles are dummies, collisions with pseudo-obstacles are ignored. An example generated by the method described here is shown in Fig. 8.
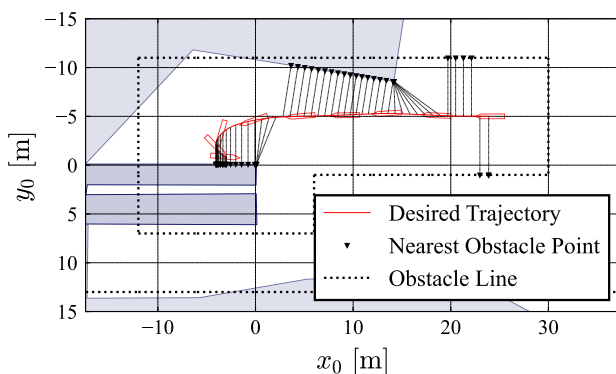
4 Results

This section presents the training results of the proposed method and the results related to the berthing maneuver. We present the tracking results of the berthing trajectories obtained from both simulation and model experiments.

In this study, for comparison of proposed methods, we prepared a trajectory tracking controller minimizing the tracking error and control effort in the absence of static obstacles. Here, $\tilde{\boldsymbol{e}}_{k,k}, \tilde{\boldsymbol{e}}_{k+\frac{T_1}{\Delta t},k}, \tilde{\boldsymbol{e}}_{k+\frac{T_2}{\Delta t},k}, \dots$ included in Eq. 11 are set to zero. The third and fourth terms of the proposed reward function described in Eq. 20 is also set to zero. This trajectory tracking controller is referred to as Ctrl-w/o-OBST, whereas the trajectory tracking controller obtained by the proposed method was Ctrl-w/-OBST.

The training results of Ctrl-w/o-OBST and Ctrl-w/-OBST are shown in Sect. 4.1, and the tracking results of the berthing trajectories using Ctrl-w/o-OBST and Ctrl-w/-OBST are shown in Sect. 4.2.

4.1 Training results of trajectory tracking controller

This section shows the training result of the proposed method. Ctrl-w/-OBST and Ctrl-w/o-OBST were trained five times each. The training was undertaken for a total of 3.0×10^7 s of simulation time.

The parameters used during training are summarized here. The environmental parameters are listed in Table 7, and the parameters describing the tracking controller and the NNs are listed in Tables 8 and 9, respectively. Here, the output dimension of the policy function is different from the dimension of the actuator command because the propeller revolution number is constant. The parameters of the objective function and the reward function are listed in Table 10. Here, the values of e_0 are set to be almost twice as large

Fig. 8 A sample of the generated pseudo-obstacles in the target harbor

Table 7 Parameters describing the training environment

$\bar{U}_{\mathrm{T}}$	Given by Weibull distribution whose shape and scale parameter are 2.0 and 1.0
$\bar{\gamma}_{\mathrm{T}}$	Given by uniform distribution whose interval is $[0, 2\pi)$
$\boldsymbol{\Sigma}_{\mathrm{sys}}$	$\mathrm{diag}\big(0.0, 1.0 \times 10^{-4.2}, 0.0, 1.0 \times 10^{-4.2}, 0.0, 1.0 \times 10^{-3.2}\big)$
$\boldsymbol{\Sigma}_{\mathrm{obs}}$	$\mathrm{diag}\big(0.03^2, 0.02^2, 0.03^2, 0.02^2, 0.2^2, 0.2^2\big)$
Δt	5.0 (s)

Table 8 Numerical values used for the parameters present in Eq. 11

T_1	5.0 (s)	T_2	10.0 (s)
T_3	20.0 (s)	T_4	40.0 (s)

Table 9 Details of the NNs architecture

	Units of layer		Activation function	
	Policy	Q function	Policy	Q function
Input	32 (22)	35 (25)	tanh	tanh
Middle 1	256	256	tanh	tanh
Middle 2	256	256	tanh	tanh
Middle 3	256	256	tanh	tanh
output	3	1	Sigmoid	Linear

Values in brackets express the inputs into Ctrl-w/o-OBST

Fig. 9 Average cumulative reward of the 20 episodes used for the evaluation

as the initial tracking error. The values of b_e and b_c are set such that the allowed range is halved after 50 s. Moreover, the hyperparameters of TD3 were the same values as those listed in Table 3 of [35] except batch size, and the batch size was 512.

The parameters of the NNs acquired during the training were saved and then evaluated for all parameters of the NNs. In the evaluation, 20 episodes were simulated in the same environment as was used in the training episodes, and the average cumulative rewards were calculated. The average cumulative rewards and the 95% confidence interval for the five training are shown in Fig. 9. We see that the results do not deviate significantly from the 95% confidence interval. Therefore, the parameter with the highest cumulative reward among the five training was selected as being the most appropriate method. Note that the cumulative rewards of Ctrl-w/-OBST and Ctrl-w/o-OBST cannot be compared directly as the reward functions are different.

4.2 Tracking results related to the berthing trajectories

This section shows the tracking results related to berthing trajectories. Here, a shorter decision-making interval for the evaluation was set to $\Delta t = 1.0$ s, the time step of the berthing trajectory was set to $\Delta t' = 0.2$ s. The constant I included in Eq. 24 is determined as $I = 5$. The simulation and model experiment results are shown in the following sections.

4.2.1 Simulation

The tracking result of Ctrl-w/-OBST and Ctrl-w/o-OBST on the berthing trajectories were compared in simulation. Simulations were terminated when a collision occurred or

when the elapsed time of the simulation reached 250 s. The collision detection was conducted using the detection area defined in Sect. 2.2.1.

The tracking results in no wind conditions are shown in Figs. 10, 11, 12 and 13. Here, the initial error of y-axis was 3.0 m, initial conditions of berthing trajectories were $\boldsymbol{\eta}_{\text{init},0}$.

In the case of Fig. 10, both controllers were able to absorb the initial error and track the desired trajectories. However, the simulation using Ctrl-w/o-OBST was terminated due to collision. The collision occurred around the target pose nearest to the berth, and the error in the heading angle increased just before the collision. The bow thruster generates a starboard side force to reduce y-axis error, and rudder angles were selected to generate astern force to reduce ship speed. In particular, the upper and lower limits of rudder angles ($\delta_p = -105$ deg., $\delta_s = 105$ deg.) were continued to be selected, and the y-axis force was not controlled. As a result, the moment generated by the bow thruster was not canceled by rudders, and the moment caused the error in the heading angle. On the other hand, Ctrl-w/-OBST terminated the simulation without a collision, and the error of the heading angle was kept small.

In the case of Fig. 12, the simulation using Ctrl-w/o-OBST was terminated due to collision. In this case, although the controller got the sway and yaw velocity close to zero, the collision occurred before the velocities reached zero. In other words, the small overshoot of x_0 and ψ occurred and caused the collision. Therefore, in berthing maneuvers, collisions occur due to small tracking errors even if the performance of the tracking controller is high. Besides, in the

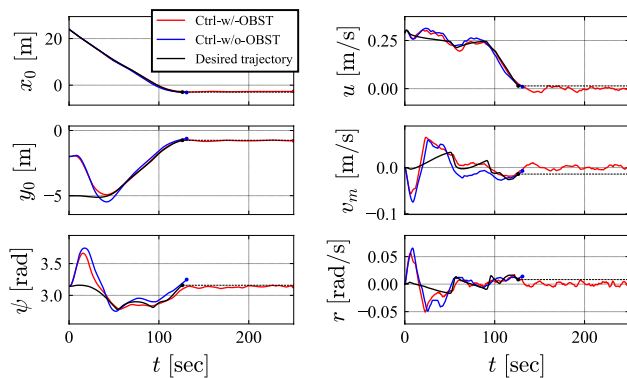**Table 10** Numerical values used for the parameters present in Eqs. 5 and 20

γ	0.99	λ	1/300	$u_{c,1}$	-75	$u_{std,1}$	110
e_0	$2L$	c_0	1.0	$u_{c,2}$	75	$u_{std,2}$	110
e_∞	$B/2$	c_∞	0.5	$u_{c,3}$	10.0	$u_{std,3}$	1.0
b_e	$(\log 2)/50$	b_c	$(\log 2)/50$	$u_{c,4}$	0.0	$u_{std,4}$	30.0
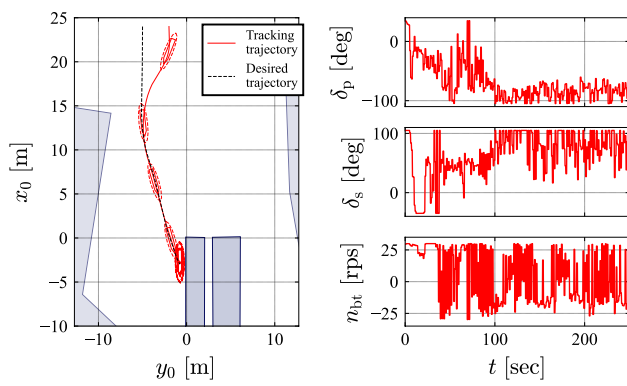
(a) Trajectories and control inputs using Ctrl-w/-OBST
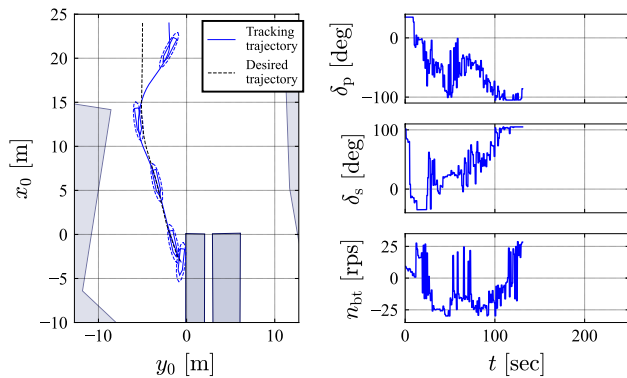
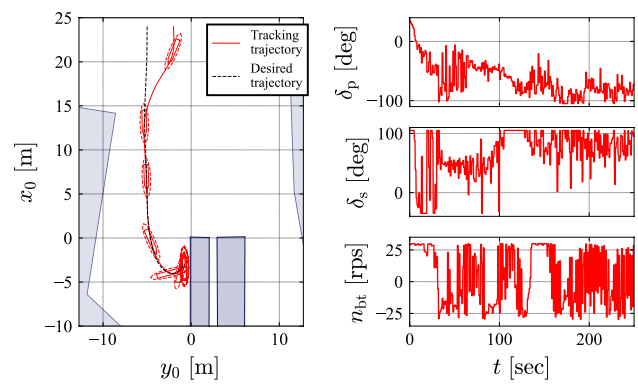(b) Trajectories and control inputs using Ctrl-w/o-OBST
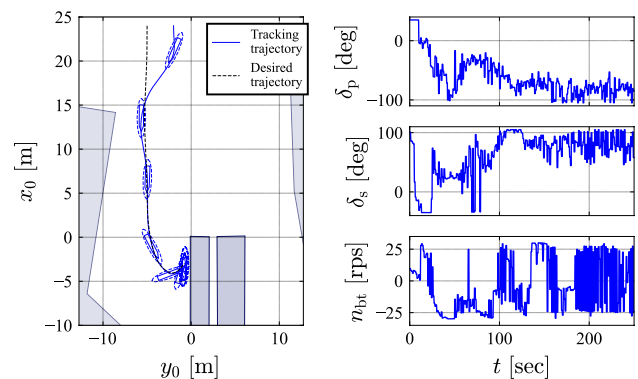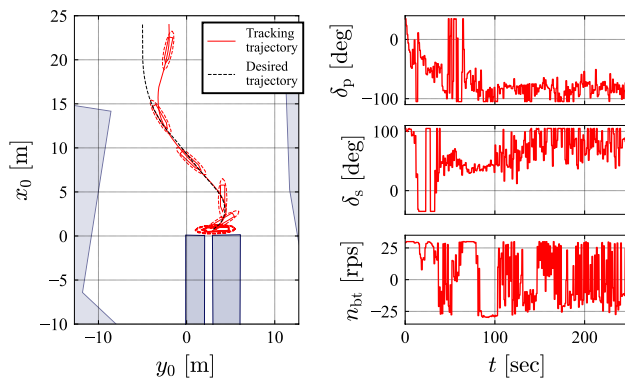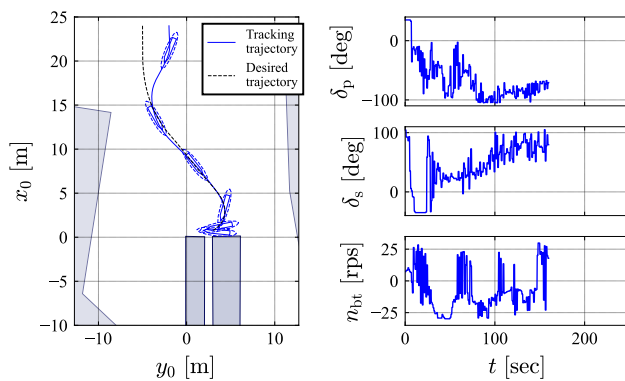

(c) The time series of the pose and velocity vector.

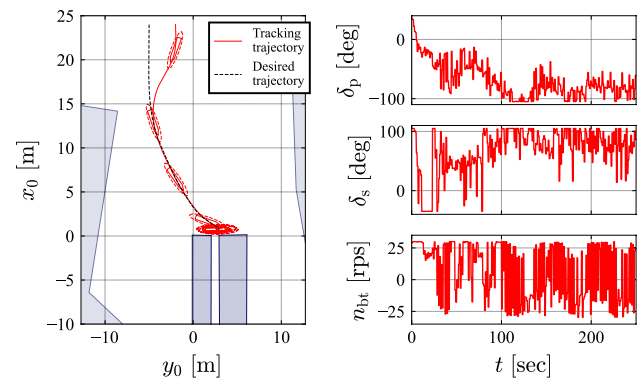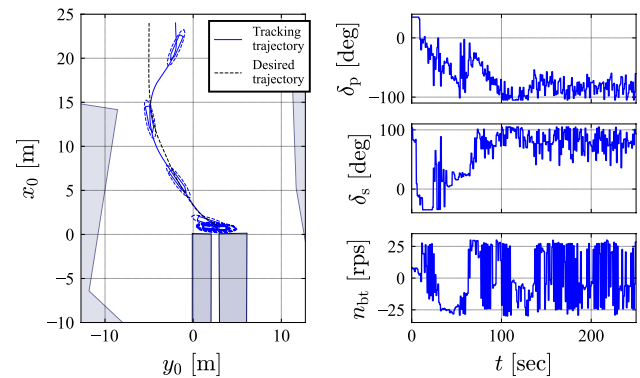**Fig. 10** Tracking results of the berthing trajectory in no wind condition. The terminal condition of the berthing trajectory is $\boldsymbol{\eta}_{\mathrm{des},1}$

(a) Trajectories and control inputs using Ctrl-w/-OBST

(b) Trajectories and control inputs using Ctrl-w/o-OBST

(c) The time series of the pose and velocity vector.

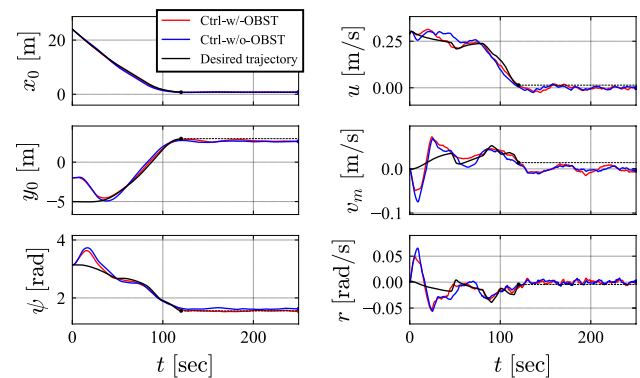**Fig. 11** Tracking results of the berthing trajectory in no wind condition. The terminal condition of the berthing trajectory is $\boldsymbol{\eta}_{\mathrm{des},2}$

cases of Figs. 11 and 13, both controllers were able to track the berthing trajectory, and the simulation was completed without collision.

However, the occurrence of collisions can vary due to factors such as berthing trajectories, wind disturbances, and initial tracking errors. Therefore, we compare Ctrl-w/-OBST and Ctrl-w/o-OBST in terms of collision probability during the berthing maneuvers to demonstrate the effectiveness of

Ctrl-w/-OBST. The collision probability is defined as the probability of colliding with a static obstacle before the end of a given simulation.

The collision probability was computed by fixing the terminal condition of the berthing trajectory and the mean

(a) Trajectories and control inputs using Ctrl-w/-OBST

(b) Trajectories and control inputs using Ctrl-w/o-OBST
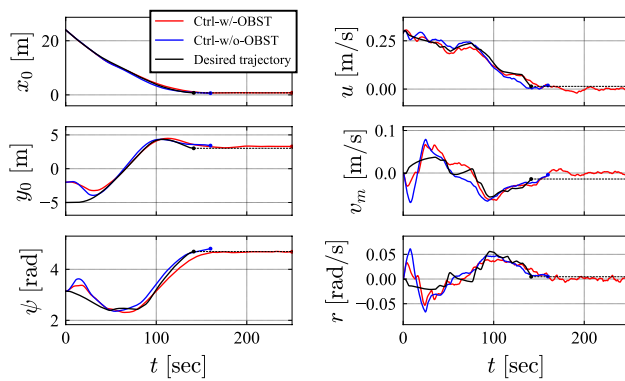
(c) The time series of the pose and velocity vector.

Fig. 12 Tracking results of the berthing trajectory in no wind condition. The terminal condition of the berthing trajectory is $\boldsymbol{\eta}_{\text{des},3}$

(a) Trajectories and control inputs using Ctrl-w/-OBST

(b) Trajectories and control inputs using Ctrl-w/o-OBST

(c) The time series of the pose and velocity vector.

Fig. 13 Tracking results of the berthing trajectory in no wind condition. The terminal condition of the berthing trajectory is $\boldsymbol{\eta}_{\text{des},4}$

wind speed, and varying the initial condition of the berthing trajectory, the initial tracking error, and the mean wind direction. In other words, 100 trials of tracking control for a berthing trajectory were conducted by changing the initial tracking error and the mean wind direction, and the

collision probability was derived by summing the results from 11 initial conditions of the berthing trajectory. The initial tracking error is given to follow a uniform distribution over the intervals listed in the Table 4, and the mean wind direction and noise are determined as shown in Table 7. Due to the potential issue of having an insufficient sample size, collision probabilities were calculated

five times, and the mean and standard deviation for each were calculated. The mean values and standard deviations of the collision probabilities are listed in Table 11. Here, considering the wind pressure and the upper limit of the thrust that the subject ship can generate in bollard conditions, the collision probabilities were computed with mean wind speeds below 1.5 m/s.

In the case where the Ctrl-w/o-OBST was used, we see that the collision probability was relatively high in the berthing trajectories whose terminal conditions are $\eta_{des,1}$ or $\eta_{des,3}$. Collision probabilities were high even at low mean wind speeds because small tracking errors caused collisions, as shown in Figs. 10b and 12b.

In the case where the Ctrl-w/-OBST was used, we see that the collision probability is relatively high in the berthing trajectories whose terminal conditions are $\eta_{des,2}$. One of the reasons for this is that the collision detection area of the bow side contacts an obstacle when turning as shown in Fig. 14. A slight tracking error causes a collision during turning because these berthing trajectories are generated considering the collision detection area described in Sect. 2.2.1. The collision probabilities were therefore recalculated with a semi-major axis of the collision detection ellipse of 0.5L. The obtained collision probabilities are listed in Table 12. It was found under these conditions that the collision probability is similar to those obtained for the other terminal conditions.
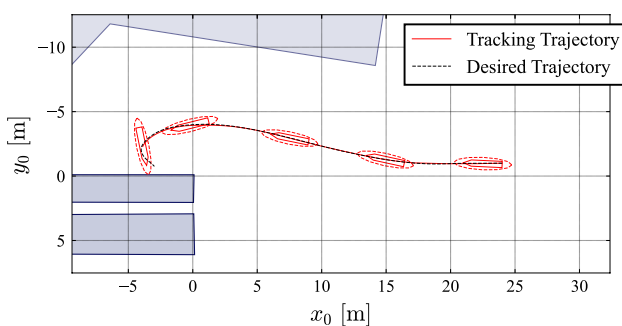
Table 12 The mean values and standard deviations of the collision probabilities

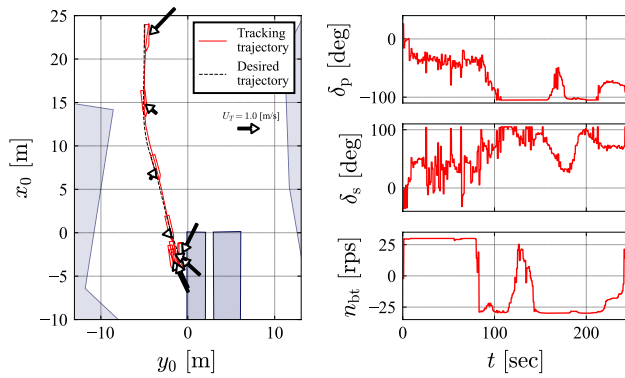| | $\eta_{des,2}$ | |
	w/	w/o
$\bar{U}_T$ (m/s)		
0.0	0.1 ± 0.1	0.0 ± 0.0
0.5	0.5 ± 0.2	0.0 ± 0.0
1.0	17.6 ± 3.2	2.2 ± 0.5
1.5	40.9 ± 4.5	36.6 ± 4.7

A semi-major axis of the collision detection ellipse is changed to 0.5L. Probabilities are expressed as percentages

Comparing the collision probabilities of Ctrl-w/-OBST and Ctrl-w/o-OBST, we can see that the collision probabilities of Ctrl-w/-OBST are smaller than those of Ctrl-w/o-OBST, except when the terminal condition is $\eta_{des,2}$. However, in the case where the terminal condition is $\eta_{des,2}$, the collision probability for Ctrl-w/-OBST is comparable to those for other terminal conditions. Therefore, penalizing tracking errors that lead to collisions with obstacles in the reward function reduces the collision probability in the tracking control of the berthing trajectory.
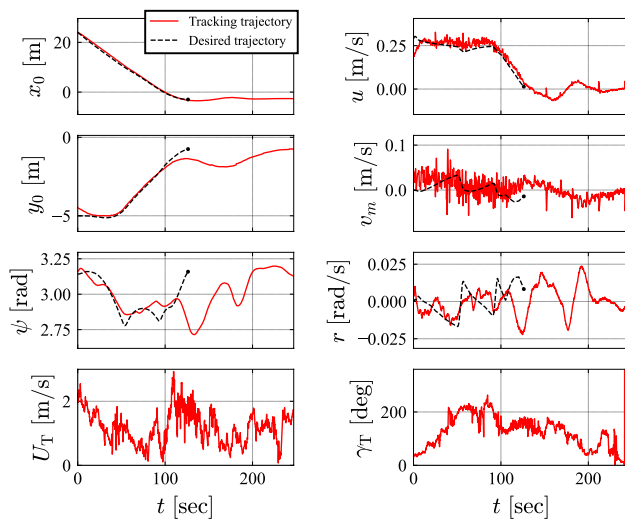
4.2.2 Model experiment

The Ctrl-w/-OBST was evaluated via model experiments. The model experiment was carried out at Inukai pond of Osaka University. The measurement equipment of the model ship closely resembled those employed in a previous study [47]. In summary, x_0 and y_0 were measured through Global Navigation Satellite System (GNSS). ψ and r were measured by Fiber Optical Gyro (FOG). u and v_m are calculated based on the speed over ground, the course over ground, and the heading angle, which were measured by GNSS and FOG, respectively. The berthing trajectories with terminal conditions of $\eta_{des,1}$ and $\eta_{des,2}$ were used. The tracking results related to these berthing trajectories are shown in Figs. 15 and 16, respectively.

In the case of Fig. 15, the model ship was able to track the berthing trajectory during the approach phase before

Fig. 14 A tracking result of the berthing trajectory using the Ctrl-w/-OBST. The terminal condition of the desired trajectory is $\eta_{des,2}$, and mean wind speed, $\bar{U}_T$, is 0.0 m/s

Table 11 The mean values and standard deviations of the collision probabilities

| | $\eta_{des,1}$ | | $\eta_{des,2}$ | | $\eta_{des,3}$ | | $\eta_{des,4}$ | |
	w/	w/o	w/	w/o	w/	w/o	w/	w/o
$\bar{U}_T$ (m/s)								
0.0	7.9 ± 1.1	94.3 ± 0.6	37.8 ± 0.7	1.0 ± 0.2	0.0 ± 0.0	95.4 ± 0.9	0.4 ± 0.1	3.2 ± 0.4
0.5	6.7 ± 2.0	90.8 ± 1.1	39.0 ± 1.8	6.4 ± 0.5	0.0 ± 0.0	92.7 ± 0.7	0.3 ± 0.2	3.1 ± 0.6
1.0	20.2 ± 5.2	72.6 ± 3.9	42.6 ± 4.2	30.0 ± 3.4	1.1 ± 0.8	70.1 ± 3.6	3.7 ± 0.9	6.4 ± 1.4
1.5	45.2 ± 7.3	63.6 ± 5.6	48.7 ± 4.7	48.9 ± 4.4	38.4 ± 4.1	73.2 ± 5.6	37.5 ± 2.8	52.6 ± 4.1

Probabilities are expressed as percentages

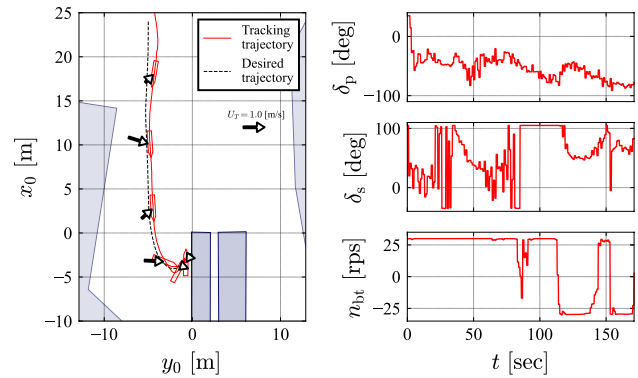(a) Trajectories and the time series of control inputs

(b) The time series of the pose and velocity vector.

Fig. 15 A result of model experiments tracking the berthing trajectory using the Ctrl-w/-OBST algorithm. The terminal condition of the desired trajectory was $\boldsymbol{\eta}_{\mathrm{des},1}$, and wind conditions were $\bar{U}_{\mathrm{T}} = 1.19$ (m/s), $\bar{\gamma}_{\mathrm{T}} = 0.69\pi$ (rad)

(a) Trajectories and the time series of control commands
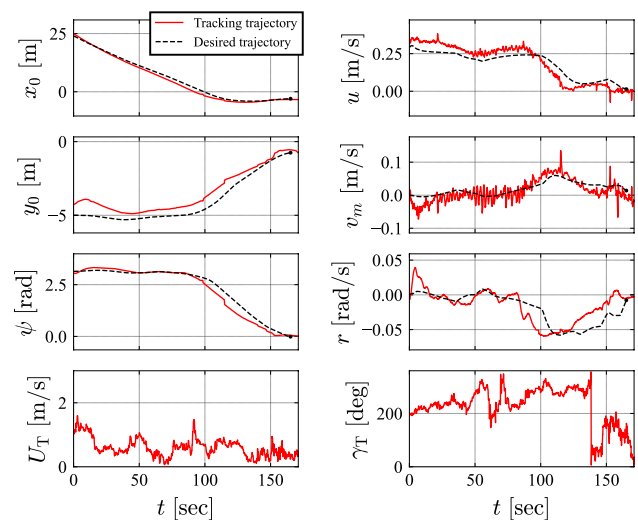
(b) The time series of the vessel pose and velocity.

Fig. 16 A result of model experiments tracking the berthing trajectory using the Ctrl-w/-OBST algorithm. The terminal condition of the desired trajectory was $\boldsymbol{\eta}_{\mathrm{des},2}$, and wind conditions were $\bar{U}_{\mathrm{T}} = 0.60$ (m/s), $\bar{\gamma}_{\mathrm{T}} = -0.62\pi$ (rad)

$t = 100$ s. On the other hand, after $t = 100$ s, the model ship deviated from the berthing trajectory. However, it can be observed that the Ctrl-w/-OBST aims to bring the model ship closer to the desired trajectory by generating an astern force with the VecTwin rudder and a negative sway force with the bow thruster. Therefore, this tracking error can be attributed to an increase in wind speed.

In this case, the model ship reached the berthing point without collision. However, the probability of a collision would increase if the wind direction was different. Therefore, in this wind speed, a decision to stop the berthing maneuvers is required. Further research into the wind speed that permits safe berthing maneuvers is required.

Figure 16 shows that the Ctrl-w/-OBST could track a berthing trajectory, including turning, and reduce the vessel speed to a value close to the allowable velocity, $\boldsymbol{v}_{\mathrm{tol}}$, near

the berthing point. In this case, the wind conditions were relatively calm.

5 Discussion

Simulation results of Sect. 4.2.1 show that the trained controllers were able to track berthing trajectories. This means that the proposed reward function is effective for the tracking controller, and randomly generated trajectories during training were effective enough for tracking controllers to track the prepared berthing trajectories.

The proposed method reduced the collision probability. Therefore, the proposed reward function and the generation method of static pseudo-obstacles are effective to reduce

tracking errors that may cause a collision. However, the collision probability remains too high to be applicable to practical use. In this study, wind disturbance has not been integrated into the controller feedback input nor considered in the trajectory planning. In order to improve the tracking performance and reduce the collision probability of berthing operations, one of our future tasks is to take wind into account in the tracking controller and trajectory planning.

In practical use, the wind disturbance is not controllable. If the force of wind disturbance exceeds the maximum force that the actuators can generate, berthing operations will be difficult no matter how the actuators are controlled. Therefore, another future task is to investigate the threshold of the wind disturbance that makes berthing operations difficult.

Furthermore, to calculate the collision probability, it was evaluated whether a collision would occur before the elapsed time reached 250 s. The effect of fenders was not taken into account. For instance, collisions below a certain speed may be acceptable due to the presence of fenders. Thus, more detailed studies with a definition of successful berthing operations.

In Sect. 4.2.2, the trained controllers were able to track berthing trajectories even in model experiments, and in some cases were able to do so without collision. This study simplifies the modeling by using models proposed in previous studies. Specifically, although the hydrodynamic forces induced by the VecTwin rudder are complex, the model of the VecTwin rudder was a simplified model proposed in previous studies [38]. The process noise was introduced into the environment to compensate for the simplicity of the model. As a result, the trained controllers were able to track berthing trajectories in model experiments. However, Further study of the model to improve tracking performance is one of our future tasks.

In the proposed method, the controller was trained using obstacles consisting of line segments longer than the vessel length. However, actual harbors may have more complex shapes. Therefore, research related to training methods for obstacles consisting of shorter line segments is necessary.

6 Conclusion

This study proposed a training method based on RL for a trajectory tracking controller that reduces the probability of collision with static obstacles. The proposed training method is required to prepare the training environment including the maneuvering model of the subject ship. However, the desired trajectories for the tracking controller were randomly generated from a maneuvering simulation, and the static pseudo-obstacles were generated automatically. Therefore, it is not essential to prepare a large number of desired trajectories and obstacle information for training.

Moreover, this paper summarized the application of the obtained trajectory tracking controller to berthing maneuvers. The target harbor was the Inukai pond of Osaka University, and the berthing trajectories were generated by the trajectory planning method based on a time-minimizing problem. The desired state for the trained controllers was selected according to the berthing trajectories and the current state. Pseudo-obstacles were introduced at the target harbor because the obstacles of the target harbor have a larger space than the pseudo-obstacles generated in training.

To demonstrate the effectiveness of the proposed method, this paper showed the results of both simulations and model experiments related to the tracking of berthing trajectories in a target harbor. In the simulation environment, the trained controllers were able to track berthing trajectories, and the proposed method reduced the collision probability in berthing maneuvers. Furthermore, the trained controllers were able to track berthing trajectories without collisions, even in model experiments.

References

1. Shouji K, Ohtsu K, Mizoguchi S (1992) An automatic berthing study by optimal control techniques. IFAC Proc Vol 25:185

2. Shouji K, Ohtsu K (1992) A study on the optimization of ship maneuvering by optimal control theory (1st report). J Soc Nav Archit Jpn 1992:365

3. Ahmed YA, Hasegawa K (2015) Consistently trained artificial neural network for automatic ship berthing control. TransNav Int J Mar Navig Saf Sea Transp 9(3):417

4. Bitar G, Martinsen AB, Lekkas AM, Breivik M (2020) Trajectory planning and control for automatic docking of ASVs with full-scale experiments this work is supported by the research council of Norway through the project number 269116 as well as through the centres of excellence funding scheme with project number 223254. IFAC-PapersOnLine 53(2), 14488, 21st IFAC World Congress

5. Martinsen AB, Bitar G, Lekkas AM, Gros S (2020) Optimization-based automatic docking and berthing of ASVs using exteroceptive sensors: theory and experiments. IEEE Access 8:204974

6. Sawada R, Hirata K, Kitagawa Y, Saito E, Ueno M, Tanizawa K, Fukuto J (2021) Path following algorithm application to automatic berthing control. J Mar Sci Technol (Jpn) 26(2):541

7. Bitar G, Eriksen BOH, Lekkas AM, Breivik M (2021) Three-phase automatic crossing for a passenger ferry with field trials. In: 2021 European control conference (ECC), pp 2271–2277

8. Shuai Y, Li G, Cheng X, Skulstad R, Xu J, Liu H, Zhang H (2019) An efficient neural-network based approach to automatic ship docking. Ocean Eng 191:106514

9. Li S, Liu J, Negenborn RR, Wu Q (2020) Automatic docking for underactuated ships based on multi-objective nonlinear model predictive control. IEEE Access 8:70044

10. Miyauchi Y, Sawada R, Akimoto Y, Umeda N, Maki A (2022) Optimization on planning of trajectory and control of autonomous berthing and unberthing for the realistic port geometry. Ocean Eng 245:110390

11. Rachman DM, Maki A, Miyauchi Y, Umeda N (2022) Warm-started semionline trajectory planner for ship's automatic docking (berthing). Ocean Eng 252:111127

12. Bartels S, Helling S, Meurer T (2022) Inequality constrained optimal control for rope-assisted asv docking maneuvers. IFAC-PapersOnLine 55(31), 44 (2022). 14th IFAC conference on control applications in marine systems, robotics, and vehicles CAMS

13. Ødven PK, Martinsen AB, Lekkas AM (2022) Static and dynamic multi-obstacle avoidance and docking of ASVs using computational geometry and numerical optimal control. IFAC-PapersOnLine 55(31), 50 (2022). 14th IFAC conference on control applications in marine systems, robotics, and vehicles CAMS

14. Fossen TI (2000) A survey on nonlinear ship control: from theory to practice. IFAC Proc Vol 33(21):1

15. Sørensen AJ (2011) A survey of dynamic positioning control systems. Annu Rev Control 35(1):123

16. Ding F, Wang Y, Wang Y (2011) Trajectory-tracking controller design of underactuated surface vessels. In: OCEANS'11 MTS/IEEE KONA, pp 1–5

17. Zheng H, Negenborn RR, Lodewijks G (2014) Trajectory tracking of autonomous vessels using model predictive control. IFAC Proc Vol 47(3):8812

18. Yang Y, Du J, Liu H, Guo C, Abraham A (2014) A trajectory tracking robust controller of surface vessels with disturbance uncertainties. IEEE Trans Control Syst Technol 22(4):1511

19. Wen G, Ge SS, Chen CLP, Tu F, Wang S (2019) Adaptive tracking control of surface vessel using optimized backstepping technique. IEEE Trans Cybern 49(9):3420

20. Jiang X, Huang L, Peng M, Li Z, Yang K (2022) Nonlinear model predictive control using symbolic computation on autonomous marine surface vehicle. J Mar Sci Technol 27(1):482

21. Rachman DM, Miyauchi Y, Umeda N, Maki A (2021) Feasibility study on the use of evolution strategy: CMA-ES for ship automatic docking problem. In: Proceedings of 1st international conference on the stability and safety of ships and ocean vehicles (2021), STAB &S

22. Cheng Y, Zhang W (2018) Concise deep reinforcement learning obstacle avoidance for underactuated unmanned marine vessels. Neurocomputing 272:63

23. Martinsen AB, Lekkas AM (2018) Straight-path following for underactuated marine vessels using deep reinforcement learning. IFAC-PapersOnLine 51(29):329

24. Martinsen AB, Lekkas AM (2018) Curved path following with deep reinforcement learning: results from three vessel models. In: OCEANS 2018 MTS/IEEE Charleston, pp 1–8

25. Martinsen AB, Lekkas AM, Gros S, Glomsrud JA, Pedersen TA (2020) Reinforcement learning-based tracking control of USVs in varying operational conditions. Front Robot AI 7:32

26. Meyer E, Heiberg A, Rasheed A, San O (2020) Colreg-compliant collision avoidance for unmanned surface vehicle using deep reinforcement learning. IEEE Access 8:165344

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

28. Martinsen AB, Lekkas AM, Gros S (2022) Reinforcement learning-based NMPC for tracking control of ASVs: theory and experiments. Control Eng Pract 120:105024

29. Sutton RS, Barto AG (2018) Reinforcement learning: an introduction, 2nd edn. MIT Press, Cambridge

30. Japan Hamworthy & Co., Ltd. (2022) Steering systems: new VecTwin system. https://www.japanham.com/en/service/new_vectwin.html. Accessed 12 July 2023

31. Hasegawa K, Kang D, Sano M, Nabeshima K (2006) Study on the maneuverability of a large vessel installed with a mariner type Super VecTwin rudder. J Mar Sci Technol 11:88

32. Rachman DM, Aoki Y, Miyauchi Y, Umeda N, Maki A (2023) Experimental low-speed positioning system with VecTwin rudder for automatic docking (berthing). J Mar Sci Technol 28:689–703

33. Silver D, Lever G, Heess N, Degris T, Wierstra D, Riedmiller M (2014) Deterministic policy gradient algorithms. In: Proceedings of the 31st international conference on machine learning, Proceedings of machine learning research, vol 32, ed. by Xing EP, Jebara T (eds) (PMLR, Bejing, China), Proceedings of machine learning research, vol 32, pp 387–395

34. Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, Tassa Y, Silver D, Wierstra D (2016) Continuous control with deep reinforcement learning. In: Bengio Y, LeCun Y (eds) 4th international conference on learning representations, ICLR 2016, San Juan, Puerto Rico, May 2–4, 2016, conference track proceedings

35. Fujimoto S, van Hoof H, Meger D (2018) Addressing function approximation error in actor-critic methods. In: Proceedings of the 35th international conference on machine learning, Proceedings of machine learning research, vol 80, Dy J, Krause A (eds) (PMLR), Proceedings of machine learning research, vol 80, pp 1587–1596

36. Ogawa A, Kasai H (1978) On the mathematical model of manoeuvring motion of ships. Int Shipbuild Prog 25(292):306

37. Yoshimura Y, Nakao I, Ishibashi A (2009) Unified mathematical model for ocean and harbour manoeuvring. In: International conference on marine simulation and ship maneuverability, pp 116–124

38. Kang D, Nagarajan V, Hasegawa K, Sano M (2008) Mathematical model of single-propeller twin-rudder ship. J Mar Sci Technol 13(3):207

39. Kobayashi E (1988) A simulation study on ship manoeuvrability at low speeds. Akishima Laboratory, Ocean Engineering Research

Section, Mitsubishi Heave Industries Ltd. Published in: Mitsubishi Technical Bulletin No. 180

40. Fujiwara T, Ueno M, Nimura T (1998) Estimation of wind forces and moments acting on ships. J Soc Nav Archit Jpn 1998(183):77

41. Maki A, Maruyama Y, Dostal L, Sakai M, Sawada R, Sasa K, Umeda N (2022) Practical method for evaluating wind influence on autonomous ship operations. J Mar Sci Technol 27:1302

42. Hino M (1971) Spectrum of gusty wind. In: Proceedings of the 3rd international conference on wind effects on buildings and structures, Tokyo, Japan, vol 77

43. Maki A, Sakamoto N, Akimoto Y, Nishikawa H, Umeda N (2020) Application of optimal control theory based on the evolution strategy (CMA-ES) to automatic berthing. J Mar Sci Technol 25(1):221

44. Maki A, Akimoto Y, Naoya U (2021) Application of optimal control theory based on the evolution strategy (CMA-ES) to automatic berthing (part: 2). J Mar Sci Technol 26(3):835

45. Sakamoto N, Akimoto Y (2017) Modified box constraint handling for the covariance matrix adaptation evolution strategy. In: Proceedings of the genetic and evolutionary computation conference companion. Association for Computing Machinery, New York, NY, USA, GECCO '17, pp 183–184

46. Kose K, Fukudo J, Sugano K, Akagi S, Harada M (1986) On a computer aided maneuvering system in harbours. J Soc Nav Archit Jpn 1986(160):103

47. Miyauchi Y, Maki A, Umeda N, Rachman DM, Akimoto Y (2022) System parameter exploration of ship maneuvering model for automatic docking/berthing using CMA-ES. J Mar Sci Technol 27:1065