Journal of Marine Science and Technology (2022) 27:772–785
https://doi.org/10.1007/s00773-021-00867-1

ORIGINAL ARTICLE

On neural network identification for low-speed ship maneuvering model

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

Received: 7 July 2021 / Accepted: 21 December 2021 / Published online: 15 January 2022

Abstract
Several studies on ship maneuvering models have been conducted using captive model tests or computational fluid dynamics (CFD) and physical models, such as the maneuvering modeling group (MMG) model. A new system identification method for generating a low-speed maneuvering model using recurrent neural networks (RNNs) and free running model tests is proposed in this study. We especially focus on a low-speed maneuver such as the final phase in berthing to achieve automatic berthing control. Accurate dynamic modeling with minimum modeling error is highly desired to establish a model-based control system. We propose a new loss function that reduces the effect of the noise included in the training data. Besides, we revealed the following facts—an RNN that ignores the memory before a certain time improved the prediction accuracy compared with the "standard" RNN, and the manual random maneuver test was effective in obtaining an accurate berthing maneuver model. In addition, several low-speed free running model tests were performed for the scale model of the M.V. Esso Osaka. As a result, this paper showed that the proposed method using a neural network model could accurately represent low-speed maneuvering motions.

Keywords System identification · Recurrent neural network · Automatic berthing · Manual random maneuver

1 Introduction

In Japan, a shortage in human resources has recently become a critical issue in coastal shipping due to the aging of ship crews [1]. Therefore, the autonomization in domestic shipping is required. To realize this autonomization, automating berthing control is one of the challenges. So far, various studies have been conducted on the automatic berthing problem [2–7]. In addition, we have conducted studies of offline berthing path planning [6, 7] and online berthing controller [8–11]. Since these control methods are model-based controls, their performances can depend on the modeling error of the given state equations. Although the modeling error should be absorbed by the online feedback control method, the smaller the modeling error, the more realistic the berthing path planning and online control can be achieved.

Some significant studies on developing ship maneuvering models have been conducted [12, 13]. The widely used simulation models are based on the outcome of hydrodynamics, and they can be represented by a set of equations of motion with several coefficients to be determined by captive model tests [13] and computational fluid dynamics (CFD) [14]. Dynamical system models can be categorized into two—*white-box* models, where the systems are usually understandable and explicitly described as a form of "standard" equations, and *black-box* models, where the systems do not necessarily have an explicit form of "standard" equations of motion and usually incomprehensible.

Examples of the white-box maneuvering models are the MMG model [13] mainly developed in Japan and the Abkowitz model [12]; they are commonly used in practice. The coefficients included in these models can be

✉ 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

determined through several captive model tests or CFD. Determining the coefficients by the captive model tests in the tanks use common techniques, such as circular motion test (CMT), planer motion mechanism (PMM) test, and so on. One problem is the difference in the Reynolds number between the scale model and full-scale ship. One of the final goals of modeling is to obtain the maneuvering model in full scale to achieve automatic berthing. Therefore, the difference in Reynolds number can deteriorate modeling accuracy in a full-scale ship. Although it can be possible to conduct full-scale computation in CFD, it requires an extremely high computational cost at present. System identification (SI) from the actual ship maneuvers is considered a practical method [15]. So far, several studies on SI for maneuvering models have been conducted [12, 16–22]. However, the tests using the actual vessel are usually too expensive and time-consuming. Therefore, as an alternative, model tests should be used for predicting the performance of the actual vessel.

Despite the abovementioned advantage, SI with white-box models has several drawbacks. One of the ways to enhance the performance of at white-box model is to increase the number of the component inside the model. Particularly, for the MMG model [13], enhancing the modeling based on hydrodynamic insight is essential. In a low-speed berthing maneuver, the flow field around a vessel is apparently complex, so its hydrodynamic modeling is complicated. For instance, in the conventional MMG model, the included parameters, which must implicitly depend on propeller slip ratio or ship speed, e.g., steering resistance reduction factor t_R, rudder force increase factor a_H, are often considered as constant [13, 23]. Moreover, it is essential to switch to a hydrodynamic model depending on the sign of the longitudinal speed and propeller revolution based on tank test results and hydrodynamic insight, which means using numerous conditional branches, i.e., `if` sentences. Therefore, these efforts seem difficult, and it is time-consuming. To achieve these, the dynamical system model becomes more complex. Thus, black-box models are considered suitable for such cases.

Many studies have been conducted on SI for ship maneuvers using black-box models, such as support vector machine [24–27], random forest [28], and neural networks (NNs) [29–35]. In particular, an NN can approximate various functions using a set of the appropriate number of parameters, as shown in the universal approximation theorem (UAT) [36–38], and the gradient can easily be calculated using the backpropagation (BP) method [39]. Among NNs, a recurrent NN (RNN), which considers memory, seems to be suitable for a maneuvering model prediction problem that is categorized into the partially observable system. Moreover, several studies [29–31, 34, 35] have shown that the RNN could be a practical method for predicting maneuvering models.

However, in the abovementioned studies, the following issues are still left as an open problem. The first issue is the choice of the network structure. In previous studies, two types of "RNN" were used—the "standard" RNN which was used in [29, 31, 34, 35], and the RNN that ignores the memory before a certain time, which was used [30]. In this study, although it is unclear whether the employed latter NN is an RNN, we treat it as RNN. Both types of RNN are used for the ship maneuvering model. To the best of our knowledge, no study directly compares these two types of RNN.

The second issue is the objective function in the training process. In the continuous control problem of ship maneuver, the right side of the state equation outputs the instantaneous accelerations with respect to each instantaneous state variable, such as velocities and yaw angular velocity. Therefore, the dynamical system modeling employs an NN is expected to output instantaneous accelerations similar to the conventional ship maneuvering mathematical model. In most previous studies that use NN for the prediction of ship maneuvering dynamical system modeling, instantaneous accelerations, which are outputs of the NN, were directly evaluated by loss function in the optimization process. This evaluation process is considered appropriate for numerically generated training data. However, training data measured in a real environment contain enormous observation noise and uncertainties in the acceleration component. For instance, in our experimental system, accelerations were obtained from the time-differentiation of positions X and Y measured by the global navigation satellite system (GNSS), and apparently the derivative operations are likely to induce noise. Therefore, it is not easy to predict a ship maneuvering model in the actual uncertain environment using existing methods.

The third issue is the target speed and maneuver range of a ship maneuvering model. In previous studies, NNs were not trained and tested for complex maneuvers, such as the final phase maneuvering motion in berthing control. To train an NN model for low-speed manuevering, both combinations of positive/negative propeller revolution per second (RPS) and ship-positive/negative velocity should be included in the training datasets. In almost all SI studies on ship maneuvering models using NNs [29, 31–33], the target is the maneuvering motion with positive speed and positive propeller revolution number, and the maneuver scenarios are mainly "usual maneuver," such as zigzag and turning tests. In [30], the propeller revolution number was constant. However, if the target is a low-speed maneuver, such as the berthing maneuver, the training dataset that only comprises zigzag and turning tests are insufficient, and not only the NN structure but also the acquisition procedure of the training data itself should be analogously reconsidered. This is because to successfully proceed with the training for a low-speed maneuver, it is unavoidable to correct a "better" dataset, including positive/negative

propeller RPS and forward/backward longitudinal speeds. Therefore, it is desired that the combination of states and these inputs should be uniformly distributed.

In this study, an identification method for the low-speed maneuvering model using model test data is investigated. Although the final target of this study is to predict the maneuvering model of the actual ship, this paper deals with the prediction of the maneuvering model of a model ship as the first step of the research. The major contributions of this article are as follows. (i) Proposing a methodology to predict the dynamical system modeling using an NN that is applicable for a low-speed maneuver, such as berthing maneuver. (ii) Revealing that the RNN that ignores the memory before a certain time improves the prediction accuracy compared with the "standard" RNN. (iii) Proposeing a new loss function to reduce the effect of noise using numerical simulation. (iv) Revealing the effectiveness of manual random maneuvering tests to obtain more uniformly distributed combinations of state and input used when a berthing maneuver is predicted using NN.

This article is organized as follows. Section 2 describes the problem settings, an NN architecture, the newly introduced loss function, and a white-box model used for validation—an MMG model. Section 3 describes the subject ship and training data acquisition in our experimental pond. In Sect. 4, the states and trajectories calculated from the trained black-box model are presented, and the prediction performances of the emplayed NN and MMG model are compared for validation. The dependency of the prediction accuracy on the amount of the data is also discussed. Finally, Sect. 5 concludes this study.

2 Mathematical modeling with NN

In this section, we explain the methodology to estimate the maneuvering model using an NN model.

Two coordinate systems are used (Fig. 1). One is an earth-fixed coordinate system $\mathrm{O}-XY$, and the other is a ship-fixed coordinate system $\mathrm{O}-xy$, with the midship as the origin.

A state equation of maneuvering motion assuming the Markov property is described as follows. The inputs of the state equation are the state variable, control variables, and external disturbance. Concerning the external disturbance, the wind is assumed to be measurable. Other disturbances, such as tide, and vortices may also affect the ship's maneuverability, but these properties are regarded as unknown. Meanwhile, the outputs of the state equation are instantaneous acceleration components. Summarizing the above, the usual state equation of maneuvering motion, which is the Markov process is as follows:

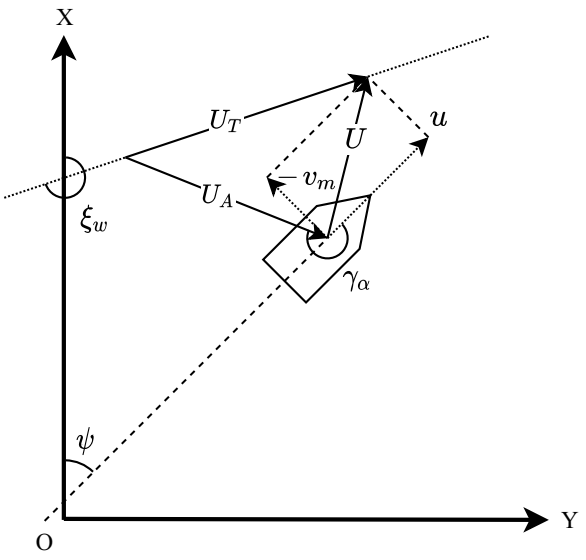

Fig. 1 Coordinate systems

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w}), \tag{1}$$

where the overdot $\dot{}$ denotes the derivative with respect to time t. The state vector $\boldsymbol{x} \in \mathbb{R}^6$, the control vector $\boldsymbol{u} \in \mathbb{R}^2$, and the wind disturbance vector $\boldsymbol{w} \in \mathbb{R}^2$ were defined as follows,

$$\begin{cases} \boldsymbol{x} \equiv \left(X, u, Y, v_m, \psi, r \right)^{\mathrm{T}} \\ \boldsymbol{u} \equiv \left(n, \delta \right)^{\mathrm{T}} \\ \boldsymbol{w} \equiv \left(U_A, \gamma_\alpha \right)^{\mathrm{T}}. \end{cases} \tag{2}$$

Here, X [m] and Y [m] are the ship's position in the earth-fixed coordinate system. The heading (yaw) ψ [rad] is the angle between the earth-fixed coordinate system and the ship-fixed coordinate system. u [m/s] and v [m/s] are the longitudinal (surge) velocity and the lateral (sway) velocity at the center of gravity, respectively. These velocities are defined in the ship-fixed coordinate system. The yaw angular velocity is denoted by r [1/s]. The relation between v and v_m [m/s] is defined as follows:

$$v_m = v - x_G r, \tag{3}$$

where v_m and x_G are the sway velocity at the midship and the distance of the center of gravity from the midship, respectively. The control inputs are the propeller RPS n [1/s] or [rps] and the rudder angle δ [rad]. Meanwhile, the wind disturbance vector comprises of the apparent wind speed U_A [m/s] and the apparent wind direction γ_α [rad]. These apparent wind properties, i.e., U_A [m/s] and γ_α [rad], are calculated from the absolute wind speed U_{T} [m/s],

absolute wind direction ξ_w [rad], longitudinal velocity u, and the lateral velocity v_m as shown in Fig. 1

Besides, for convenience, the state variables $\boldsymbol{x}$ were categorized into two —position and velocity components— as follows:

$$\begin{cases} \boldsymbol{p} \equiv (X, Y, \psi)^{\mathrm{T}} \\ \boldsymbol{v} \equiv \left(u, v_m, r\right)^{\mathrm{T}}. \end{cases} \tag{4}$$

Then, the following relationship between $(X,\ Y,\ \psi)^{\mathrm{T}}$ and $\left(u,\ v_m,\ r\right)^{\mathrm{T}}$ should be satisfied:

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} X \\ Y \\ \psi \end{pmatrix} = \begin{pmatrix} u\cos\psi - v_m \sin\psi \\ u\sin\psi + v_m \cos\psi \\ r \end{pmatrix}. \tag{5}$$

Using the above relationship, it is easy to predict the derivative of the position component $\boldsymbol{p}$ in Eq. (1). Therefore, only the derivative of the velocity component $\boldsymbol{v}$ is the target to be predicted.

Up to this point, for simplicity, we have assumed that the state equation has the Markov property. However, the state equation often violates the Markov property. In such a situation, the output of the state equation depends also on the past, i.e., $\dot{\boldsymbol{x}}$ at time τ depends on $\boldsymbol{x}_{t\le\tau}$, $\boldsymbol{u}_{t\le\tau}$, and $\boldsymbol{w}_{t\le\tau}$. Hence, our target is to estimate $\dot{u}$, $\dot{v}_m$, $\dot{u}$ using the NN, where their inputs are replaced with $\boldsymbol{x}_{t\le\tau}$, $\boldsymbol{u}_{t\le\tau}$, and $\boldsymbol{w}_{t\le\tau}$.

2.1 NN model

A artificial NN [39, 40] is a computational model that imitates the NN of the human brain. As already mentioned, an NN can represent various functions using the set of appropriate parameters, as shown in the UAT [36–38], and the gradient can be easily calculated using the BP method [39]. NNs are widely used in many engineering fields, such as image recognition [41], language processing [42], and anomaly detection [43]. Considering these advantages of NNs, we conducted a study on SI for ship maneuver using an NN.

As stated above, in addition to the state variable $\boldsymbol{x}$, the control variable $\boldsymbol{u}$, and the wind disturbance $\boldsymbol{w}$, there are several unknown, uncertain, and unobservable factors such as the vortices around the hull. The vortices induce the memory effect, so the state equation is no more a Markov process. In this sense, the RNN, which considers memory, is appropriate for this partially observable and non-Markov system. From such a perspective, some studies on the prediction of maneuvering models using RNNs have been conducted [29–31], and their effectiveness has been shown. Therefore, we apply the RNN to a low-speed maneuvering prediction problem.

In this study, we assumed that the acceleration vector $\boldsymbol{a} \equiv (\dot{u}, \dot{v}_m, \dot{r})$, which is predicted as an output of the NN, is independent of the positions and yaw angle vector $\boldsymbol{p}$. Therefore, $\boldsymbol{v}$, $\boldsymbol{u}$, and $\boldsymbol{w}'$ are considered the input features. Concerning $\boldsymbol{w}'$ in the input, directly using the apparent wind direction γ_α is appropriate for an input feature because of its sudden jump of value at 0 or 2π. The apparent wind speed and direction (U_A, γ_α) are converted into the apparent wind speed vector $\boldsymbol{w}'$ on the ship-fixed coordinate system as follows:

$$\boldsymbol{w}' \equiv \left(U_A \cos\gamma_\alpha, U_A \sin\gamma_\alpha\right)^{\mathrm{T}}. \tag{6}$$

As mentioned above, two types of RNNs have been used in previous studies. One is the "standard" RNN [29, 31, 34, 35], which considers memory by substituting the output of a previous step into the input of the present step; notably, all previous memories are considered. The other one is the RNN [30] in which a limited number of previous memory is explicitly used. In this RNN, the state variables in the recent past are considered, and the states in the distant past are ignored. Generally, states of the recent past strongly affect the present motion. From such a perspective, explicitly considering the recent past states could be beneficial. Therefore, in this study, both models are employed and compared.

We now explain two types of networks used in this paper. First, the "standard" RNN is formulated as follows:

$$\begin{cases} \boldsymbol{z}_1(t_i) = \tanh(\boldsymbol{W}_{x0}^{\mathrm{T}}\boldsymbol{v}(t_i) + \boldsymbol{W}_{u0}^{\mathrm{T}}\boldsymbol{u}(t_i) + \\ \qquad\qquad \boldsymbol{W}_{w0}^{\mathrm{T}}\boldsymbol{w}'(t_i) + \boldsymbol{W}_{r0}^{\mathrm{T}}\boldsymbol{z}_1(t_{i-1}) + \boldsymbol{b}_0) - \tanh(\boldsymbol{b}_0) \\ \boldsymbol{z}_2(t_i) = \tanh(\boldsymbol{W}_1^{\mathrm{T}}\boldsymbol{z}_1(t_i) + \boldsymbol{b}_1) - \tanh(\boldsymbol{b}_1) \\ \boldsymbol{z}_3(t_i) = \tanh(\boldsymbol{W}_2^{\mathrm{T}}\boldsymbol{z}_2(t_i) + \boldsymbol{b}_2) - \tanh(\boldsymbol{b}_2) \\ \boldsymbol{a}_{NN}(t_i) = \boldsymbol{W}_3^{\mathrm{T}}\boldsymbol{z}_3(t_i), \end{cases} \tag{7}$$

where t_i is the discretized time and is given by $t_i = t_0 + i\Delta t, (i = 0, 1, 2\ldots)$, with t_0: initial time; Δt: sample period; $\boldsymbol{a}_{NN}$ is the output vector predicting $\boldsymbol{a}$; $\boldsymbol{z}_1$,$\boldsymbol{z}_2$ and $\boldsymbol{z}_3$ are the vectors of latent variables and $\boldsymbol{z}_1(t_0) = \boldsymbol{0}$. $\boldsymbol{W}_{x0} \in \mathbb{R}^{3\times200}$, $\boldsymbol{W}_{u0} \in \mathbb{R}^{2\times200}$, $\boldsymbol{W}_{w0} \in \mathbb{R}^{2\times200}$, $\boldsymbol{W}_{r0} \in \mathbb{R}^{200\times200}$, $\boldsymbol{W}_1 \in \mathbb{R}^{200\times200}$, $\boldsymbol{W}_2 \in \mathbb{R}^{200\times200}$, $\boldsymbol{W}_3 \in \mathbb{R}^{200\times3}$, $\boldsymbol{b}_0 \in \mathbb{R}^{200}$, $\boldsymbol{b}_1 \in \mathbb{R}^{200}$, and $\boldsymbol{b}_2 \in \mathbb{R}^{200}$ are the parameters included in the model; these parameters are represented collectively as $\boldsymbol{\theta}$. Formally, the input-output relation at time t_i is expressed as follows:

$$\boldsymbol{a}_{NN}(t_i) = \boldsymbol{f}_{1NN}(\boldsymbol{v}(t_i), \boldsymbol{u}(t_i), \boldsymbol{w}'(t_i), \boldsymbol{z}_1(t_{i-1}); \boldsymbol{\theta}), \tag{8}$$

where $\boldsymbol{z}_1(t_{i-1})$ plays the role of the memory representing the past, i.e., $\boldsymbol{v}(t_j)$, $\boldsymbol{u}(t_j)$ and $\boldsymbol{w}'(t_j)$ for $j < i$.

The second terms of first, second, and third expressions in Eq. (7), $\tanh(\boldsymbol{b}_k)$ for $k = 0, 1, 2$, are subtracted to impose zero output on the NN model at the origin of input features. This type of NN model was introduced in [44]. The constraints enable us to consider that the accelerations

must be zero in principle without the existence of external forces and vessel speeds.

Second, the NN that explicitly considers the recent past memories of states variables can be expressed as follows:

$$\begin{cases} z_1(t_{i-m+1}) = \tanh(\boldsymbol{W}_{x0}^{\mathrm{T}}\boldsymbol{v}(t_{i-m+1}) + \boldsymbol{W}_{u0}^{\mathrm{T}}\boldsymbol{u}(t_{i-m+1}) \\ \qquad\qquad + \boldsymbol{W}_{w0}^{\mathrm{T}}\boldsymbol{w}'(t_{i-m+1}) + \boldsymbol{b}_0) - \tanh(\boldsymbol{b}_0) \\ z_1(t_{i-m+2}) = \tanh(\boldsymbol{W}_{x0}^{\mathrm{T}}\boldsymbol{v}(t_{i-m+2}) + \boldsymbol{W}_{u0}^{\mathrm{T}}\boldsymbol{u}(t_{i-m+2}) \\ \qquad + \boldsymbol{W}_{w0}^{\mathrm{T}}\boldsymbol{w}'(t_{i-m+2}) + \boldsymbol{W}_{r0}^{\mathrm{T}}z_1(t_{i-m+1}) + \boldsymbol{b}_0) - \tanh(\boldsymbol{b}_0) \\ \qquad\qquad \vdots \\ z_1(t_i) = \tanh(\boldsymbol{W}_{x0}^{\mathrm{T}}\boldsymbol{v}(t_i) + \boldsymbol{W}_{u0}^{\mathrm{T}}\boldsymbol{u}(t_i) \\ \qquad\qquad + \boldsymbol{W}_{w0}^{\mathrm{T}}\boldsymbol{w}'(t_i) + \boldsymbol{W}_{r0}^{\mathrm{T}}z_1(t_{i-1}) + \boldsymbol{b}_0) - \tanh(\boldsymbol{b}_0) \\ z_2(t_i) = \tanh(\boldsymbol{W}_1^{\mathrm{T}}z_1(t_i) + \boldsymbol{b}_1) - \tanh(\boldsymbol{b}_1) \\ z_3(t_i) = \tanh(\boldsymbol{W}_2^{\mathrm{T}}z_2(t_i) + \boldsymbol{b}_2) - \tanh(\boldsymbol{b}_2) \\ \boldsymbol{a}_{NN}(t_i) = \boldsymbol{W}_3^{\mathrm{T}}z_3(t_i)\,. \end{cases} \tag{9}$$

The parameters of Eq. (9) are equivalent to those of Eq. (7). The difference between Eqs. (7) and 9 is that Eq. (9) ignores the memory before t_{i-m+1} by explicitly substituting the input features of t_{i-j} for $j = 0, 1, \ldots, m-1$. Equation (7) can be understood as the limit as $m \to \infty$ in Eq. (9).

Formally, the input-output relation at time t_i is expressed as follows:

$$\begin{aligned} \boldsymbol{a}_{NN}(t_i) = \boldsymbol{f}_{2NN}(&\boldsymbol{v}(t_i), \boldsymbol{v}(t_{i-1}), \ldots, \boldsymbol{v}(t_{i-m+1}), \\ &\boldsymbol{u}(t_i), \boldsymbol{u}(t_{i-1}), \ldots, \boldsymbol{u}(t_{i-m+1}), \\ &\boldsymbol{w}'(t_i), \boldsymbol{w}'(t_{i-1}), \ldots, \boldsymbol{w}'(t_{i-m+1}); \boldsymbol{\theta})\,, \end{aligned} \tag{10}$$

The prediction accuracies of the two models will be compared in Sect. 4.

2.2 Optimization Methods

In previous studies on maneuvering models using NN [29, 31, 32], the maneuvering models were predicted by optimizing a loss function in which the instantaneous accelerations were directly evaluated Eq. 16. In those existing researches, training data were obtained from numerically simulation with the artificial noises. Therefore, direct evaluation of accelerations was not a critical issue.

However, the measurement data taken in the actual environment include unignorable noise and uncertainties. In particular, acceleration terms are likely to contain noise, as will be discussed in Sect. 3. In such a case, directly using the measured accelerations in the loss function deteriorates the optimization performance of the NN parameters.

To handle such a difficulty, in this study, we newly proposed a new loss function to reduce the measurement noise. Thus, as an alternative to directly use acceleration, the loss function evaluates the positions and velocities from numerical simulation. We now explain two loss functions: (I) loss function evaluating acceleration and (II) the proposed loss function evaluating state variables, i.e., positions and velocities.

First, we explain (I). The training process is summarized in Table 1. In this method, the trajectory dataset of state variables $\hat{\boldsymbol{x}}(t_{n,i})$, control inputs $\hat{\boldsymbol{x}}(t_{n,i})$, wind disturbances $\hat{\boldsymbol{w}}'(t_{n,i})$, and accelerations $\hat{\boldsymbol{a}}(t_{n,i})$ are required. Subsequently, hat $\hat{}$ denotes the training data. N_T is the number of time steps of the trajectories, N is the number of these trajectories, and $t_{n,i}$ represents the discretized time of the i-th step of the n-th trajectory. Then, the acceleration is predicted by the NN model using Eqs. 12 and 14), or Eqs. 13 and 15. Thus, the loss function is calculated from the predicted acceleration using Eq. 16. Here, $a_{NN,j}$ and $\hat{a}_j$ represented the j-th components of $\boldsymbol{a}_{NN}$ and $\hat{\boldsymbol{a}}$, respectively. $\sigma_{\hat{a}_j}$ is the standard deviation of $\hat{a}_j$ in the training dataset. Then, by optimizing this loss function using the gradient descent method, every parameter inside the NN can be determined. In this study, Adam [45] was used as the optimization method. However, as mentioned above, the

Table 1 Training process

	The case of Eq. (7)	The case of Eq. (9)
Required dataset	$\left\{ \{\hat{\boldsymbol{x}}(t_{n,i}), \hat{\boldsymbol{u}}(t_{n,i}), \hat{\boldsymbol{w}}'(t_{n,i}), \hat{\boldsymbol{a}}(t_{n,i})\}_{i=0,\ldots,N_T-1} \right\}_{n=1\ldots,N}$ (11)	
Initialization	$z_1(t_0) = \boldsymbol{0}$ (12)	$\boldsymbol{a}_{NN}(t_{n,j}) = \hat{\boldsymbol{a}}_{NN}(t_{n,j})$ for $j = 0, \ldots, m-1$ (13)
Iteration process	$\boldsymbol{a}_{NN}(t_{n,i}) = \boldsymbol{f}_{1NN}(\hat{\boldsymbol{v}}(t_{n,i}), \hat{\boldsymbol{u}}(t_{n,i}), \hat{\boldsymbol{w}}(t_{n,i}), z_1(t_{n,i-1}); \boldsymbol{\theta})$ (14)	$\boldsymbol{a}_{NN}(t_{n,i}) = \boldsymbol{f}_{2NN}(\hat{\boldsymbol{v}}(t_{n,i}), \hat{\boldsymbol{v}}(t_{n,i-1}), \ldots, \hat{\boldsymbol{v}}(t_{n,i-m+1}), \hat{\boldsymbol{u}}(t_{n,i}), \hat{\boldsymbol{u}}(t_{n,i-1}), \ldots, \hat{\boldsymbol{u}}(t_{n,i-m+1}), \hat{\boldsymbol{w}}'(t_{n,i}), \hat{\boldsymbol{w}}'(t_{n,i-1}), \ldots, \hat{\boldsymbol{w}}'(t_{n,i-m+1}); \boldsymbol{\theta})$ (15)
Loss function	$\mathcal{L}_{\mathrm{acc}}(\boldsymbol{\theta}) = \frac{1}{N}\frac{1}{N_T}\sum_{n=1}^{N}\sum_{i=0}^{N_T-1}\sum_{j=1}^{3}\left\lvert\frac{a_{NN,j}(t_{n,i}) - \hat{a}_j(t_{n,i})}{\sigma_{\hat{a}_j}}\right\rvert^2$ (16)	

Here, acceleration components are directly evaluated. In this table, both formulations for Eq. (7) or Eq. (9) are listed

measured accelerations $\hat{\boldsymbol{a}}$ taken in the actual environment contained an unignorable noise, so applying Eq. 16 to such noisy data could deteriorate the prediction accuracy.

Second, we explain (II). This loss function evaluates the positions and velocities alternative instead of directly evaluating accelerations. The training process is shown in Table 2. In this method, the trajectory dataset of state variables $\hat{\boldsymbol{x}}(t_{n,i})$, control inputs $\hat{\boldsymbol{u}}(t_{n,i})$, and wind disturbance $\hat{\boldsymbol{w}}'(t_{n,i})$ are required, whereas that of acceleration $\hat{\boldsymbol{a}}(t_{n,i})$ is not. To predict the state variables $\boldsymbol{x}(t_{n,i})$, numerical simulation is preformed with the measured control inputs $\hat{\boldsymbol{u}}(t_{n,i})$, wind disturbances $\hat{\boldsymbol{w}}'(t_{n,i})$, and initial conditions using Eq. 18 or Eq. 19. In this study, the Euler method is used to solve ordinary differential equations solver. The prediction of positions and yaw angle $\boldsymbol{p}(t_{n,i+1})$ are simulated using Eq. 20. Notably, $u(t_{n,i})$, $v_m(t_{n,i})$, and $r(t_{n,i})$ are elements of $\boldsymbol{v}(t_{n,i})$ calculated using the Euler method, and $\psi(t_{n,i})$ was an element of $\boldsymbol{p}(t_{n,i})$ calculated also using the Euler method. Besides, at the same time, the velocity and angular velocity $\boldsymbol{v}(t_{n,i+1})$ can be calculated as Eq. 21 or Eq. 22. At this time, the acceleration $\boldsymbol{a}_{NN}(t_{n,i})$ is calculated by Eq. (7) or Eq. (9). Notably, the state variables $\boldsymbol{v}(t_{n,i})$ are obtained from the simulation, whereas the control input $\hat{\boldsymbol{u}}(t_{n,i})$ and wind disturbance $\hat{\boldsymbol{w}}'(t_{n,i})$ are from the given training data set. Therefore, the position and velocity components can be calculated from the outputs of NN. Finally, the loss function is defined as Eq. 23, where, x_j and $\hat{x}_j$ represent j-th components of $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$, respectively. $\sigma_{\hat{x}_j}$ is the standard deviation of $\hat{x}_j$ in the training data set. Notably that $\boldsymbol{x}(t_{n,i})$ comprises $\boldsymbol{p}(t_{n,i})$ and $\boldsymbol{v}(t_{n,i})$. Then, by optimizing this loss function using the gradient descent method, every parameter inside the NN can be determined.

By introducing the numerical simulation, the direct use acceleration was avoided in the loss function, thus a higher optimization performance can be expected. Analogously, the accumulation error in the simulation can be considered in the loss function. Therefore, using this loss function, it is also expected that the noise effect is relatively reduced. Moreover, as easily imagined, the computational cost due to numerical simulation becomes higher. However, since this calculation is usually performed not online but offline, the cost does not seem to be a critical issue. Further to increase the computational speed, a graphics processing unit (GPU) is used.

The comparison of the two loss functions will be shown in the next Sect. 4.

2.3 MMG model

In this article, the MMG model [13] is used for comparison. A brief description of the MMG model used in study is given here.

First, the equation of maneuvering motion was defined as Eq. (24) based on the MMG model concept:

Table 2 Training process

	The case of Eq. (7)	The case of Eq. (9)
Required dataset	$\left\{\left\{\hat{\boldsymbol{x}}(t_{n,i}), \hat{\boldsymbol{u}}\left(t_{n,i}\right), \hat{\boldsymbol{w}}'\left(t_{n,i}\right)\right\}_{i=0,\ldots,N_T-1}\right\}_{n=1\ldots,N}$ (17)	
Initialization	$\begin{cases} \boldsymbol{p}(t_{n,0}) = \hat{\boldsymbol{p}}(t_{n,0}) \\ \boldsymbol{v}(t_{n,0}) = \hat{\boldsymbol{v}}(t_{n,0}) \end{cases}$ (18)	$\begin{cases} \boldsymbol{p}(t_{n,i}) = \hat{\boldsymbol{p}}(t_{n,i}) \\ \boldsymbol{v}(t_{n,i}) = \hat{\boldsymbol{v}}(t_{n,i}) \end{cases}$ for $i = 0, \ldots, m-1$ (19)
Iteration process (position)	$\begin{cases} \dot{\boldsymbol{p}}(t_{n,i}) = \begin{pmatrix} u(t_{n,i})\cos\psi(t_{n,i}) - v_m(t_{n,i})\sin\psi(t_{n,i}) \\ u(t_{n,i})\sin\psi(t_{n,i}) + v_m(t_{n,i})\cos\psi(t_{n,i}) \\ r(t_{n,i}) \end{pmatrix} \\ \boldsymbol{p}(t_{n,i+1}) = \boldsymbol{p}(t_{n,i}) + \Delta t\dot{\boldsymbol{p}}(t_{n,i}) \end{cases}$ (20)	
Iteration process (velocity)	$\begin{cases} \boldsymbol{a}_{NN}(t_{n,i}) = \\ \quad \boldsymbol{f}_{1NN}(\boldsymbol{v}(t_{n,i}), \hat{\boldsymbol{u}}(t_{n,i}), \hat{\boldsymbol{w}}'(t_{n,i}), z_1(t_{n,i-1}); \boldsymbol{\theta}) \\ \boldsymbol{v}(t_{n,i+1}) = \boldsymbol{v}(t_{n,i}) + \Delta t\boldsymbol{a}_{NN}(t_{n,i}) \end{cases}$ (21)	$\begin{cases} \boldsymbol{a}_{NN}(t_{n,i}) = \\ \quad \boldsymbol{f}_{2NN}(\boldsymbol{v}(t_{n,i}), \boldsymbol{v}(t_{n,i-1}), \ldots, \boldsymbol{v}(t_{n,i-m+1}), \\ \qquad \hat{\boldsymbol{u}}(t_{n,i}), \hat{\boldsymbol{u}}(t_{n,i-1}), \ldots, \hat{\boldsymbol{u}}(t_{n,i-m+1}), \\ \qquad \hat{\boldsymbol{w}}'(t_{n,i}), \hat{\boldsymbol{w}}'(t_{n,i-1}), \ldots, \hat{\boldsymbol{w}}'(t_{n,i-m+1}); \boldsymbol{\theta}) \\ \boldsymbol{v}(t_{n,i+1}) = \boldsymbol{v}(t_{n,i}) + \Delta t\boldsymbol{a}_{NN}(t_{n,i}) \end{cases}$ (22)
Loss function	$\mathcal{L}_{\text{state}}(\boldsymbol{\theta}) = \frac{1}{N}\frac{1}{N_T}\sum_{n=1}^{N}\sum_{i=0}^{N_T-1}\sum_{j=1}^{6}\left\lvert\frac{x_j(t_{n,i}) - \hat{x}_j(t_{n,i})}{\sigma_{\hat{x}_j}}\right\rvert^2$ (23)	

Here, only state variables are evaluated, and acceleration components are not directly evaluated. In this table, both formulations for Eq. (7) or Eq. (9) are listed

Table 3 Principal particulars

Item	Value
Length: L_{pp}	3.0m
Breadth: B	0.49m
Draft: d	0.20m
Block coefficients: C_b	0.83

Table 4 Ranges of control inputs

Item	Range
n[rps]	$[-20, 20]$
δ[deg.]	$[-35, 35]$

$$\begin{cases} (m+m_x)\dot{u} - (m+m_y)v_m r - x_G m r^2 = X_H + X_P + X_R \\ (m+m_y)\dot{v}_m + (m+m_x)ur + x_G m\dot{r} = Y_H + Y_P + Y_R \\ (I_{zz} + J_{zz} + x_G^2 m)\dot{r} + x_G m(\dot{v}_m + ur) = N_H + N_P + N_R \,. \end{cases} \tag{24}$$

where m, m_x, and m_y represent the ship mass, longitudinal added mass and lateral added mass, respectively; I_{zz} and J_{zz} represent the moment of inertia and added moment of inertia, respectively; and x_G represents the longitudinal position of the center of gravity. On the right-hand side of the equations, X, Y, and N represent the longitudinal force, lateral force, and yawing moment, respectively. The subscripts H, P, and R represent the hull, propeller, and rudder components, respectively.

The hydrodynamic forces acting on the hull are predicted using a unified model for low-speed maneuvering and open sea navigation [46], which is widely used in cases with large oblique angles. For the positive longitudinal speed and positive propeller RPS, the mathematical modeling of the propeller and rudder were based on the conventional MMG model [13]. Meanwhile, for the negative propeller RPS, the models of [47] for thrust, [48] lateral force, moment by a propeller, and rudder force [49] are used. For further details of applied models, refer to [50].

3 Subject ship and experiment

In this section, the details of free running model tests are explained. The subject model ship was the M.V. ESSO OSAKA (Fig. 2), and the model scale was 1/108.33. The principal particulars of this model are listed in Table 3.

The onboard control devices and systems of this model were developed by Wada et al. [51]. It uses robot operating system (ROS) as the middleware. The upper and lower limits of control variables in the model tests are shown in Table 4.

One optical fiber gyro, three GNSS receivers, and two ultrasonic anemometers were installed in the system. The state variable $\boldsymbol{x}$ and disturbance $\boldsymbol{w}$ were measured from these measurement devices. The optical fiber gyro directly measured the yaw angular velocity r. Although the optical fiber gyro also output ψ as the integration results of r, the accumulation of error generated drift of ψ. To avoid this drift error, the heading angle ψ was calculated from two or three GNSS data in this experiment. The time series of the control variable $\boldsymbol{u}$ were simultaneously recorded. For details of the measurements and analysis methods, see [50].

In almost all SI studies on the ship maneuvering models using NN, the target is basically to maneuver with positive longitudinal speed, and maneuver modes are zigzag and turning tests [29, 31–33]. In [30], the propeller RPS was constant. However, this study proposed a low-speed berthing maneuvering model that has the form of an NN. In the berthing maneuver, there was a state combination of positive/negative propeller RPSs and the positive/negative longitudinal speeds, hence the model significantly differs from the usual maneuver. Therefore, we handled the training data carefully. As a countermeasures, we conducted manual random maneuver test to acquire the uniformly distributed combination of states and input.

The manual random maneuver test is a free running test in which the control input was randomly generated regardless of the state. So, uniformly distributed state and control combinations were expected. In this experiment, to establish

Fig. 2 Photo of the subject model ship: M.V. ESSO OSAKA

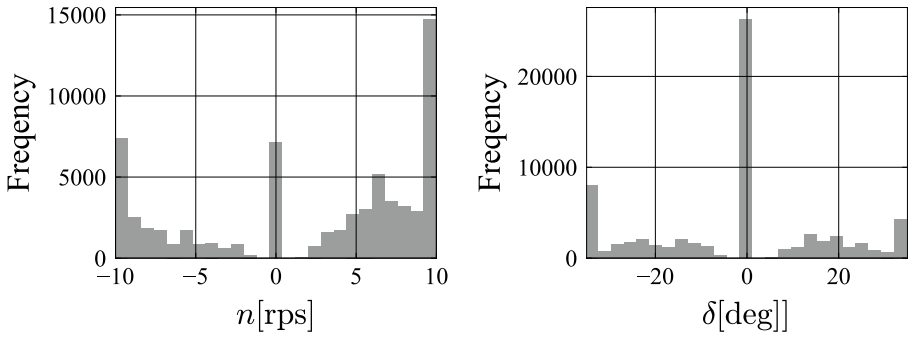

Fig. 3 Histgram of control input in manual random maneuvering test. In randam maneuvering test, the range of propeller RPS n is $[-10, 10]$

Table 5 Model tests and its labels

Label	Type of model test
T	Turning
Z	Zigzag
B	Berthing - STBD. & PORT.
R	Random

Table 6 List of training data sets

Data set	T [s]	Z [s]	R [s]	B [s]	Total [s]
Train-TZB	1490	737.1	0.0	335.8	2562.9
Train-TZRB	556.4	342.9	1301.2	335.8	2536.3
Train-TZRB+	5674.7	1151	5861.2	788.9	13475.8
Test (TZRB)	424.6	193.8	717.9	380.6	1716.9

Each sampling frequency was 10 [Hz]

the uniformly distributed input and state, manual controls by the gaming controller were conducted. The histogram of the control input in the manual random maneuver test is shown in Fig. 3. In reality, owing to the limitation of the geometrical size of the pond, the combination did not completely become uniform distribution. In particular, the frequencies are biased at the upper and lower limits and zero. However, we can see that the control input is tried to achieve the uniform distribution in the other values.

Besides, turning and zigzag tests were conducted to collect data. In addition to these tests, the berthing test was conducted. This test was conducted in the middle of the experimental pond, and the existence of the berth is not taken into account. The tests conducted are summarized in Table 5.

In this study, we prepared one test dataset and three training data sets (Table 6). The sampling frequency in each test was 10 [Hz]. Each test has a label as listed in Table 5, and hereafter, for instance, "TZRB" in Train-TZRB means that Train-TZRB has turning tests, zig-zag tests, manual random maneuvering tests, and berthing tests.

In Train-TZR and Train-TZRB, the amount of training data was intentionally limited to approximately 7 h in the actual ship scale. Meanwhile, the Train-TZRB+ had almost five times that amount of data. The purpose of using Train-TZRB+ was to confirm the dependency of data quantity.

Table 7 The computational experiments conducted in this study

	NN model	Loss function	Data set
Type-1	Eq. (9)	Eq. (23)	Train-TZRB+
Type-2	Eq. (7)	Eq. (23)	Train-TZRB+
Type-3	Eq. (9)	Eq. (16)	Train-TZRB+
Type-4	Eq. (9)	Eq. (23)	Train-TZB
Type-5	Eq. (9)	Eq. (23)	Train-TZRB

4 Results

In this section, the final results and discussion are shown. To confirm the dependency of the prediction accuracy on the NN model, loss function, and data set, the prediction for the maneuvering model was conducted using five methods (Table 7). As will be detailed later, Type-1 has the best performance, so we focus on the explanation of the reason. In Sect. 4.1, Type-1 and Type-2 are compared. The dependency of prediction accuracy on the NN model is discussed. In Sect. 4.2, Type-1 and Type-3 are compared. We discuss the dependency of prediction accuracy on the loss function. In Sect. 4.3, Type-1, Type-4, and Type-5 are compared. We discuss the dependency of prediction accuracy on the data set. In particular, comparing Type-4 and Type-5, the effectiveness of the manual random maneuvering test is confirmed. In Sect. 4.4, we compare the NN models, particularly Type-1 and Type-5, and the MMG model, whose coefficients are determined by the captive model tests.

To prevent the problem of overfitting, the dataset was divided into two parts—the training and validation datasets. The training data were used for optimization, whereas the validation data were used to detect the start of overfitting. The parameters inside the NN that minimized the loss function in the validation data were considered the final solution. Python was used as the programming language,

Table 8 Hyperparameter

Batch size	512
Learning rate : η	2.0×10^{-5} or 1.0×10^{-4} (if loss function is Eq. (16))
Predicted steps : N_T	60 (6.0 [s])
Memory steps : m	10 (1.0 [s])

and PyTorch was used as the deep learning library. The hyperparameters used in the training are listed in Table 8.

The training process and prediction results were shown before the comparison of each method. The descent process of the loss functions in the validation dataset during training is shown in Fig. 4. Generally, optimization of the NN parameters is likely to depend on the initial distribution of parameters, so the training of the NN was performed five times with different random seeds. Figure 4 shows that the results did not strongly depend on the random seeds in this case. Notably, since the loss functions or the quantity of the datasets were different for each method, the direct comparison of loss value did not have meaning.

We explain the procedure to evaluate the estimation accuracy of the trained model. To quantitatively evaluate the prediction accuracy, the mean squared error (MSE) of the standardized state of trajectories between the prediction results of the NN $\boldsymbol{x}$ and experimental results $\hat{\boldsymbol{x}}$ was employed. This index was defined as follows:

$$\mathcal{L}_{\mathrm{MSE}} = \frac{1}{N_T} \sum_{i=0}^{N_T} \sum_{j=1}^{6} \left| \frac{x_j(t_i) - \hat{x}_j(t_i)}{\sigma_{\hat{x}_j}} \right|^2 . \tag{25}$$

Here, $\boldsymbol{\sigma}_{\hat{\boldsymbol{x}}}$ is the standard deviation of $\hat{\boldsymbol{x}}$ in the test data set.

The comparative results of Eq. (25) between the NN and MMG models are summarized in Table 9. The results of the NN model were shown as the mean values and standard deviations since the calculations were conducted for five different random seeds. The manual random maneuvering and turning tests had long measurement records. Therefore, to avoid the accumulation of error, the time domain simulation was restarted at every 100 s. By avoiding the accumulation of error, we highly expected the optimization performance to improve. At every restart, the initial condition was reset

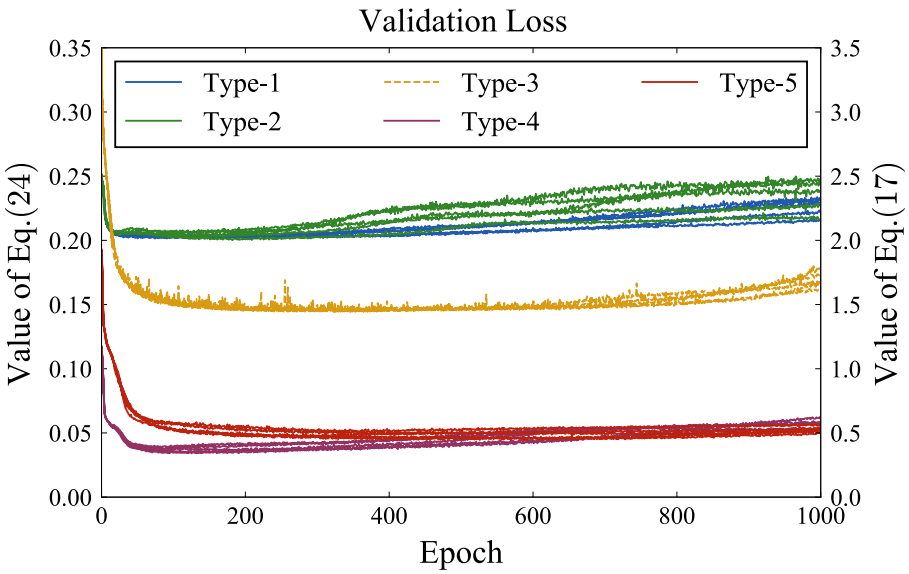

Fig. 4 Validation loss value during training

Table 9 Mean squared error $\mathcal{L}_{MSE}$ of the prediction in test data set

	MMG	Type-1		Type-2		Type-3		Type-4		Type-5	
	EDF Coeff.	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Random	0.272	0.236	0.003	0.241	0.005	4834	4041	0.488	0.022	0.262	0.007
Turning	0.858	0.713	0.011	0.728	0.017	0.864	0.100	0.701	0.019	0.785	0.024
Zigzag	1.015	0.086	0.063	0.124	0.141	761.1	1519	0.122	0.054	0.856	0.288
Berthing1	0.036	0.052	0.024	0.058	0.031	634.8	1264	0.144	0.053	0.055	0.030
Berthing2	0.189	0.069	0.014	0.140	0.101	91.76	124.6	0.246	0.107	0.182	0.047
Berthing3	0.100	0.138	0.023	0.167	0.053	1.163	0.617	0.302	0.062	0.294	0.043

to the measured experimental value. Besides, since the setting of initial values in Type-2 was different from others, the way initial values were assigned was changed. While $i = 0, 1, \ldots, m-1$, after the simulation was calculated using Eqs. 20 and 21, the position $\boldsymbol{p}(t_{n,i})$ and velocity $\boldsymbol{v}(t_{n,i})$ were overwritten by Eq. 19. At this time, the latent variable $\boldsymbol{z}_1$ was not overwritten, but was resubstituted as an input feature in the next time step.

4.1 Comparison of NN models

In this section, by comparing Type-1 and Type-2, the dependency of the prediction accuracy on the model was discussed. In Type-1, the RNN that ignores the memory before a certain time expressed in Eq. (9) was used. In Type-2, the "standard" RNN model expressed in Eq. (7) was used. Equation 23 was used as the loss function, and Train-TZRB+ was used as the training data set.

Table 9 shows that the value $\mathcal{L}_{MSE}$ of Type-1 was always lower than the value of the Type-2. Generally speaking, if the memory span m of the NN model in Type-1 was made longer, the prediction accuracy could be improved. However, Table 9 shows that the estimation accuracy did not necessarily improve in large memory step m, which implied that using long distant past information might deteriorate the performance of the dynamic system. Considering the zigzag test. Zig-zag motion is a result of periodic control. Then, in its long periodicity might be considered. However, its long-distant memories were not generally essential for dynamical system modeling. Therefore, when predicting the maneuvering model, the memory span m should be the span that is physically affected. In other words, if the memory span m is an appropriate span that is not too long, the prediction accuracy will improve.

4.2 Comparison of loss function

In this section, comparing Type-1 and Type-3, the dependency of the prediction accuracy on the loss function was discussed. In Type-1, the proposed loss function Eq. 23 was used. In Type-3, the loss function that directly evaluates acceleration Eq. 16 was used. Equation (9) was used as the NN model, and Train-TZRB+ was used as the data set. In addition to Table 9, the simulation results of Type-1 and Type-3 in the manual random maneuver test dataset are shown in Fig. 5.

In Table 9, Type-3 generally showed extreamly larger prediction error than Type-1. This is because, in Random and Zigzag manuevers, the simulation of Type-3 shows the tendency of divergence.

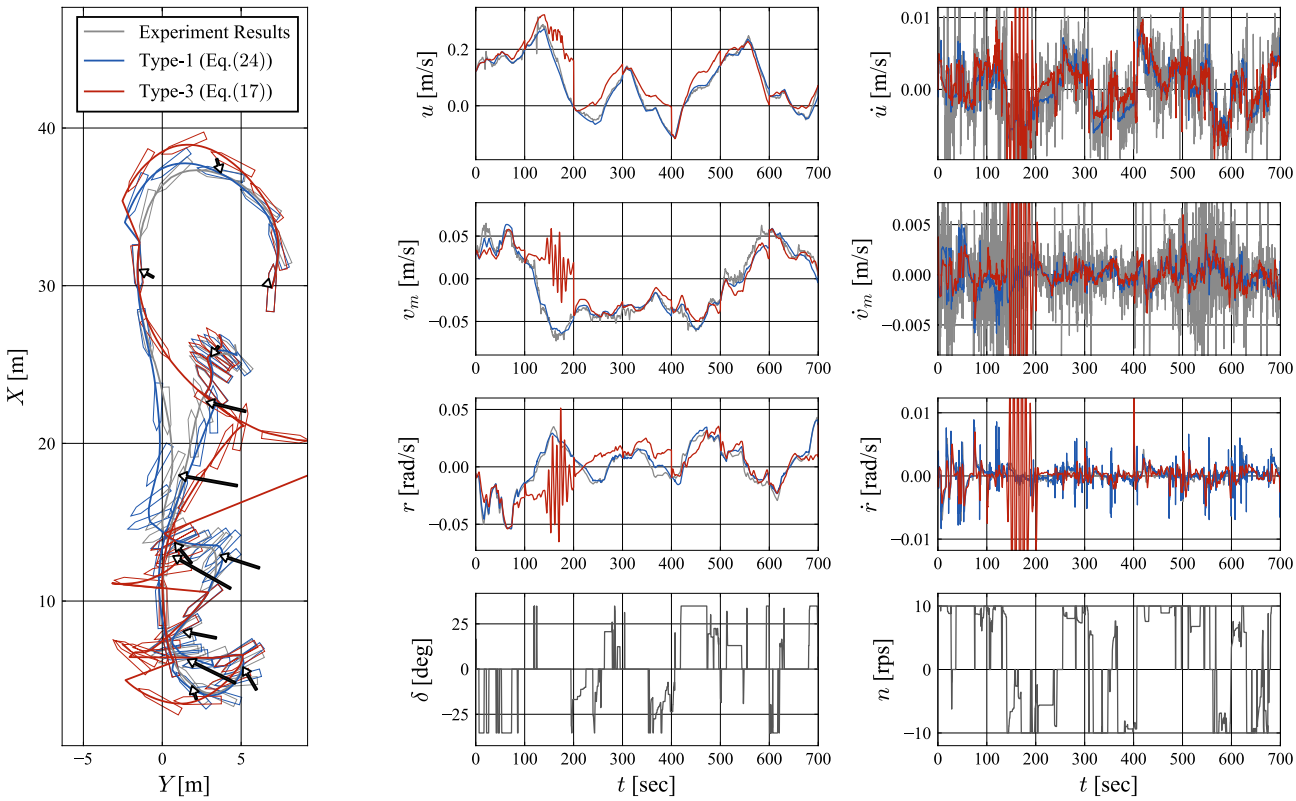

Fig. 5 Comparison of the trajectories obtained for two loss functions Eqs. 23 and 16. The training dataset is Train-TZRB+ in both methods

Besides, Fig. 5 shows that, in the proposed loss function, the predicted position and velocities almost coincided with those of the experimental results, whereas the accelerations did not show a good coincidence. Notably, as stated above, experimentally obtained accelerations, $\dot{u}$ and $\dot{v}_m$, were likely to include noises and spikes since these properties were derived from the time differentiation for the position data from GNSSs. Some parts of the spikes in experimental data were induced by data processing, i.e., time differentiation. These results showed that the proposed loss function Eq. 23 enabled us to predict the maneuvering model by avoiding the direct evaluation of accelerations.

Moreover, the NN and experimental results of the yaw angular acceleration $\dot{r}$ coincided, and the tendency was not the same as the acceleration components, $\dot{u}$ and $\dot{v}_m$. In our data acquisition system, the yaw angular velocity was directly measured from FOG, and the yaw angular acceleration $\dot{r}$ was obtainable from one differential operation. Therefore, the spikes in the yaw angular acceleration $\dot{r}$ were likely to be lesser than those in the acceleration component. Therefore, the spikes that appeared in the experimental data of $\dot{r}$ were not necessarily noises.

Therefore, we concluded that the proposed loss function Eq. 23 could reduce the effect of noise, thereby significantly improve the prediction accuracy of positions and velocities.

4.3 Comparison of dataset

In this section, comparing Type-1, Type-4, and Type-5, the dependency of prediction accuracy on the data set was assessed. In Type-1, Train-TZRB+, which has the largest data, was used as the dataset. In Type-4, Train-TZB, which did not contain manual random maneuvering test data, was used as the dataset. In Type-5, Train-TZRB, which contained the manual random maneuvering test data, was used as the dataset. To confirm the effectiveness of the manual random maneuvering test, Train-TZB and Train-TZRB have the same amount of data.

From Table 9, the performance of Type-1 was better than those of Type-4 and Type-5, which implied that a larger amount of dataset made the NN model more accurate. However, it is unrealistic to acquire an extremely large amount of training data. Thereofre, the balance of the amount of data and the prediction accuracy should be considered in practical applications.

Comparing Type-4 and Type-5, in Table 9, the prediction accuracy of the manual random maneuver test using the NN trained for Type-5, namely Train-TZRB, was better. Meanwhile, in the turning and zigzag tests, the prediction accuracy of the NN trained for Type-4, namely, Train-TZB, was better. These simply implied that the prediction accuracy for test datasets apparently depended on the inclusion of similar data in the training dataset.

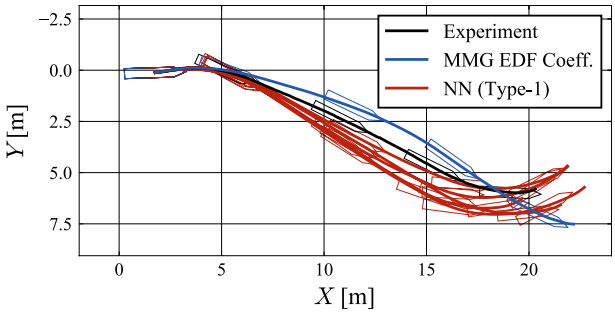

Fig. 6 Comparison of prediction accuracy by each dataset in Berthing1 test. Time of test in model scale is 147.0[s]

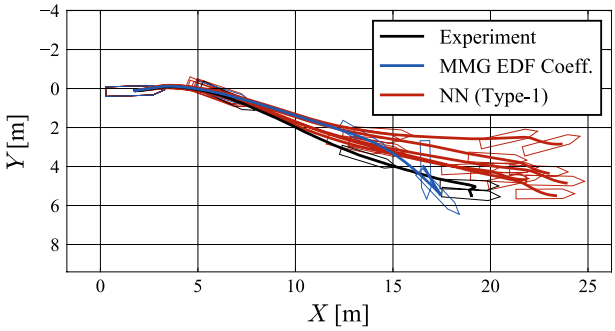

Fig. 7 Comparison of prediction accuracy by each dataset in Berthing2 test. Time of test in model scale is 127.6[s]

From Table 9, the result of Type-5 was better in Berthing1 and Berthing2 , and slightly better in Berthing3. In other words, in berthing test, which was the focus of this study, the prediction accuracy of Type-5 was generally better than that of Type-4. Therefore, the inclusion of manual random maneuvering tests in the training dataset successfully made the prediction performance of NN better, even in the berthing control.

4.4 Comparison with MMG

In this section, the results using the MMG model and the NN model, particularly Type-1 and Type-5, were compared. The coefficients of the MMG model were determined by the captive model tests [50]. The simulation results of Type-1 and the MMG model in berthing test dataset are shown in Figs. 6, 7 and 8. In Type-1, the results of five patterns with different random seeds was drawn.

Table 9 shows that, in the random, turning, and zigzag tests, the prediction accuracies of Type-1 and Type-5 were better than that of the MMG model, which might be because

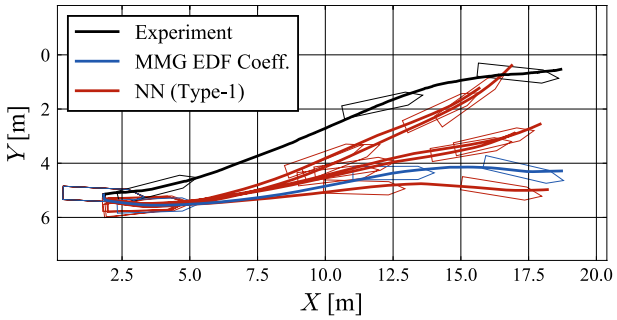

Fig. 8 Comparison of prediction accuracy by each dataset in Berthing3 test. Time of test in model scale is 105.7[s]

the training dataset contained enough motion data similar to the test dataset.

Meanwhile, in Berthing1 and Berthing3, the prediction accuracy of the MMG model was better than those of Type-1 and Type-5, whereas the prediction accuracy of Type-1 and Type-5 was better than that of the MMG model in Berthing2, which could be considered that although maneuvers, which imitates the berthing motion were conducted in manual random maneuvering tests, the prepared training data failed to sufficiently cover the complex motion in actual berthing maneuver.

However, the comparisons of all five trajectories for Type-1 are shown in Figs. 6, 7 and 8. From these figures, although the results slightly depended on the random seeds, it was understood that the Type-1 results showed good prediction performance compared with the MMG model results. Moreover, increasing the percentage of berthing tests in the training data significantly contributes in improving the prediction accuracy of the berthing maneuvers.

5 Conclusion

In this research, the following five main outcomes were uncovered. (i) Low-speed maneuvering model could be well approximated by RNN. (ii) Comparison of two RNN models showed that the prediction accuracy of the RNN that ignores the memory before a certain time was better than the "standard" RNN. (iii) To eliminate or reduce the adverse effect of the noise included in the measurement data, a loss function that evaluated possitions and velocities instead of acceleration directly was newly proposed. Then, the NN trained under the proposed loss function showed a significant improvement in prediction accuracy. (iv) To obtain a better NN that can predict ship trajectories under berthing control, not only the usual maneuvering data, such as turning and zigzag tests, but also manual random maneuver tests were used in the training phase of NN. Then, it was confirmed that the inclusion of manual random maneuver tests in the training data significantly improved prediction performance. (v) The performance of the NN and MMG models were compared. Concerning manual random maneuvering, turning, and zigzag tests, which were sufficiently included in the training dataset, the prediction accuracy of the NN model was better than that of the MMG model. Meanwhile, concerning the berthing maneuvers, which were not sufficiently included in the training data, the prediction accuracy of the MMG model was better than the NN model.

In the proposed prediction method using NN, the prediction accuracy strongly depended on the dataset, in particular the amount of data and types of manuevers in the data set. In addition, the prediction accuracy depended on random seeds. Therefore, the proposed method using RNN is not necessarily considered to be perfect. To make the proposed method more relizable and precise, preparation of the appropriate and sufficient amount of training data are essential. The detailed exploration for this point will be one of our future studies.

Acknowledgements This study was supported by a Grant-in-Aid for Scientific Research from the Japan Society for Promotion of Science (JSPS KAKENHI Grant #19K04858). The study also received assistance from JFY2018 Fundamental Research Developing Association for Shipbuilding and Offshore (REDAS) in Japan. Authors also would like to express gratitude to Mr. Satoru Konishi, Magellan Systems Japan Inc., to the technical support on GNSS measurement during the free run model test.

References

1. Ministry of Land, Infrastructure, Transport and Tourism (2020) White paper on land, infrastructure, transport and tourism in Japan
2. Kose K, Hinata H, Hashizume Y, Futagawa E (1984) On a mathematical model of maneuvering motions of ships in low speeds. J Soc Naval Arch Jpn 155:132–138
3. Yamato H (1990) Automatic berthing by the neural controller. Proc Nineth Ship Control Syst Sympos 3:3183–3201
4. Shouji K, Ohtsu K (1992) A study on the optimization of ship maneuvering by optimal control theory (1st report). J Soc Naval Arch Jpn 172:365–373
5. Hasegawa K, Kitera K (1993) Automatic berthing control system using network and knowledge-base. J Kansai Soc Naval Arch Jpn 220:135–143
6. 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):221–233
7. Miyauchi Y, Sawada R, Akimoto Y, Umeda N, Maki A (2021) Optimization on planning of trajectory and control of autonomous berthing and unberthing for the realistic port geometry. Ocean Eng. (to be published). arXiv:2106.02459
8. Rachman DM, Maki A, Umeda N (2020) Numerical simulation of automatic berthing by CMA-ES in real time. In: Conference proceedings, the Japan Society of Naval Architects and Ocean Engineers (in Japanese), 11
9. Rachman DM, Umeda N, Maki A and Yoshiki M (2021) Feasibility study on the use of evolution strategy: CMA-ES for ship

automatic docking problem, In: conference proceeding, 1st International Conference on the Stability and Safety of Ships and Ocean Vehicles, 2021
10. Akimoto Y, Miyauchi Y, Maki A (2021) Saddle point optimization with approximate minimization oracle and its application to robust berthing control (to be published). arXiv:2105.11586
11. Wakita K, Akimoto Y, Shoji K, Miyauchi Y, Umeda N, Maki A (2021) On transfer learning of the pre-trained policy of ship tracking control in a real environment. In: Conference proceedings, Japan Society of Naval Architects and Ocean Engineers, vol 32, pp 23–30 (in Japanese)
12. Abkowitz M (1980) Measurement of hydrodynamic characteristics from ship maneuvering trials by system identification. Transactions of Society of Naval Architects and Marine Engineers, vol 88, pp 283–318
13. Yasukawa H, Yoshimura Y (2015) Introduction of mmg standard method for ship maneuvering predictions. J Mar Sci Technol 20(1):37–52
14. Sakamoto N, Ohashi K, Araki M, Kume K, Kobayashi H (2019) Identification of kvlcc2 manoeuvring parameters for a modular-type mathematical model by rans method with an overset approach. Ocean Eng 188:106257
15. Sutulo S, Soares CG (2014) An algorithm for offline identification of ship manoeuvring mathematical models from free-running tests. Ocean Eng 79:10–25
16. Araki M, Hamid S-H, Sanada Y, Tanimoto K, Umeda N, Frederick S (2012) Estimating maneuvering coefficients using system identification methods with experimental, system-based, and cfd free-running trial data. Ocean Eng 51:63–84
17. Miyauchi Y, Maki A, Umeda N, Dimas M R, Shimoji T, Wakita K, Akimoto Y (2020) On system identification for low-speed maneuvering model by using CMA-ES (4th report). In: Conference proceedings, Japan Society of Naval Architects and Ocean Engineers, volume 31, pp 43-54 (in Japanese)
18. Mansilla RM, Aranda J, Díaz JM, de la Cruz J (2009) Parametric model identification of high-speed craft dynamics. Ocean Eng 36(12):1025–1038
19. Nagumo J, Noda A (1967) A learning method for system identification. IEEE Trans Autom Control 12(3):282–287
20. Källström CG, Åström KJ (1981) Experiences of system identification applied to ship steering. Automatica 17(1):187–198
21. Åström KJ (1980) Maximum likelihood and prediction error methods. Automatica 16(5):551–574
22. Perera LP, Oliveira P, Guedes Soares C (2015) System identification of nonlinear vessel steering. ASME J Offshore Mech Arct Eng 137(3):031302. https://doi.org/10.1115/1.4029826
23. Yasukawa H, Ishikawa T, Yoshimura Y (2021) Investigation on the rudder force of a ship in large drifting conditions with the MMG model. J Mar Sci Technol 26:1078–1095. https://doi.org/10.1007/s00773-020-00789-4
24. Xu H, Hinostroza MA, Zihao Wang, Guedes Soares C (2020) Experimental investigation of shallow water effect on vessel steering model using system identification method. Ocean Eng 199:106940
25. Luo W, Moreira L, Guedes Soares C (2014) Manoeuvring simulation of catamaran by using implicit models based on support vector machines. Ocean Eng 82:150–159
26. Weilin L (2016) Parameter Identifiability of Ship Manoeuvring Modeling Using System Identification. Math Probl Eng 2016:8909170. https://doi.org/10.1155/2016/8909170
27. Man Z, Wuqiang S, Axel H, Yuanqiao W, Changshi X, Wei T (2020) Adaptive modeling of maritime autonomous surface ships with uncertainty using a weighted ls-svr robust to outliers. Ocean Eng 200:107053
28. Mei B, Sun L, Shi G (2019) White-black-box hybrid model identification based on rm-rf for ship maneuvering. IEEE Access 7:57691–57705
29. Moreira L, Guedes Soares C (2003) Dynamic model of manoeuvrability using recursive neural networks. Ocean Eng 30(13):1669–1697
30. Oskin DA, Dyda AA, Markin VE (2013) Neural network identification of marine ship dynamics. IFAC Proc 46(33):191–196 (**9th IFAC Conference on Control Applications in Marine Systems**)
31. Chiu F-C, Chang T-L, Go J, Chou S-K, Chen W-C (2004) A recursive neural networks model for ship maneuverability prediction. In: Oceans '04 MTS/IEEE Techno-Ocean '04 (IEEE Cat. No.04CH37600), vol 3, pp 1211–1218. https://doi.org/10.1109/OCEANS.2004.1405752
32. Rajesh G, Bhattacharyya SK (2008) System identification for nonlinear maneuvering of large tankers using artificial neural network. Appl Ocean Res 30(4):256–263
33. Weilin L, Zhicheng Z (2016) Modeling of ship maneuvering motion using neural networks. J Mar Sci Appl 15(4):426–432
34. Koda T, Furutachi K, Furukawa Y, Ibaragi H (2020) Development of RNN-based prediction model for ship manoeuvring motion. In: Conference proceedings, Japan Society of Naval Architects and Ocean Engineers, vol 30, pp 609–612 (in Japanese)
35. Koda T, Furutachi K, Furukawa Y, Ibaragi H (2020) Development of RNN-based prediction model for ship manoeuvring motion under external disturbances. In: Conference proceedings, Japan Society of Naval Architects and Ocean Engineers, vol 31, pp 87–92 (in Japanese)
36. Ken-Ichi F (1989) On the approximate realization of continuous mappings by neural networks. Neural Netw 2(3):183–192
37. Cybenko G (1989) Approximation by superpositions of a sigmoidal function. Math Control Signals Syst 2(4):303–314
38. Kurt H (1991) Approximation capabilities of multilayer feedforward networks. Neural Netw 4(2):251–257
39. Rumelhart DE, Hinton GE, Williams RJ (1985) Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science
40. Frank R (1958) The perceptron: a probabilistic model for information storage and organization in the brain. Psychol Rev 65(6):386–408
41. Krizhevsky A, Sutskever I, Geoffrey E (2012) Hinton. Imagenet classification with deep convolutional neural networks. In: Pereira F, Burges CJC, Bot-Tou L and Weinberger KQ (eds) Advances in neural information processing systems, 25, Curran Associates, Inc, pp 1097–1105
42. Vinyals O, Le Quoc V (2015) A neural conversational model. arXiv preprint arXiv:1506.05869
43. Chalapathy R, Menon AK, Chawla S (2018) Anomaly detection using one-class neural networks. arXiv preprint arXiv:1802.06360
44. Nakanishi H, Kohda T, Inoue K (1997) A design method of optimal state feed-back control systems by use of neural network. Trans Soc Instrum Control Eng 33(9):882–889
45. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980
46. Yoshimura Y, Nakao I, Ishibashi A (2009) Unified mathematical model for ocean and harbour manoeuvring. In: International Conference on Marine Simulation and Ship Maneuverability, Proceedings of MARSIM2009, pp 116–124
47. Yasukawa H (2008) Simulations of ship maneuvering in waves. J Jpn Soc Naval Arch Ocean Eng 7:163–170
48. Hachii T (2004) The prediction of manoeuvring motion on ships with low speed using standard mmg model. Master's thesis, Osaka University (**in Japanese**)
49. Kitagawa Y, Tsukada Y, Miyzaki H (2015) GS1-1 a study on mathematical models of propeller and rudder under maneuvering

with propeller reverse rotation. Conf Proc Jpn Soc Naval Arch Ocean Eng 20:117–120
50. Miyauchi Y, Maki A, Umeda N, Rachman DM, Akimoto Y (2021) System parameter exploration of ship maneuvering model for automatic docking/berthing using CMA-ES. arXiv:2111.06124
51. Wada S, Umeda N, Maki A (2019) Development of general purpose free-running model ship with ROS: enhanced model ship experiments system. Conf Proc Jpn Soc Naval Arch Ocean Eng (in Japanese) 28:587–594