

Article

Investigation of a Multitasking System for Automatic Ship Berthing in Marine Practice Based on an Integrated Neural Controller

Van Suong Nguyen 

Faculty of Navigation, Vietnam Maritime University, 484 Lachtray Str., Haiphong City 180000, Vietnam; nguyenvansuong@vimaru.edu.vn; Tel.: +84-328-754-256

Received: 15 June 2020; Accepted: 8 July 2020; Published: 16 July 2020



Abstract: In this article, a multitasking system is investigated for automatic ship berthing in marine practices, based on artificial neural networks (ANNs). First, a neural network with separate structures in hidden layers is developed, based on a head-up coordinate system. This network is trained once with the berthing data of a ship in an original port to conduct berthing tasks in different ports. Then, on the basis of the developed network, an integrated mechanism including three negative signs is linked to achieve an integrated neural controller. This controller can bring the ship to a berth on each side of the ship in different ports. The whole system has the ability to berth for different tasks without retraining the neural network. Finally, to validate the effectiveness of the proposed system for automatic ship berthing, numerical simulations were performed for berthing tasks, such as different ports, and berthing each side of the ship. The results indicate that the proposed system shows a good performance in automatic ship berthing.

Keywords: automatic ship berthing; multitasking system; artificial neural networks; integrated neural controller; head-up coordinate system

1. Introduction

When moving towards a berth at a low speed, ship dynamics become highly nonlinear; therefore, the maneuverability of the ship is decreased significantly. To ensure that ship berthing occurs safely and properly, the states of the ship, such as the position, ship heading, and ship speed, must be controlled simultaneously and appropriately by the rudder and propeller. In marine practice, because of the nonlinear character of ship dynamics at low speeds, the maneuvering process of ship berthing is usually conducted by three steps: course changing, deceleration, and engine stopping. Because artificial neural networks (ANNs) have the ability to model any multiple input–multiple output nonlinear systems, they are commonly applied to learn and to imitate the actions of the human brain during the stages of ship berthing.

Based on the advantages of neural networks, they have been used not only as centralized controllers, but also as identifiers to approximate unknown functions in dynamic systems. In automatic ship berthing, Yamato et al. (1990) [1] was the first researcher to use ANNs as a main controller, but changed their approach to the expert theory for ship berthing (Yamato et al., 1991) [2]. In that study, the inputs of the ANN included the ship position, ship heading, and ship speed, and the outputs of the controller consisted of the rudder and propeller. Later on, a multi-variable neural network was proposed by Zhang et al. (1997) [3] for ship berthing. In that work, an online training process for the network was considered via the feedback of the ship state, and the control signals were used to control the rudder and the propeller.

In 2001, Im and Hasegawa (2001) [4] proposed a parallel controller with two neural networks in a hidden layer. Numerical simulations showed that the results of ship berthing obtained by the proposed controller were better than those obtained by a centralized network. To solve ship berthing under strong winds and currents, two neural network identifiers were added to the main controller to estimate the sway velocity and the yawing rate under strong disturbances (Im and Hasegawa, 2002) [5]. These values are additionally used to correct the rudder angle and the propeller. Subsequently, selective neural networks were proposed by Im (2007) [6] for ship berthing to bring the ship into the berth from any point around the berthing area. Meanwhile, a neural controller with an adaptive interaction algorithm for online training was applied in the research of Nguyen (2007) [7]. In this work, a predetermined route for ship berthing was considered and the proposed neural controller regulated the ship heading and ship speed to track that route.

To generate consistent teaching data for training a neural controller, the studies by Ahmed and Hasegawa (2012), Ahmed and Hasegawa (2013) and Ahmed and Hasegawa (2014) [8–10] proposed a nonlinear programming language and a virtual window; the ANN controller was then verified for ship berthing with no disturbance cases. In 2018, Im and Nguyen (2018) and Nguyen et al. (2018) [11,12] proposed neural controllers for an automatic ship berthing based on head-up coordinate system, where relative bearing and distance are introduced to the inputs of the network. The advantage of these studies is the use of the information mentioned in navigational practice for automatic ship berthing. Recently, Zhang et al. (2019) [13] considered an adaptive neural control for the auto-berthing of a ship under unknown ship dynamics and in unknown environment, but the actuators are difficult to apply to marine ships. Meanwhile, an efficient neural network approach with feature selection and a genetic algorithm was proposed by Shuai et al. (2019) [14] for ship docking. In addition, a deep learning algorithm was applied for automatic ship berthing by Lee et al. (2019) [15].

With the advances in marine technology, auxiliary devices such as the side thruster and tugboat have been applied to assist ship berthing. By incorporating these devices for ship berthing, the studies of Tran and Im (2012) [16] and Ahmed and Hasegawa (2014) [17] developed a neural network controller and a PD (proportional-derivative)-ANN controller to regulate the rudder angle, propeller revolution, bow thruster, and tugboat signals for automatic ship berthing. Based on an adaptive back-stepping technique, Park and Kim (2014) [18] suggested a control law for a cruise ship, considering the uncertainty of wind disturbance. This system can control the lateral force of a ship with a crabbing motion. However, this system has a drawback, in that the ship's motion in the longitudinal direction is assumed to be zero. On the other hand, Nguyen (2019) and Nguyen and Im [19,20] introduced fuzzy controllers for ship berthing in near areas of the berth. These systems can control the ship into the berth without any conditions, unlike previous studies.

By using other approaches, such as a sliding mode control—Bui et al. (2011) [21], model prediction control—Mizuno et al. (2012) [22] and Li et al. (2020) [23], optimal control—Mizuno et al. (2015) and Maki et al. (2019), [24,25] and fuzzy logic—Liao et al. (2019) [26], the automatic control systems were also considered for ship berthing. Compared to ANN, it is difficult to realize the stages of ship berthing as human brain when using these approaches.

Although the aforementioned studies have achieved significant results in applying the ANN to automatic ship berthing problems, none of them consider a multitasking system for ship berthing without retraining the network. To use ANN controllers, as in previous studies, for different tasks of ship berthing, the neural network structure needs to be trained again by new teaching data. This is too time-consuming, costly, and complex for the control system.

To overcome the above mentioned limitations, a multitasking system based on ANNs is developed in this study to conduct different tasks of automatic ship berthing in marine practice, without retraining the network structure. First, a neural controller with separate structures in hidden layers is developed based on the parameters of a head-up coordinate system. This neural controller is trained once with the berthing data of a ship in an original port to conduct the berthing tasks for different terminals, as well as different ports. Second, on the basis of the developed controller, an integrated mechanism,

including three negative signs, is linked to achieve an integrated neural controller. This controller has the ability to berth from each side of the ship in different ports. The main contributions of this study can be summarized as follows:

- The multitasking system for automatic ship berthing is proposed based on an integrated neural controller to carry out different tasks without retraining the neural network structure.
- The ship can be automatically controlled by the proposed system to berth in different terminals as well as different ports.
- The proposed system can control the ship into the berth on each side of ship in the original port as well as different ports.
- Compared to other approaches, the proposed system is applicable for marine practices because the ANN has the ability to learn and perform the berthing stages as a human brain.

Finally, to validate the effectiveness of the proposed system for automatic ship berthing, the numerical simulations were performed for berthing tasks, such as different ports, and on each side. The results showed that this proposed system effectively performs automatic ship berthing.

2. Preliminaries and Coordinate Systems

2.1. Kinematic Model

To express the ship motion with respect to time, the kinematics of the ship need to be considered in two coordinate systems [27]: the earth-fixed frame ($O-\eta\xi$) and the body-fixed frame ($G-xy$). Normally, the motion of the marine vehicle is represented in six DOFs (degree of freedoms), however, for a surface ship, the components of ship motion in the heave, pitch, and roll axes are neglected, as shown in Figure 1. The relationship between the two coordinate systems is represented by Equation (1).

$$\begin{aligned} \dot{\eta} &= u \cos \psi - v \sin \psi \\ \dot{\xi} &= u \sin \psi + v \cos \psi \\ \dot{\psi} &= r \end{aligned} \tag{1}$$

where u, v, r are the surge, sway, and yaw velocity, respectively, determined in the body-fixed coordinates system; Ψ is the ship heading.

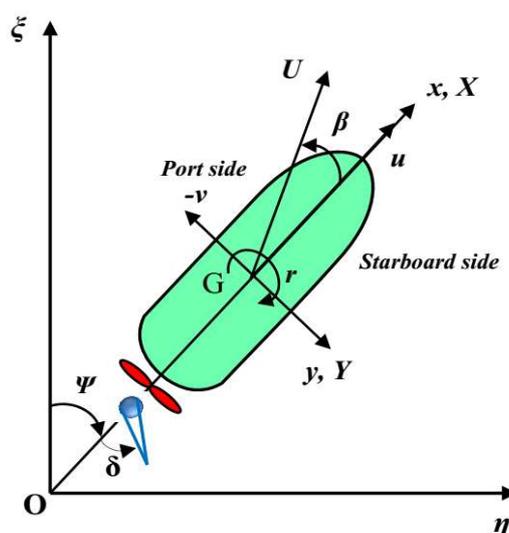


Figure 1. The ship coordinate system.

2.2. Ship Dynamic Model

For ship berthing, a third-order degree of freedoms (3-DOFs) model must be employed to express the dynamics of the ship on the surge, sway, and yawing axes in the body-fixed coordinate system. In this study, the Maneuvering Modeling Group (MMG) model was used to represent the ship motion in the berthing process, where the hydrodynamic forces and moments acting on the ship were divided into modular components such as the hull, rudder, and propeller. The motion equation of the ship based on the MMG model is presented as follows:

$$\begin{aligned}
 (m + m_x)\dot{u} - (m + m_y)vr &= X_H + X_P + X_R \\
 (m + m_y)\dot{v} + (m + m_x)ur &= Y_H + Y_R \\
 (I_{zz} + J_{zz})\dot{r} &= N_H + N_R
 \end{aligned}
 \tag{2}$$

where the subscripts *H*, *P* and *R* denote the components of hull, propeller, and rudder, respectively. The components of the hydrodynamic forces and moments acting on the ship hull are shown in Equation (3), which is proposed by Kijima (1990) [28].

$$\begin{aligned}
 X_H &= \frac{\rho}{2}LdU^2\left(X'_{\beta r} \sin \beta + X'_{uu} \cos^2 \beta \frac{rL}{U}\right) \\
 Y_H &= \frac{\rho}{2}LdU^2\left(Y'_{\beta\beta} + Y'_r \frac{rL}{U} + Y'_{\beta\beta}\beta|\beta| + Y'_{rr} \frac{rL}{U} \left|\frac{rL}{U}\right| + Y'_{\beta\beta r}\beta^2 \frac{rL}{U} + Y'_{\beta rr}\beta\left(\frac{rL}{U}\right)^2\right) \\
 N_H &= \frac{\rho}{2}L^2dU^2\left(N'_{\beta\beta}\beta + N'_r \frac{rL}{U} + N'_{\beta\beta}\beta|\beta| + N'_{rr} \frac{rL}{U} \left|\frac{rL}{U}\right| + N'_{\beta\beta r}\beta^2 \frac{rL}{U} + N'_{\beta rr}\beta\left(\frac{rL}{U}\right)^2\right),
 \end{aligned}
 \tag{3}$$

where $X'_{\beta r}$, X'_{uu} , ..., $N'_{\beta\beta r}$, and $N'_{\beta rr}$ are the hydrodynamic coefficients investigated by the method of the authors (Kijima (1990)) [28]; *L*, *d*, *U*, *r* and β are the ship’s length, the ship’s draft, its total velocity, its yaw velocity, and the ship’s drift; ρ is the density of water.

The hydrodynamic force due to the propeller and the propeller thrust force proposed by Kijima (1990) [28] can be considered as follows:

$$X_P = (1 - t_p)T \tag{4}$$

$$T = \rho D_p^4 rps^2 K_T(J)$$

The hydrodynamic forces and moments obtained by the rudder can be represented via the work of Kijima (1990) [28] as follows:

$$\begin{aligned}
 X_R &= -(1 - t_R)F_N \sin \delta \\
 Y_R &= -(1 + a_H)F_N \cos \delta \\
 N_R &= -(x_R + a_H x_H)F_N \cos \delta
 \end{aligned}
 \tag{5}$$

where

$$F_N = \frac{\rho}{2}f_\alpha(\Lambda)A_R U_R^2 \sin \alpha_R \tag{6}$$

In order to control the ship’s motion in the berthing process, the actuators including the rudder angle (δ) and the propeller speed (*rps*) were regulated simultaneously. For the numerical simulation, the training ship (named SAENURI) of Mokpo National Maritime University was used as the model ship. The particulars of this ship are shown in Table 1.

Table 1. The principle ship particulars.

Type	Training Ship
Length Overall	103 [m]
Length between perpendiculars	94 [m]
Breadth	15.6 [m]
Draft	5.4 [m]
Thruster (Bow)	49,000 [N]

The hydrodynamic coefficients predicted by the Kijima model are presented in Table 2.

Table 2. The ship hydrodynamic coefficients.

Coefficient	Value	Coefficient	Value
$X'_{\beta r}$	-0.5012	$Y'_{\beta rr}$	-0.3561
X'_{uu}	0.183	N'_{β}	0.1379
Y'_{β}	0.2496	N'_r	-0.0488
Y'_r	0.0542	$N'_{\beta\beta}$	-0.0782
$Y'_{\beta\beta}$	0.8755	N'_{rr}	-0.0430
Y'_{rr}	-0.0028	$N'_{\beta\beta r}$	-0.0480
$Y'_{\beta\beta r}$	0.8937	$N'_{\beta rr}$	-0.3789

2.3. Dynamic Feature in Ship Turning

When the ship’s rudder is turned to each side to alter the ship heading, there is a change in the ship dynamic velocities. The surge velocity (u) is similar in both cases. Meanwhile, the values of the sway velocity (v) and the yaw rate (r), when the ship heading alters to the starboard side, are approximate to those in case of the port side. Additionally, the sign of ($v r$) in the whole starboard turn is opposite with those in the port turn process. The expression and illustration of this feature are shown as Equation (7). This characteristic is used to propose a mechanism integrated for the controller. With this mechanism, a neural controller can be applied for berth-approaching on each side even if the controller is only trained by the teaching data to berth from one side.

$$\begin{aligned}
 u_{port} &\cong u_{starboard} \\
 v_{port} &\cong -v_{starboard} \\
 r_{port} &\cong -r_{starboard}
 \end{aligned}
 \tag{7}$$

where u_{port} , v_{port} , and r_{port} are the surge velocity, sway velocity, and yaw velocity, respectively, when the ship turns to the port side; $u_{starboard}$, $v_{starboard}$, and $r_{starboard}$ are the surge velocity, sway velocity, and yaw velocity, respectively, when the ship turns to the starboard side.

2.4. Head-Up Coordinate System

A head-up coordinate system was introduced by Im and Nguyen [11] to consider two new inputs for a neural controller instead of using the parameters including the geographical coordinates (η, ξ) and ship heading (Ψ) like in previous studies. Two parameters proposed in the head-up coordinate system, as illustrated in Figure 2, consist of the relative bearing (Ψ_{REL}) and the distance from the ship to the berth (D). Based on these inputs, the rudder (δ_{ord}) and propeller (rps) are determined properly by the neural controller. In combination with the distance to the imaginary line (d_1) and the remaining distance to the berth (d_2), four parameters (Ψ_{REL} , D , d_1 and d_2) are found in this coordinate system.

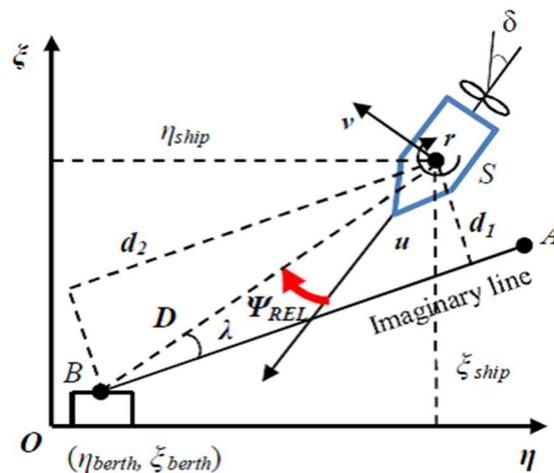


Figure 2. The coordinate parameters of inputs for the artificial neural network (ANN) controller.

To simulate the ship motion, Equations (1) and (2) are employed to obtain the ship states with respect to time. In every case, the ship dynamics (u, v, r) and parameters (d_1, d_2) are originally kept, and a data converter is suggested to transform the ship states (η, ξ, Ψ) into inputs considered as (Ψ_{REL}, D) in the head-up coordinate system for the proposed controller. The formula of the data converter is shown as in Equation (8).

$$\begin{aligned}
 D &= \sqrt{(\xi_{ship} - \xi_{berth})^2 + (\eta_{ship} - \eta_{berth})^2} \\
 BA &= kD \quad (k > 1) \\
 \lambda &= \cos^{-1} \left(\frac{\vec{BS} \cdot \vec{BA}}{|\vec{BS}| |\vec{BA}|} \right) \\
 d_1 &= D \sin(\lambda) \quad d_2 = D \cos(\lambda) \\
 \psi_{REL} &= \angle(\text{Heading line}, \vec{SB})
 \end{aligned} \tag{8}$$

where ψ_{REL} is defined as the directional angle (in degrees) from the ship heading to a straight line drawn from the ship $S(\eta_{ship}, \xi_{ship})$ to the berth $B(\eta_{berth}, \xi_{berth})$, and λ is the angle between the vector \vec{BS} and the vector \vec{BA} .

3. Multitasking System for Ship Berthing Based on an Integrated Neural Controller

3.1. Concept of a Multitasking Ship Berthing System Based on ANN

In theory, an ANN has no ability to determine the control signal well when the initial inputs entered for the network are entirely different from those considered in the teaching data. This means that the ANN controller cannot work when the initial conditions of the inputs are different from those in the teaching data. In the previous studies, the inputs of the teaching data always contain the geographical coordinates of the ship (η_1, ξ_1). Because the geographical coordinates of the ship (η_1, ξ_1) in the trained port are always different from those (η_2, ξ_2) in other ports, the ability of the ANN to perform different tasks of ship berthing is limited. To overcome that problem, we use two parameters: relative bearing (Ψ_{REL}) and distance from the ship to berth (D).

First, in the original port, teaching data consisting of the $\Psi_{REL1}, D_1, d_{1(1)}, d_{2(1)}, u_1, v_1, r_1, \delta_{ord}$, and rps are generated to train an artificial neural network. This neural network is applied as a main controller, and is a basis to construct an integrated neural controller. In other ports, where the initial conditions of inputs ($\Psi_{REL2}, D_2, d_{1(2)}, d_{2(2)}, u_2, v_2, r_2$) are similar to those ($\Psi_{REL1}, D_1, d_{1(1)}, d_{2(1)}, u_1, v_1,$

r_1) achieved in the teaching data of the original port, the trained ANN can calculate the rudder angle (δ_{ord}) and propeller speed (rps) to bring the ship into the berth successfully, as illustrated in Figure 3.

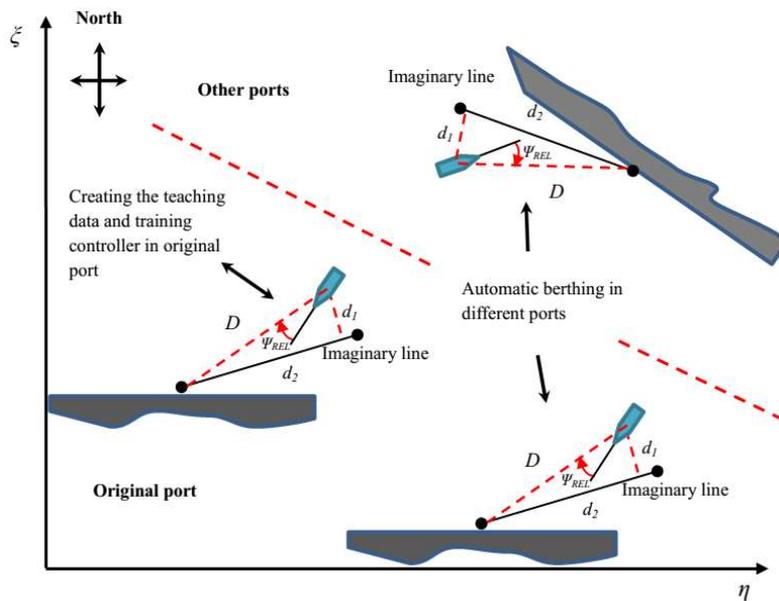


Figure 3. Automatic ship berthing for different ports.

Second, it assumes that teaching data including the $\Psi_{REL1}, D_1, d_{1(1)}, d_{2(1)}, u_1, v_1, r_1, \delta_{ord}$ and rps are obtained, in which the ship approaches the berth from the port side. Because the dynamic feature of the ship on the port side is different to that of the starboard side, the trained neural networks in previous studies, as well as in the first part of this research, cannot berth from the starboard side of the ship. To solve this problem, an integrated mechanism with three negative signs is linked to the main controller to achieve an integrated neural controller where two negative signs are added to the velocities (u and v), and one negative sign is connected to the rudder angle. For the starboard side, where the initial conditions of inputs ($\Psi_{REL3}, D_3, d_{1(3)}, d_{2(3)}, u_3, v_3, r_3$) are similar to those of the port side ($\Psi_{REL1}, D_1, d_{1(1)}, d_{2(1)}, u_1, v_1, r_1$), the integrated neural controller can berth successfully based on the changing of the signs of velocities (u, v) and the rudder angle (δ_{ord}), as illustrated in Figure 4.

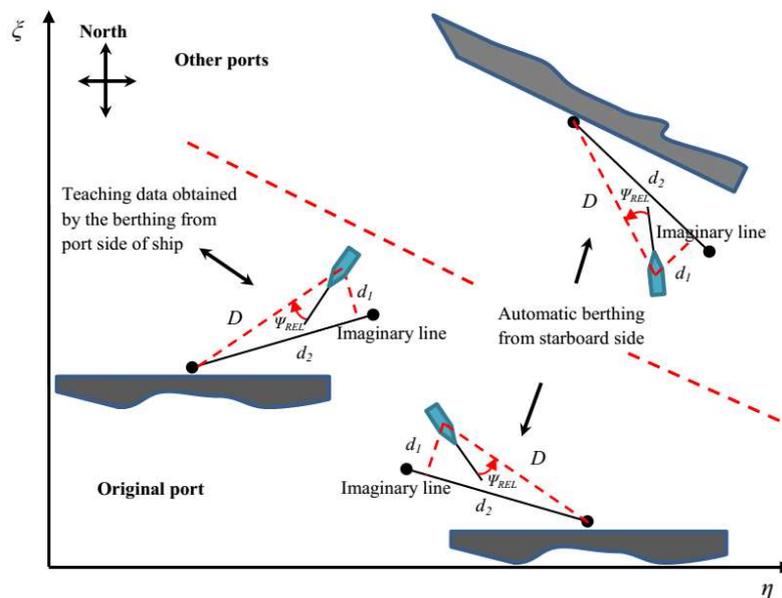


Figure 4. Automatic ship berthing for the berth-approaching on each side.

3.2. Artificial Neural Network Trained by Teaching Data in the Original Port

In this section, a neural network of multi-layer perception (MLP) was selected for the ANN controller, which had seven inputs including the Ψ_{REL} , d_1 , v , r , u , D , and d_2 , and two outputs consisting of the rudder angle (δ_{ord}), and the revolution rounds (rps). The structure of the network was divided into two separate parts in a hidden layer, as illustrated in Figure 5a.

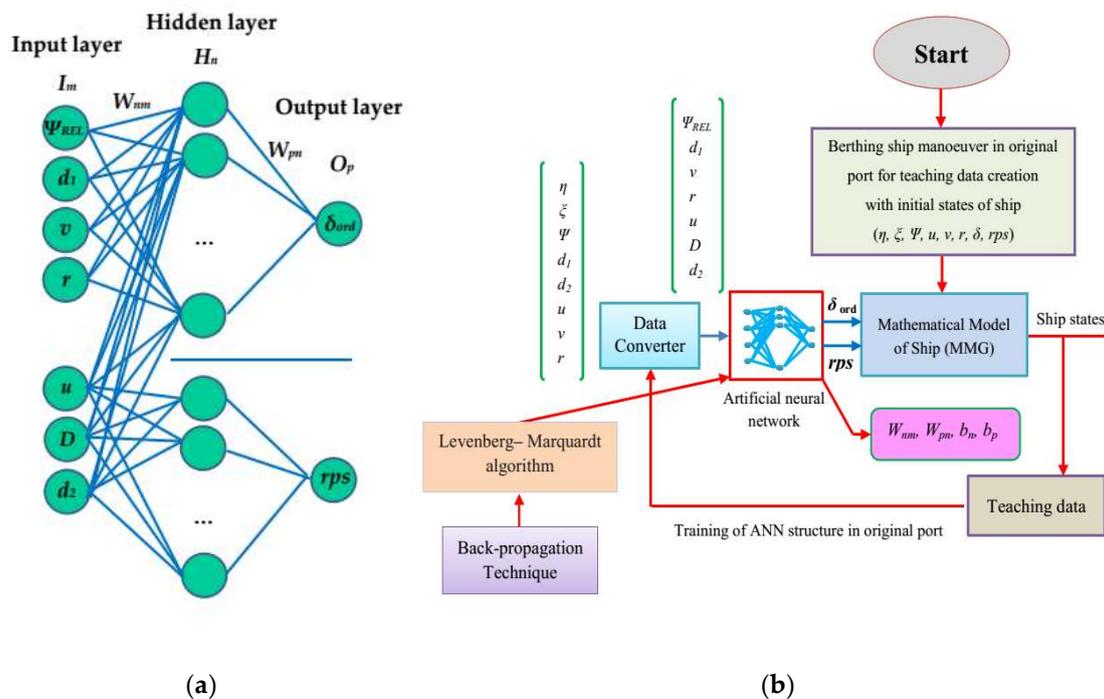


Figure 5. The scheme of training an ANN in the original port: (a) the structure of an ANN; (b) the method of training an ANN.

As shown in Figure 5b, a berthing ship’s manoeuvre parameters were used to generate teaching data, and then a data converter was applied to collect the parameters in a head-up coordinate system. A back-propagation technique with a training function of the Levenberg–Marquardt algorithm was applied to train the ANN and to obtain weights and thresholds. The objective of training is to minimize the mean square error (MSE) between the outputs of the network and the desired outputs provided by the teaching data. The training of the ANN was installed as an off-line mode to learn directly the samples in teaching data.

The number of nodes in each layer of the network is assumed to be m , n and p . The teaching data consist of samples expressed in the following form: $\{q_1, l_1\}, \{q_2, l_2\}, \dots, \{q_i, l_i\}$, where q_i and l_i are the inputs of the network and the target outputs, respectively. The input vector of the ANN is denoted as $I_m = [\Psi_{REL}, d_1, v, r, u, D, d_2]^T$.

The output of each node at the hidden layer (H_n) can be written as follows:

$$H_n = f_1(net_n) = f_1\left(\sum_{m=1}^m W_{nm}I_m + b_n\right) \tag{9}$$

A tansig-function was chosen as a transfer function at the hidden layer, the details of which are as follows:

$$f_1(net_n) = \frac{2}{1 + \exp(-2net_n)} - 1 \tag{10}$$

where f_1 is a transfer function at the hidden layer and net_n is an input at the n -th node of the hidden layer. W_{nm} and b_n are the weights on the link from node n to node m and the threshold of the hidden layer (H_n).

Similarly, the output of each node at the output layer (O_p) is written as Equation (11)

$$O_p = f_2(net_p) = f_2\left(\sum_{n=1}^n W_{pn}H_n + b_p\right) \tag{11}$$

A satlin-function is employed as the active function at the output layer, and the detail of this function is as follows:

$$f_2(net_p) = \begin{cases} -1, & \text{if } (net_p) \leq -1 \\ net_p, & \text{if } -1 \leq (net_p) \leq +1 \\ +1, & \text{if } (net_p) \geq +1 \end{cases} \tag{12}$$

where the f_2 and net_p are a transfer function at the output layer and an input at the p -th node of the output layer, respectively. W_{pn} and b_p are the weights on the link from node p to node n and the threshold of output layer (O_p).

The goal function (MSE) to train the ANN is represented as follows:

$$MSE = \frac{1}{2} \sum_{i=1}^i \sum_{p=1}^p \left\{ desired_O_p - f_2\left(\sum_{n=1}^n W_{pn}f_1\left(\sum_{m=1}^m W_{nm}I_m + b_n\right) + b_p\right) \right\}^2 \tag{13}$$

The back-propagation technique, which is a gradient descent algorithm, is applied when the network weights are moved along the negative of a gradient of a performance function. Then, the Levenberg–Marquardt algorithm is used to approach the second-order training speed without computing the Hessian matrix as required in a quasi-Newtonion method. When the performance function has the form of a sum of squares, then the Levenberg–Marquardt algorithm uses the following approximation for the Hessian matrix in order to follow a Newton-like update:

$$W_{k+1} = W_k - [J^T(W_k)J(W_k) + \mu I]^{-1} J^T(W_k)E(W_k) \tag{14}$$

where W contains of all weights of the network, J is the Jacobian matrix, I is the identity matrix.

After the training process of an ANN, the control signals and ship states are determined by the following equations

$$[\delta_{ord(t+1)}, rps_{t+1}]^T = f_2\left(\sum_{n=1}^n W_{pn}f_1\left(\sum_{m=1}^m W_{nm}[\psi_{REL(t)}, d_{1(t)}, v_t, r_t, u_t, D_t, d_{2(t)}]^T + b_n\right) + b_p\right) \tag{15}$$

$$[\eta_{t+1}, \xi_{t+1}, \psi_{t+1}, u_{t+1}, v_{t+1}, r_{t+1}]^T = f_{MMG}(\delta_{ord(t+1)}, rps_{t+1}) \tag{16}$$

where f_{MMG} is a function denoted for the MMG model.

This procedure is carried out continuously during the berthing process of a ship. In this study, a MATLAB Neural Network Toolbox, where varieties of training functions with different basic algorithms are available to train the network, is used to train the proposed controller, and to simulate the automatic ship berthing. The Levenberg–Marquardt algorithm is one of the training functions, and it is easy to get the syntax of this algorithm in MATLAB.

With the rapid advance of science, technology and theories, such as machine learning methods, deep learning methods, etc., have been developed deeply and widely to carry out advanced tasks with help from artificial intelligence. For automatic ship berthing, a MLP neural network is commonly used to learn the teaching data and to imitate the actions of a human brain in performing the stages of ship berthing. In this study, the purpose is to construct a MLP-based ship berthing system without

retraining the structure of the neural network; therefore, other approaches, such as a long short-term memory network (LSTMN), recurrent neural network (RNN), convolutional neural network (CNN), and so on, will be investigated in the future.

3.3. Integrated Neural Controller

The proposed ANN controller was trained in the original port, where the ship approaches to the berth on her port side with the initial conditions of a ship at starting time as $(\Psi_{REL1}, d_{1(1)}, v_1, r_1, u_1, D_1, d_{2(1)}, \delta_1, rps_1)$. Figure 6 shows the structure of a multitasking berthing system and integrated neural controller for different tasks for automatic ship berthing.

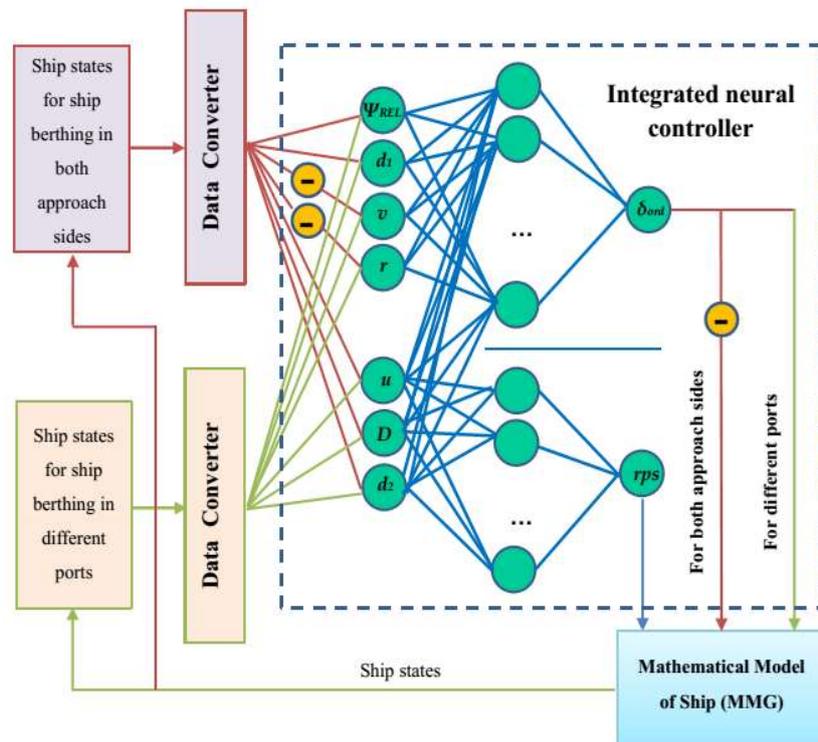


Figure 6. The structure of the multitasking berthing system and the integrated neural controller.

For the first task (as shown in Figure 3), the integrated neural controller calculates the rudder angle and the revolution speed continuously to control the ship into the berth. From the inputs such as the $\Psi_{REL1}, d_{1(1)}, v_1, r_1, u_1, D_1$ and $d_{2(1)}$, the rudder angle and the revolution speed at second time are determined. The new rudder angle (δ_2) and revolution speed (rps_2) make the ship motion dynamics change from the ship states $(\Psi_{REL1}, d_{1(1)}, v_1, r_1, u_1, D_1, d_{2(1)})$ into new ones $(\Psi_{REL2}, d_{1(2)}, v_2, r_2, u_2, D_2, d_{2(2)})$.

Generally, the ship states $(\Psi_{REL(t)}, d_{1(t)}, v_t, r_t, u_t, D_t, d_{2(t)})$ are used to find out the rudder angle (δ_{t+1}) and revolution speed (rps_{t+1}) in next time by Equation (15). Besides that, Equation (15) can be rewritten in the following form

$$\begin{cases} \delta_{t+1} = f_{ANN}(\Psi_{REL(t)}, d_{1(t)}, v_t, r_t, u_t, D_t, d_{2(t)}) \\ rps_{t+1} = g_{ANN}(\Psi_{REL(t)}, d_{1(t)}, v_t, r_t, u_t, D_t, d_{2(t)}) \end{cases} \quad (17)$$

where f_{ANN} and g_{ANN} are operators denoted as separate outputs of the integrated neural controller.

For the second task (as shown in Figure 4), when the ship berths from its starboard side with initial states similar to those in the teaching data, the integrated controller with three negative signs calculates the rudder angle and the revolution speed continuously to berth from its starboard side.

From the inputs such as the Ψ_{REL1} , D_1 , $d_{1(1)}$, $d_{2(1)}$, u_1 , v_1 , and r_1 , the rudder angle (δ_2) and the revolution speed (rps_2) at second time are determined as the outputs of the controller. To approach the berth, the rudder angle (δ_2) needs to be altered oppositely to that angle in the teaching data, as in Figure 7. Besides that, the ship dynamics of the starboard side ($-v_2, -r_2$) are also opposite to that of the port side (as shown in Equation (7)); therefore, it is necessary to invert the sign of these states ($-v_2, -r_2 \delta_2$) for a successful berthing. To achieve successful ship berthing on each side, three negative signs are linked to this controller where two of them are added in front of the inputs ($v r$), and another is placed behind the output (δ), as illustrated in Figure 6. Similarly, the integrated neural controller determines the control signals for all the steps to approach the berth from its starboard side by the following equations

$$\begin{cases} \delta_{t+1} = -f_{ANN}(\Psi_{REL(t)}, d_{1(t)}, -v_t, -r_t, u_t, D_t, d_{2(t)}) \\ rps_{t+1} = g_{ANN}(\Psi_{REL(t)}, d_{1(t)}, -v_t, -r_t, u_t, D_t, d_{2(t)}) \end{cases} \quad (18)$$

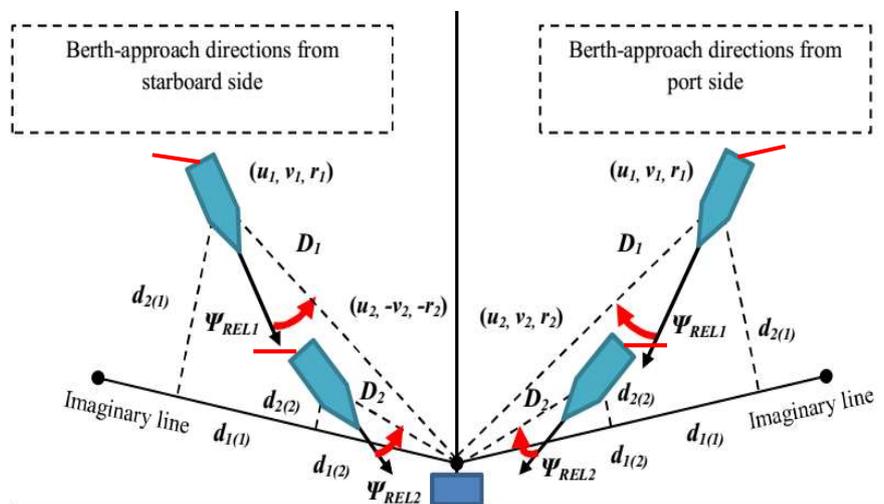


Figure 7. The approach of the berth on each side of ship.

4. Numerical Simulation and Result

In this section, numerical simulations were carried out in the MATLAB environment to verify the effectiveness of the multitasking system for automatic ship berthing. First, the teaching data are generated to train the integrated neural controller in the original port with the initial states of the ship by the maneuvering process, in which the ship approaches to the berth from its port side. Then, the proposed neural controller is simulated to perform different tasks without the retraining process of an ANN.

For the simulation, the geographical coordinates of all the ports with a latitude and longitude need to be normalized to a non-dimensional form. Because the non-dimensional forms of these ports were different, these ports were entirely different. The geographical coordinates of the ports and the ship position were normalized into the non-dimensional form. Generally, input variables are transformed into the non-dimensional form by following Equation [27].

$$\begin{aligned} \eta' &= \frac{Longitude}{L}, \quad \xi' = \frac{Latitude}{L} \\ D' &= \frac{D}{L}, \quad d_{r1} = \frac{d_1}{L}, \quad d_{r2} = \frac{d_2}{L} \\ u' &= \frac{u}{\sqrt{Lg}}, \quad v' = \frac{v}{\sqrt{Lg}} \end{aligned} \quad (19)$$

$$r' = \frac{r}{\sqrt{L/g}}, \psi'_{REL} = \psi_{REL}$$

where L is the length of the ship; g is the gravitational acceleration; $\eta', \xi', D', d'_{1}, d'_{2}, u', v', r',$ and ψ'_{REL} are the longitude, latitude, distances, velocities, and relative bearing in non-dimensional form, respectively.

4.1. Teaching Data Generation and Training of ANN in the Original Port

As shown in Figure 8, the teaching data generated in the original port consist of six cases of maneuvering patterns which are conducted to bring the ship into the berth. The range of latitude is from -2 to 8 , and the range of the longitude is from -2 to 10 . Normally, the ship berthing is considered as a successful process when the ship reaches the berth within 0.2 m/s and the ship's heading is parallel to the berth.

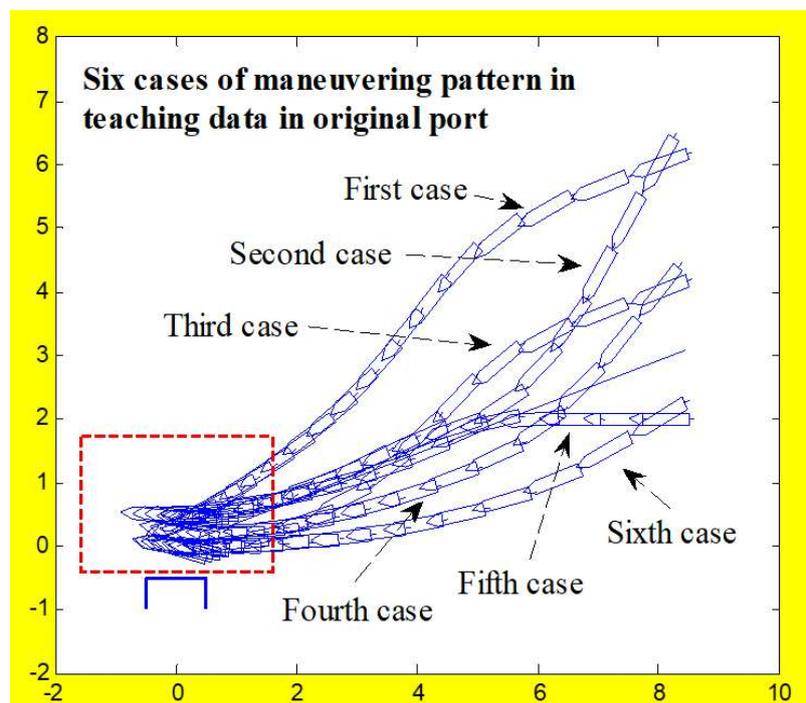


Figure 8. Teaching data generation with six samples of ship maneuvering.

The initial states of the ship in the original port include the ship position, ship heading, ship velocities, and the rudder and propeller speeds as the following set: $\eta_1/L, \xi_1/L, \Psi_1, u_1, v_1, r_1, rps, \delta_{ord}$. By using the data converter, these states are transformed into the $\Psi_{REL}, D/L, d_1/L, d_2/L, u_1, v_1, r_1, rps,$ and δ_{ord} , which are used to train the neural network. In the berth maneuvering process, all the time series of the teaching data are recorded.

In this research, six initial states of the ship in the original port were examined: $(8, 6, 250, 1.5, 0, 0, 0.75, 0), (8, 6, 210, 1.5, 0, 0, 0.75, 0), (8, 4, 220, 1.5, 0, 0, 0.75, 0), (8, 4, 250, 1.5, 0, 0, 0.75, 0), (8, 2, 240, 1.5, 0, 0, 0.75, 0),$ and $(8, 2, 270, 1.5, 0, 0, 0.75, 0)$. The corresponding teaching data had the following initial conditions, respectively: $(16.9, 10, 2.9, 9.6, 1.5, 0, 0, 0.75, 0), (-23.1, 10, 2.9, 9.6, 1.5, 0, 0, 0.75, 0), (-23.4, 8.9, 1.02, 8.88, 1.5, 0, 0, 0.75, 0), (6.6, 8.9, 1.02, 8.88, 1.5, 0, 0, 0.75, 0), (-5.96, 8.2, 0.85, 8.2, 1.5, 0, 0, 0.75, 0),$ and $(14, 8.2, 0.85, 8.2, 1.5, 0, 0, 0.75, 0)$.

The teaching data were normalized into values between 0 and 1 to train the ANN controller.

Before training the ANN, a min–max normalization is applied to normalize all the variables of the teaching data for the range (from 0 to 1) which ensures the convergence of the training algorithm for the ANN. The rule of the min–max normalization is represented as the following equation

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)} \tag{20}$$

where $x_i = (x_1, \dots, x_n)$ are the elements of data; z_i is the corresponding normalized data; $\max(x)$ is the maximum value of x_i ; $\min(x)$ is the minimum value of x_i .

For the training, the parameters for the structure of the ANN are shown as in Table 3. The result of training of ANN is presented as Figure 9.

Table 3. The parameters of the structure of the neural network.

Type	Details
Type of neural network	Multi-Layer Perception (MLP)
Training technique	Back-propagation
Training function	Levenberg–Marquardt algorithm
Number of layers	01 Input layer, 01 Hidden layer, 01 Output layer
Number of Inputs	07, $I_m = [\Psi_{REL}, d_1, v, r, u, D, d_2]^T$
Number of Outputs	02, $O_p = [\delta_{ord}, rps]^T$
Number of hidden layer	21 nodes

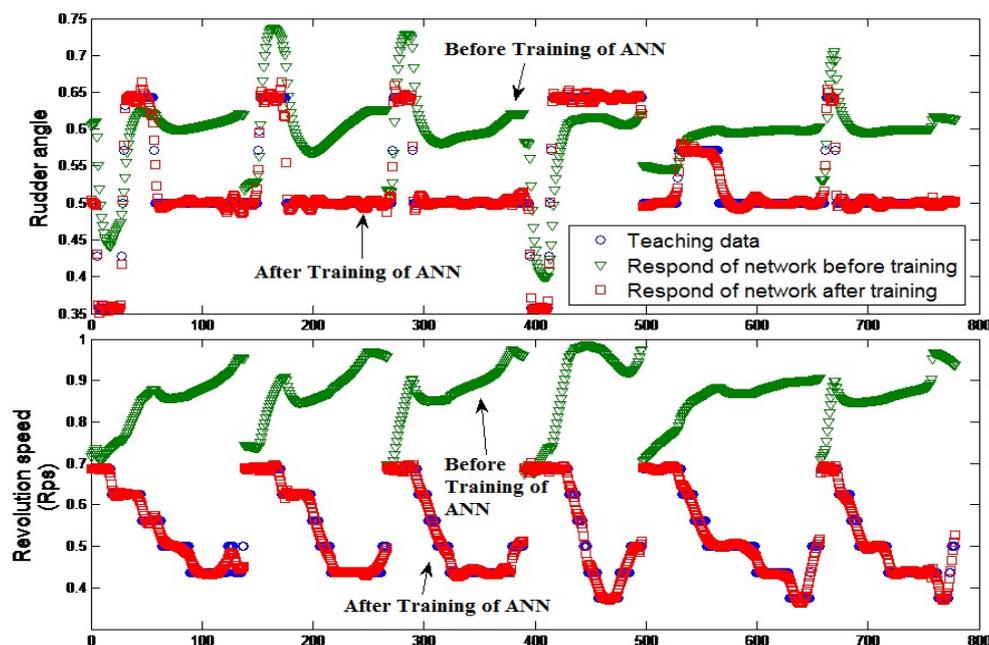


Figure 9. The result of training the ANN with teaching data in the original port.

After training, the detail of the data, such as the teaching data, testing data, and validation, are synthesized as shown in Table 4.

Table 4. The list of teaching data/validation data/testing data.

Type of Data	Amount
Teaching data	06 (as shown in Figure 8)
Validation data	06 (03 cases are presented in Figure 10)
Testing data	05 (cases are tested for initial conditions which are different to those in teaching data)

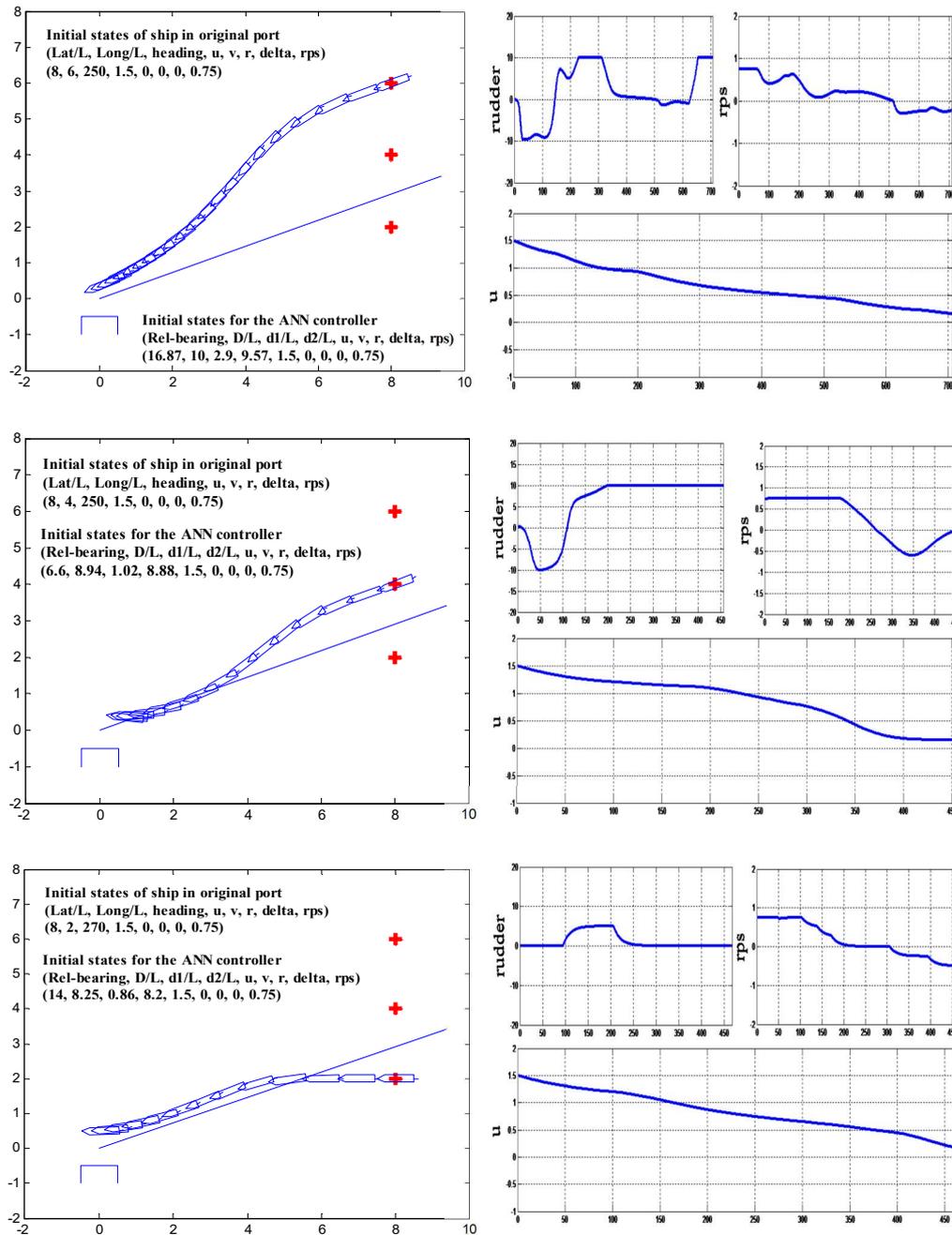


Figure 10. Automatic ship berthing in the original port.

4.2. Verification of the Integrated Neural Controller for the Original Port and Different Ports

For this task, berthing simulations were carried out for the original port, a different port, and two terminals of a real port.

As shown in Figure 10, all cases were successful in the original port where the ship approaches to the berth within 0.2 m/s and is also parallel to the berth. These results proved the learning ability of an ANN from teaching data.

Figure 10 indicates the three cases of simulation. In the first case, the ship states started with (8, 6, 250, 1.5, 0, 0, 0, 0.75), the rudder was turned to -10 degrees and the propeller was reduced. The second case started with (8, 4, 250, 1.5, 0, 0, 0, 0.75), the controller brought the ship into berth safely. In the third case, the ship states started with the initial conditions (8, 2, 270, 1.5, 0, 0, 0, 0.75), and the

controller performed the actions to bring the ship into berth. The results showed a good performance of this controller in the original port.

The different port was validated for the proposed system, in which the range of latitude is from 2 to 12, and the range of the longitude is from -15 to -2 . The ship headings in all cases in this simulation were also different to those in the original port.

The results of the automatic ship berthing in a different port are presented in Figure 11. In the first case, the ship started with the initial state of $(-13.5, 4, 70, 1.5, 0, 0, 0, 0.75)$; the rudder was turned to -10 degrees and the propeller was reduced. The second case started with the initial state of $(-13.5, 6, 70, 1.5, 0, 0, 0, 0.75)$, and the controller brought the ship into berth. In the third case, the ship started with the initial conditions of $(-13.5, 8, 80, 1.5, 0, 0, 0, 0.75)$, and the rudder angle was turned to $+05$ degrees to bring the ship into berth. The results show a successful berthing of the ship in the different port which is entirely different to the original port's geographical coordinates.

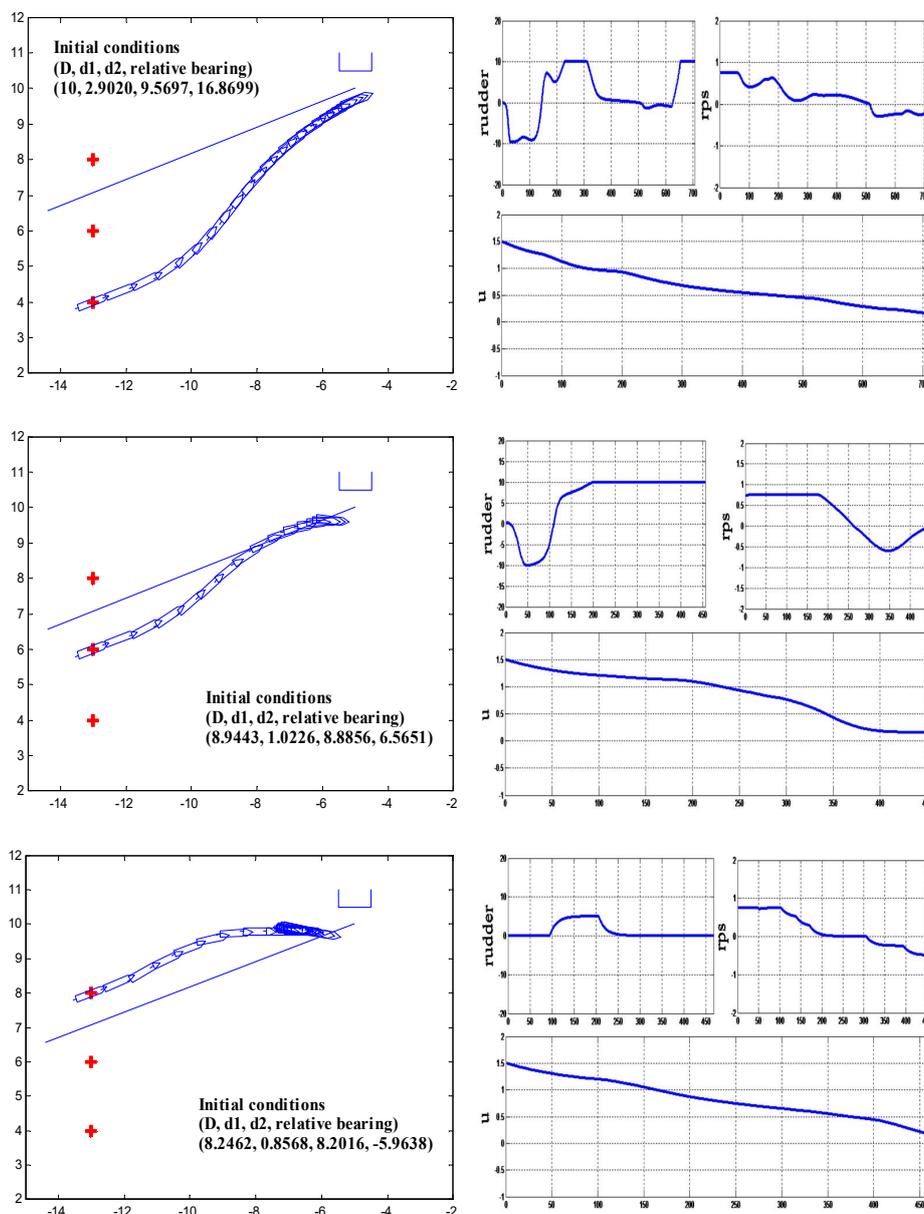


Figure 11. Automatic ship berthing in the different port.

To verify the proposed controller for a real port, the ship was simulated for the automatic berthing in the Gamcheon area of Busan Harbor, where two other terminals were applied. These terminals are named No.4 Wharf and No.6 Wharf, as shown in Figure 12. The range of the latitude is from 35°2' N to 35°5'56" N, and the longitude is from 128°58' E to 129°2'36" E. The results of ship berthing in these two terminals are synthesized in Figure 13.



Figure 12. Gamcheon area of the Busan port for ship berthing.

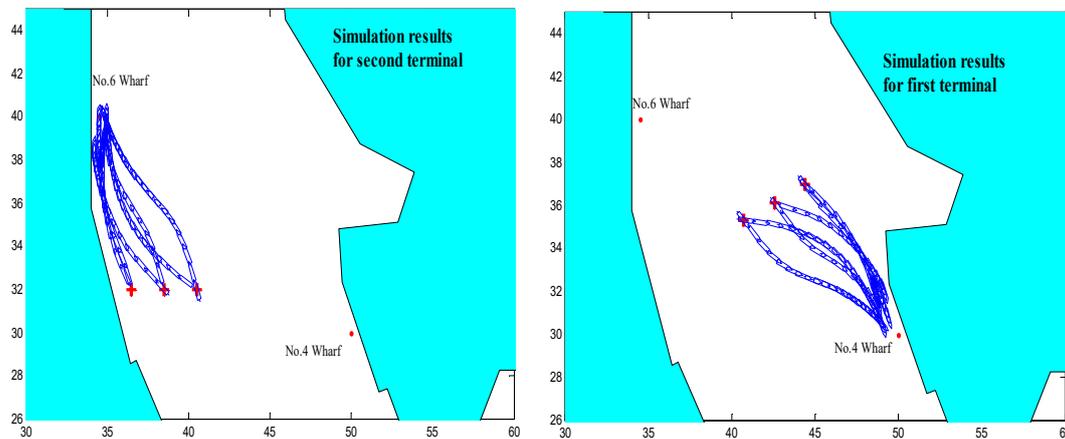


Figure 13. Automatic ship berthing for two terminals of the Busan port.

4.3. Verification of Integrated Neural Controller for Berth Approaching on Each Side

In the first section of the numerical simulation, the integrated neural controller controlled the ship into the berth automatically, but the results were only achieved when the ship approached the berth from its port side. In this section, numerical simulations for ship berthing are performed for when the ship approaches from its starboard side.

In previous studies, the controllers must be retrained by a new teaching data to berth while the integrated neural controller with three negative signs in the ANN structure can control the ship into

the berth from its starboard side, without retraining the ANN. Simulations are also applied in the original port, the different port, and two terminals of a real port.

As shown in Figure 14, the automatic ship berthing is conducted for the original port, in which the ship position and ship heading at the starting time in all cases in this simulation are symmetrical to those in teaching data. The time history shows that the rudder angles, the sway velocity and yaw rate are the opposite to those in teaching data. In the first case, the ship states started with $(-8, 6, 110, 1.5, 0, 0, 0, 0.75)$; the rudder was turned to $+10$ degrees and the propeller was reduced. The second case started with $(8, 4, 110, 1.5, 0, 0, 0, 0.75)$, and the controller brought the ship into berth safely. In the third case, the ship started with the initial conditions of $(8, 2, 100, 1.5, 0, 0, 0, 0.75)$, and the controller performed the actions to bring the ship into berth. The results showed a good performance of this controller for berthing in the original port.

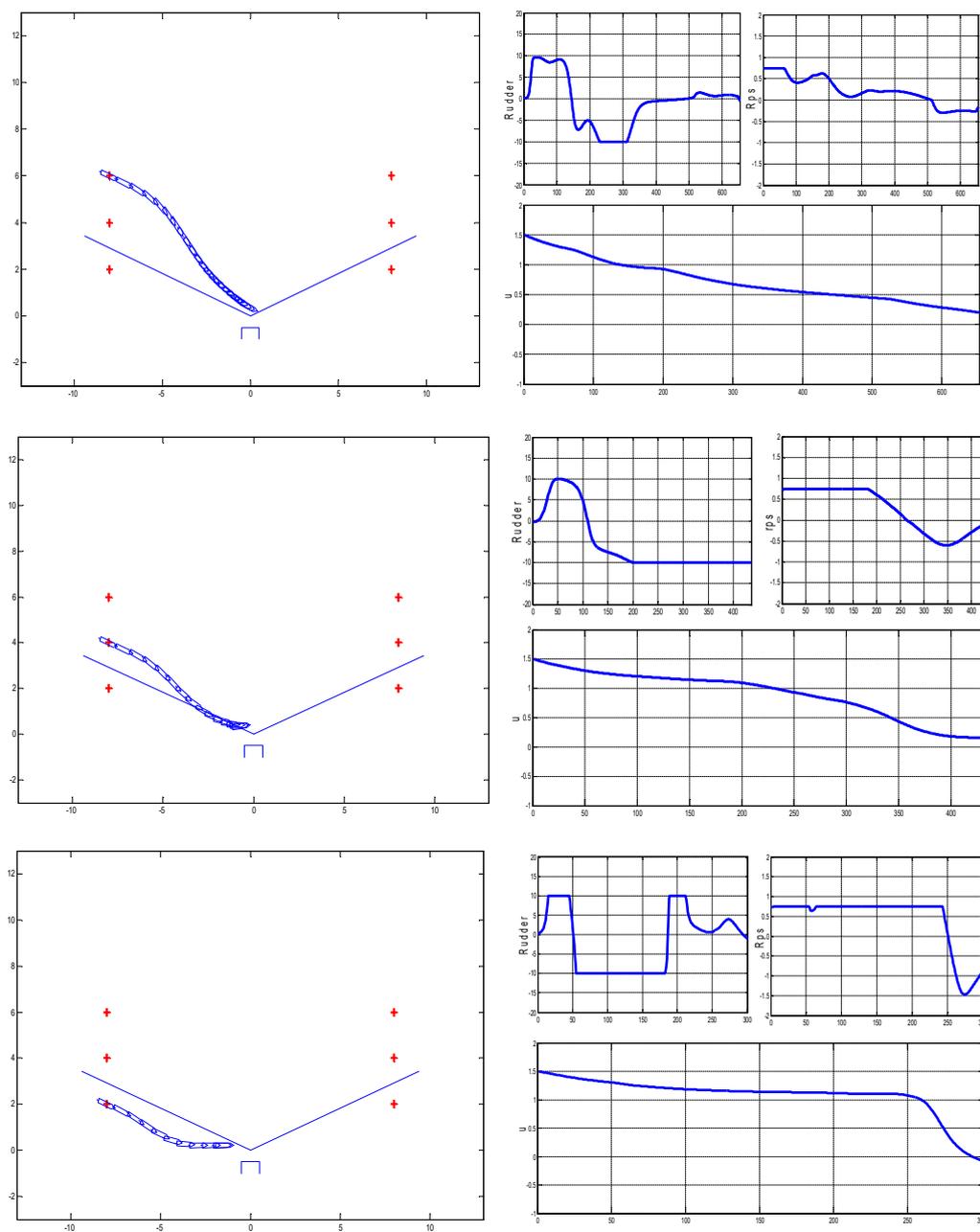


Figure 14. Automatic ship berthing in the original port from the starboard side.

In Figure 15, the automatic ship berthing carried out for the different port, in which the ship position and ship heading at starting time in all cases in this simulation were different to those in the original port. In the first case, the ship started with an initial state of $(3, 4, 290, 1.5, 0, 0, 0, 0.75)$, the rudder was turned to +10 degrees and the propeller was reduced. The second case started with $(3, 6, 290, 1.5, 0, 0, 0, 0.75)$, and the controller brought the ship into berth. In the third case, the ship started with the initial conditions $(3, 8, 280, 1.5, 0, 0, 0, 0.75)$, and the rudder angle was turned to +10 degrees to bring the ship into berth. The results presented a successful berthing into the different port, which is entirely different from the original port with regards to its geographical coordinates.

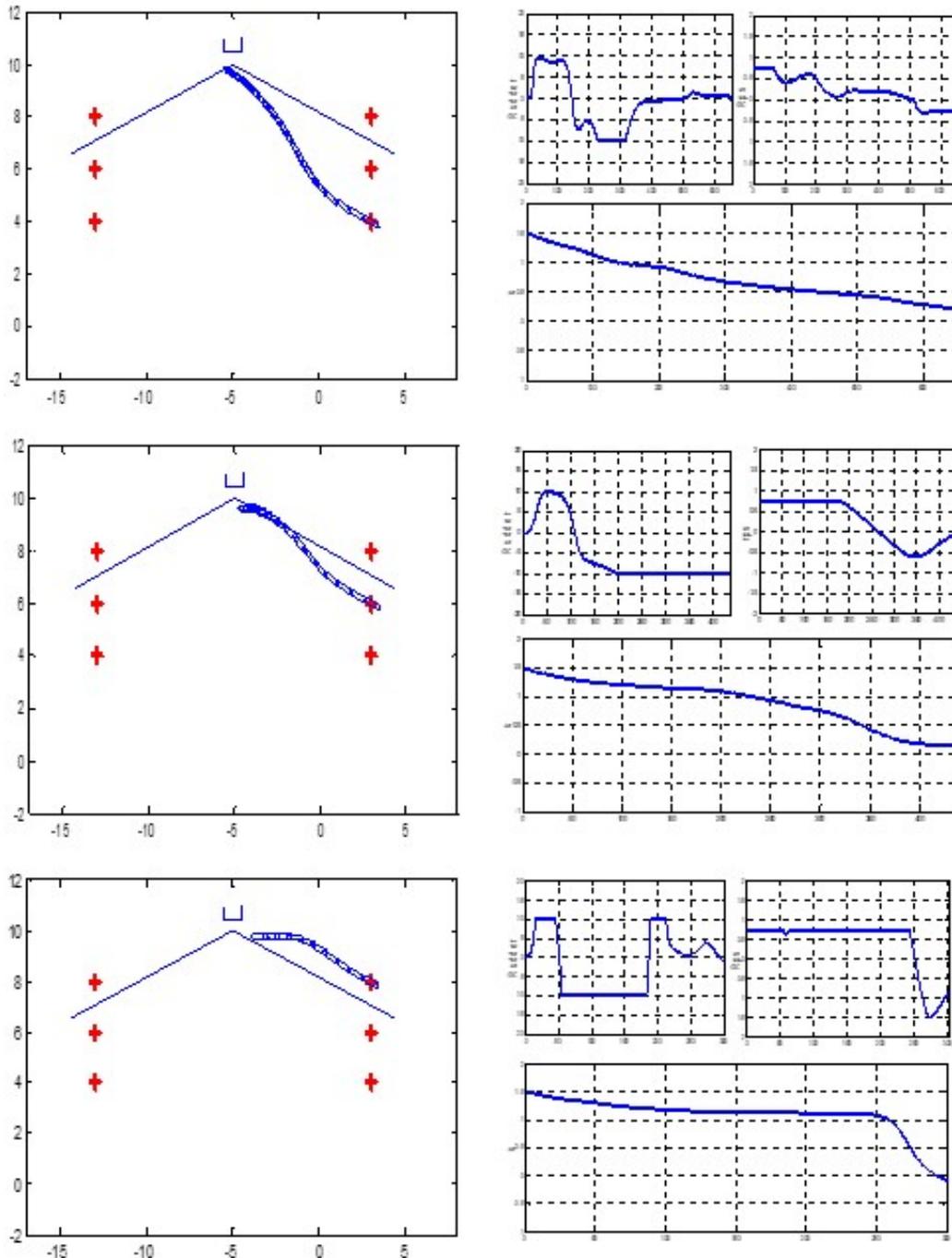


Figure 15. Automatic ship berthing in the different port from the starboard side.

Figure 16 showed the results of ship berthing at No. 4 Wharf: the ship approached this terminal from the starboard side. All the cases showed that the proposed neural controller performs well at berthing for both approach sides.

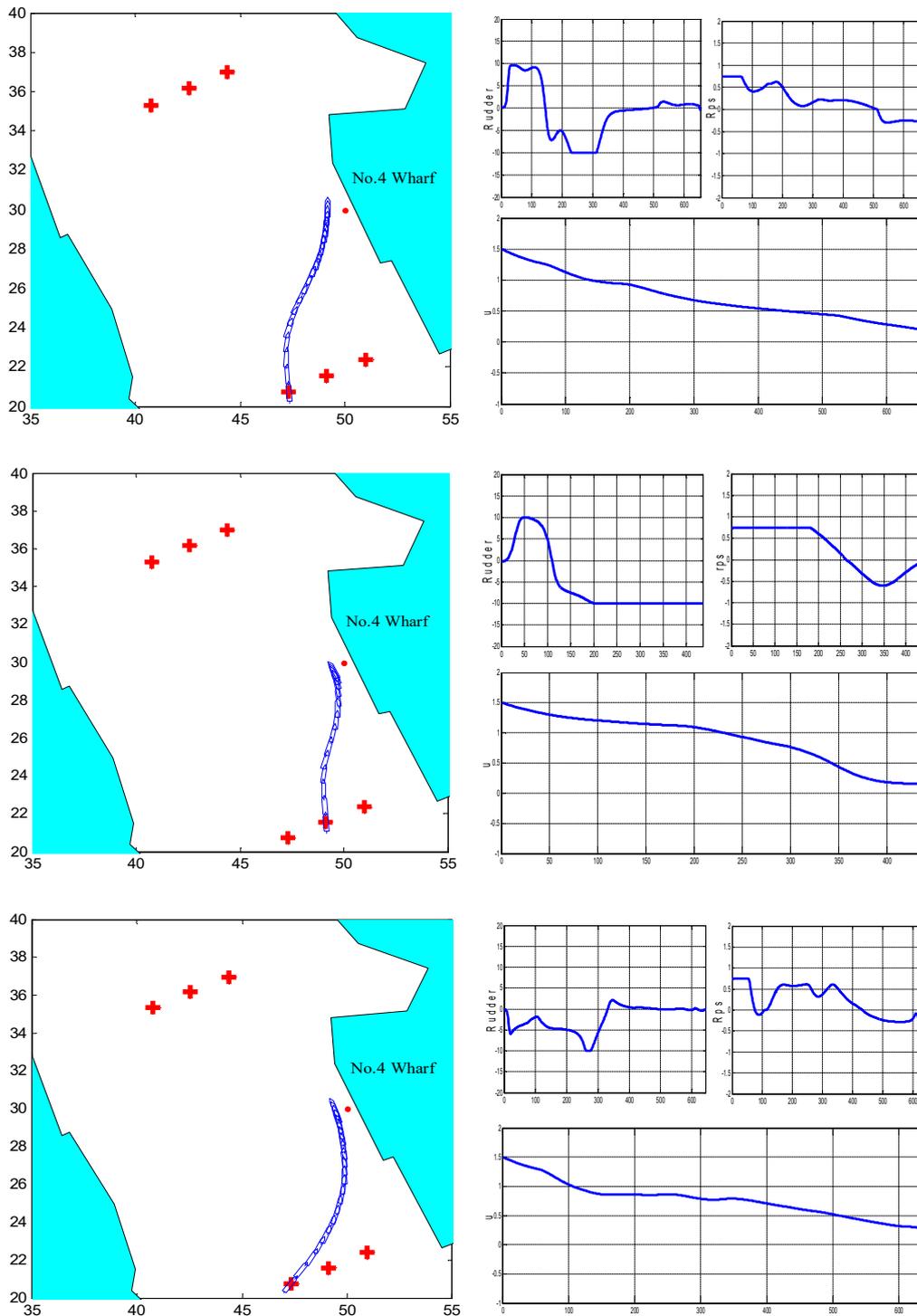


Figure 16. Automatic ship berthing for No.4 Wharf approaching from the starboard side.

Similarly, Figure 17 indicated the results of ship berthing at No. 6 Wharf. The results of this simulation were successful for the starboard side.

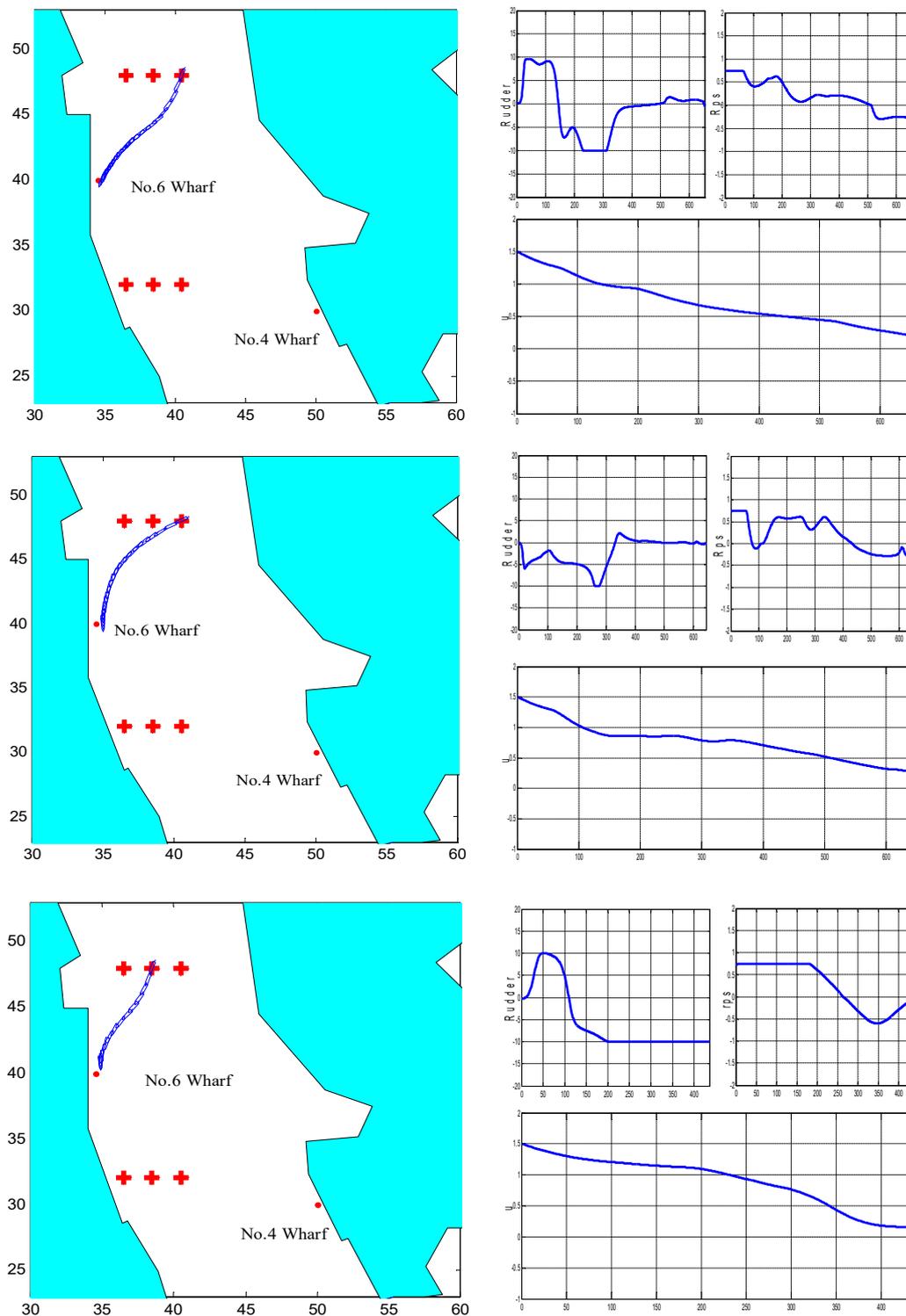


Figure 17. Automatic ship berthing for No.6 Wharf approaching from the starboard side.

In this section, numerical simulations were carried out to validate the effectiveness of the multitasking berthing system of the ship based on the integrated neural controller without retraining the ANN structure. The results indicated that the proposed controller performs well at conducting different tasks related to ship berthing problems. In the future, the author will continue to develop this system considering different tasks, such as combining it with an automatic collision avoidance system, berthing the ship in a restricted depth, and so on.

5. Conclusions

In this study, a multitasking system for automatic ship berthing was proposed based on an integrated neural controller without the retraining stages of an ANN to carry out the different tasks of ship berthing. An integrated neural controller is a combination of an artificial neural network and integrated mechanism. By using the proposed system, the ship can be controlled automatically for different ports and from both berth-approach sides without retraining the controller structure. Compared to previous studies, this system does not demand the retraining of ANN; therefore it saves time and cost when performing different tasks of ship berthing. In the future, the complex effect of disturbances, such as the wind, current, and waves, will be also investigated for this system. This study uses applicable approaches to perform real experiments of automatic ship berthing at scale for ships in the future.

Funding: This research received no external funding.

Acknowledgments: The author thanks the editors and the anonymous reviewers for their constructive comments, which helped us to improve the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

ANN	Artificial Neural Network
W_{nm}	Weight on the link from node n to node m
W_{pn}	Weight on the link from node p to node n
b_n	Threshold of hidden layer (H_n)
b_p	Threshold of output layer (O_p)
L	Ship length between perpendiculars
B	Breadth of a ship
d	Draft of ship
ρ	Water density
U	Total speed of ship
u, v, r	Surge, sway and yaw velocity of ship
δ	Rudder angle of ship
δ_{ord}	Control signal for rudder angle of ship
$rps(n)$	Revolution speed of propeller
D	Distance from ship to berth
$d-1$	Distance from ship to berth
$d2$	Distance from ship to imagination line
Ψ	Ship heading
Ψ_{REL}	Relative bearing from ship to berth
β	Drift angle of ship
m	Mass of ship
m_x	Added mass in surge direction
m_y	Added mass in sway direction
I_{zz}	Mass moment of inertia
J	Propeller advanced ratio
J_{zz}	Added mass moment of inertia
t_P	Effective thrust deduction factor by rudder to rudder normal force
AR	Profile area of moveable part of rudder
T	Propeller thrust force
$KT(J)$	Thrust coefficient of the propeller
D_p	Propeller diameter
a_H	Ratio of lateral force induced on hull by rudder to rudder normal force
t_R	Steering resistance deduction factor
UR	Longitudinal inflow velocity component
XH, YH, NH	Hydrodynamic forces and moment acting on the hull of ship

XR, YR, NR	Hydrodynamic forces and moment acting on the ship's rudder
XP	Hydrodynamic force acting on propeller
FN	Rudder normal force
$f\alpha$ (Λ)	Rudder lift gradient coefficient
xH	Longitudinal coordinate of point of the additional lateral force

References

1. Yamato, H.; Koyama, T.; Uetsaki, H. Automatic Berthing by the Neural Controller. In Proceedings of the Ninth Ship Control Systems Symposium, Maryland Bethesda, MD, USA, 10–14 September 1990; Volume 3, pp. 183–201.
2. Yamato, H.; Koyama, T.; Nakagawa, T. Automatic Berthing Using the Expert System. *IFAC Proc. Vol.* **1992**, *25*, 173–184. [[CrossRef](#)]
3. Zhang, Y. Neural Network Approaches to a Class of Ship Control Problems (Part I, II). In Proceedings of the Eleventh Ship Control Systems Symposium, Southampton, UK, 14–19 April 1997; Wilson, P.A., Ed.; Volume 1, pp. 115–150.
4. Im, N.K.; Hasegawa, K. A study on automatic ship berthing using parallel neural controller. *J. Kansai Soc. Nav. Archit. Jpn.* **2001**, *236*, 65–70.
5. Im, N.K.; Hasegawa, K. Motion identification using neural networks and its application to automatic ship berthing under wind. *J. Ship. Ocean Eng. Jpn.* **2002**, *11*, 16–26.
6. Im, N.K.; Lee, S.K.; Hyung, D.B. An application of ANN to automatic ship berthing using selective controller. *Int. J. Mar. Navig. Saf. Sea Transp.* **2007**, *1*, 101–105.
7. Nguyen, P.-H.; Jung, Y.-C. Automatic Berthing Control of Ship Using Adaptive Neural Networks. *J. Korean Navig. Port Res.* **2007**, *31*, 563–568. [[CrossRef](#)]
8. Ahmed, Y.A.; Hasegawa, K. Automatic Ship Berthing using Artificial Neural Network Based on Virtual Window Concept in Wind Condition. *IFAC Proc. Vol.* **2012**, *45*, 286–291. [[CrossRef](#)]
9. Ahmed, Y.A.; Hasegawa, K. Automatic ship berthing using artificial neural network trained by consistent teaching data using nonlinear programming method. *Eng. Appl. Artif. Intell.* **2013**, *26*, 2287–2304. [[CrossRef](#)]
10. Ahmed, Y.A.; Hasegawa, K. Experiment Results for Automatic Ship Berthing using Artificial Neural Network Based Controller. *IFAC Proc. Vol.* **2014**, *47*, 2658–2663. [[CrossRef](#)]
11. Im, N.-K.; Nguyen, V.S. Artificial neural network controller for automatic ship berthing using head-up coordinate system. *Int. J. Nav. Arch. Ocean Eng.* **2018**, *10*, 235–249. [[CrossRef](#)]
12. Nguyen, V.S.; Do, V.-C.; Im, N. Development of Automatic Ship Berthing System Using Artificial Neural Network and Distance Measurement System. *Int. J. Fuzzy Log. Intell. Syst.* **2018**, *18*, 41–49. [[CrossRef](#)]
13. Zhang, Q.; Guibing, Z.; Xin, H.; Renming, Y. Adaptive neural network auto-berthing control of marine ships. *Ocean Eng.* **2019**, *177*, 40–48. [[CrossRef](#)]
14. Shuai, Y.; Li, G.; Cheng, X.; Skulstad, R.; Xu, J.; Liu, H.; Zhang, H. An efficient neural-network based approach to automatic ship docking. *Ocean Eng.* **2019**, *191*, 106514. [[CrossRef](#)]
15. Lee, D.; Lee, S.-J.; Seo, Y.-J. Application of Recent Developments in Deep Learning to ANN-based Automatic Berthing Systems. *Int. J. Eng. Technol. Innov.* **2020**, *10*, 75–90. [[CrossRef](#)]
16. Tran, V.L.; Im, N. A study on ship automatic berthing with assistance of auxiliary devices. *Int. J. Nav. Arch. Ocean Eng.* **2012**, *4*, 199–210. [[CrossRef](#)]
17. Ahmed, Y.A.; Hasegawa, K. Artificial neural network based automatic ship berthing combining PD controlled side thrusters—A combined controller for final approaching to berth. In Proceedings of the 13th International Conference on Control Automation Robotics and Vision, Singapore, 10–12 December 2014.
18. Park, J.-Y.; Kim, N. Design of an adaptive backstepping controller for auto-berthing a cruise ship under wind loads. *Int. J. Nav. Arch. Ocean Eng.* **2014**, *6*, 347–360. [[CrossRef](#)]
19. Nguyen, V.S. Investigation on a Novel Support System for Automatic Ship Berthing in Marine Practice. *J. Mar. Sci. Eng.* **2019**, *7*, 114. [[CrossRef](#)]
20. Nguyen, V.-S.; Im, N. Automatic Ship Berthing Based on Fuzzy Logic. *Int. J. Fuzzy Log. Intell. Syst.* **2019**, *19*, 163–171. [[CrossRef](#)]
21. Bui, V.P.; Kim, Y.B.; Jang, J.S. Design of sliding mode controller for ship berthing. In Proceedings of the 11th International Conference on Control, Automation and Systems, Vienna, Austria, 1–3 September 2014.

22. Mizuno, N.; Kakami, H.; Okazaki, T. Parallel Simulation Based Predictive Control Scheme with Application to Approaching Control for Automatic Berthing. *IFAC Proc. Vol.* **2012**, *45*, 19–24. [[CrossRef](#)]
23. Li, S.; Liu, J.; Negenborn, R.R.; Wu, Q. Automatic Docking for Underactuated Ships Based on Multi-Objective Nonlinear Model Predictive Control. *IEEE Access* **2020**, *8*, 70044–70057. [[CrossRef](#)]
24. Mizuno, N.; Uchida, Y.; Okazaki, T. Quasi Real-Time Optimal Control Scheme for Automatic Berthing. *IFAC PapersOnLine* **2015**, *48*, 305–312. [[CrossRef](#)]
25. Maki, A.; Sakamoto, N.; Akimoto, Y.; Nishikawa, H.; Umeda, N. Application of optimal control theory based on the evolution strategy (CMA-ES) to automatic berthing. *J. Mar. Sci. Technol.* **2019**, *25*, 221–233. [[CrossRef](#)]
26. Liao, Y.; Jia, Z.; Zhang, W.; Jia, Q.; Li, Y. Layered berthing method and experiment of unmanned surface vehicle based on multiple constraints analysis. *Appl. Ocean Res.* **2019**, *86*, 47–60. [[CrossRef](#)]
27. Fossen, T.I. *Handbook of Marine Craft Hydrodynamics and Motion Control*; Wiley: Hoboken, NJ, USA, 2011.
28. Kijima, K. On the maneuvering performance of a ship with the parameter of loading condition. *J. Soc. Nav. Archit. Jpn.* **1990**, *168*, 141–148. [[CrossRef](#)]



© 2020 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).