


Article

Research into the Automatic Berthing of Underactuated Unmanned Ships under Wind Loads Based on Experiment and Numerical Analysis

Zaiji Piao ¹ , Chen Guo ^{1,*} and Shuang Sun ²

¹ School of Marine Electrical Engineering, Dalian Maritime University, Dalian 116026, China

² School of Maritime Economics and Management, Dalian Maritime University, Dalian 116026, China

* Correspondence: dmguoc@126.com; Tel.: +86-0411-84723720

Received: 14 August 2019; Accepted: 27 August 2019; Published: 1 September 2019



Abstract: With the continuous improvement of unmanned ship automation requirements, research into the automatic berthing of underactuated unmanned ships has important theoretical significance and practical value. In order to determine the trajectory of unmanned ships, the line of sight (LOS) algorithm was applied due to the characteristics of underactuated unmanned ships without side thrusters. In order to resist wind disturbance, the active disturbance rejection control (ADRC) method was applied to keep the ship moving on its intended trajectory. Then, to carry out the simulation analysis before the tank experiment, a remote-control simulation system based on a user datagram protocol (UDP) communication was built, and the ability of the ADRC controller to make the ship perform completely automatic berthing in both wind and no wind conditions was verified in simulations. Combined with the simulation results, a tank experiment was accomplished at the Japanese National Research Institute of Fishery Engineering. The experimental results also showed that the ADRC controller has good robustness, that the problems of insufficient autonomous route determination and the disturbance rejection ability in the process of the automatic berthing of underactuated unmanned ships are solved, and the safety of ship navigation is improved, which lays a theoretical and experimental foundation for the further development of unmanned ship control.

Keywords: active disturbance rejection control; automatic berthing; line of sight; tank experiment; unmanned ship; user datagram protocol

1. Introduction

Although the current level of ship automation is relatively high and some small unmanned ship technology has become mature, it is seldom used in practice. The cargo throughput of the world's major ports continues to rise. To improve transport efficiency, unmanned ships are a good choice in terms of safety, economy, and efficiency [1]. The world is competing to develop unmanned ship technology, with the United States and Israel leading in unmanned ship technology research [2–5]. MIT (The Massachusetts Institute of Technology) developed the Artemis unmanned ship in 1993, which can be used to test guidance, control, and propulsion systems. The unmanned ship was used to collect hydrological data from the Charles River in Boston. In 2000, MIT developed an AutoCat unmanned ship capable of remote control or autonomous navigation for survey missions. Israel has also made a tremendous contribution to the development of unmanned ships, especially regarding applications in the military. In 2003, the Rafal Corporation of Israel cooperated with the Israel Aviation Defense Systems Corporation to develop the famous unmanned ship Protector, which can be used for military tasks such as fire cover, antiterrorism, reconnaissance, antisubmarine, and electronic warfare. In 2016, Israel's Elbit Systems Corporation developed the unmanned ship Silver Marlin, which is used for

coastal intelligent patrol missions. Next, Israel developed the unmanned ships Stingray and Starfish, increasing the strength of its modern navy.

Research into the automatic berthing of underactuated ships is theoretically classified as the feedback stabilization control of underactuated systems. Trajectory design is needed in automatic berthing control. Ahmed et al. enabled a ship to navigate along an imaged route and stop at a given place 1.5 times the ship's length from the berth, which means they can complete the automatic berthing control task [6,7]. A value of 1.5 times the ship's length is set to allow sufficient maneuvering space to cope with any adverse situation, so berthing maneuvering does not necessarily bring the ship close to the berth but can also stop the ship at a certain distance outside the berth. There is also a direct approach to the berthing mode. Park and others used the propellers of a twin-propeller twin-rudder ship and the front side thrusters to complete parallel berthing [8]. Tamara et al. in Japan completed the task of first stabilizing the ship out of the berth and then approaching the berth in parallel by using the front and rear side thrusters [9]. For underactuated ships, most of the above research results are difficult to adapt because of the lack of side thrusters. Therefore, the out-of-berth stabilization has more research value in the automatic berthing of underactuated ships. Kim et al. proposed a logic-based control method to drive the ship forward and backward repeatedly along the predetermined course and gradually approach the predetermined state [10]. In theory, this overcomes the limitation of the continuous incentive conditions and allows the ship's speed to be negative, but the drawbacks are obvious. Liu Yang et al. used the nonlinear adaptive control algorithm to design the trajectory tracking controller and berthing controller. The nonholonomic system was transformed into a cascade nonlinear system, and then the adaptive neural network algorithm was used to solve the uncertainty of disturbance. The stability of the closed-loop system was analyzed using the Lyapunov theory [11].

The rest of this paper is organized as follows: Section 2 introduces the automatic berthing mode using the line of sight (LOS) algorithm. The mathematical models of ship motion and wind are also established. Section 3 introduces the design of the controller and the method of parameter tuning. Section 4 carries out the simulation analysis of an unmanned ship and establishes a remote real-time control platform based on user datagram protocol (UDP) communication. Section 5 shows the results of an automatic berthing experiment for unmanned ships at the Japanese National Research Institute of Fishery Engineering.

The main contributions of this article are summarized as follows:

- (1) A mathematical model of underactuated unmanned ships and wind force is established to enhance berthing simulation, and an automatic berthing route determination method based on the LOS algorithm is proposed, with the control states set as the rudder angle and RPM (revolutions per minute), which meets the actual engineering requirements better.
- (2) The parameter tuning methods of a proportional-integral-derivative controller (PID) and active disturbance rejection control (ADRC) in the automatic berthing process are given. The tuning method of PID parameters adopts a closed-loop gain shaping algorithm, and ADRC adopts an innovative method based on the ship maneuverability index.
- (3) A remote real-time control platform based on UDP communication is established. It is a good alternative to the experiment tank system for the Japanese National Research Institute of Fishery Engineering.
- (4) An automatic berthing experiment for underactuated unmanned ships is carried out at the Japanese National Research Institute of Fishery Engineering. It is concluded that the ADRC controller can achieve automatic berthing in windy conditions, and the result is better than that of the PID controller. The experiment proves that we have created a new, realizable, strategy for automatic berthing that improves the autonomous navigation ability of unmanned ships.

2. Problem Formulation

2.1. Automatic Berthing Mode Based on LOS Algorithm

Ship berthing is one of the most complex and difficult ship maneuvering motions. In practice, the berthing of a ship is affected by shallow water, low speed, the shore wall effect, and so on. The berthing often relies on tugboat assistant operation, but it can also be operated by self-reliance when conditions permit. Therefore, research on the automatic berthing control of underactuated ship has important practical significance.

The main difficulties in the design of automatic berthing control for underactuated ship lie in the following aspects: There is no lateral pushing device, but there is lateral drift motion; the maneuverability and rudder efficiency of ships are poor when moving at low speed; the problem of motion trajectory should be taken into account in stabilization control; the inertia of ships is relatively large, and the accuracy of speed and position of ships is demanded at the same time. In this paper, the method to stabilize berthing outside berth proposed by Hasegawa [12], a Japanese scholar, was adopted. The purpose of berthing is that the ship stops at a place, twice the length of the ship from the designated point, and the speed is less than 1 knot. The process of automatic berthing is shown in Figure 1.

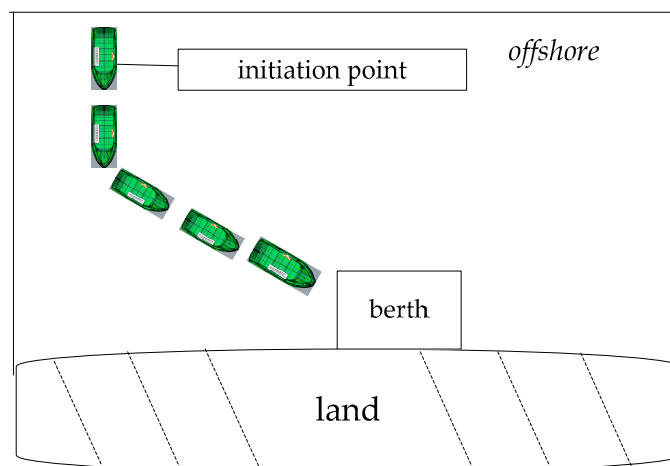


Figure 1. Ship berthing diagram.

The problem of automatic berthing can be simplified as following a fixed trajectory to a specified position after a ship has reached a certain position, which is essentially a ship trajectory tracking problem. The ship trajectory tracking problem belongs to the dynamic positioning problem in a broad sense. The control schematic diagram is shown in Figure 2.

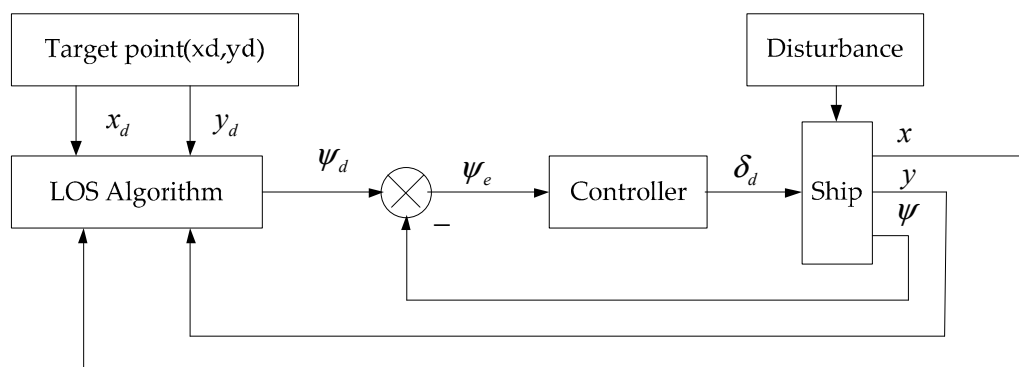


Figure 2. Principle diagram of automatic berthing control for an unmanned ship.

The navigation principle of the LOS algorithm is embodied in its intuitive understanding between rudder steering and ship motion [13–17]. The principle holds that if the course of the ship is kept at the line of sight angle, then after proper control, the desired position will be reached and track tracking will be achieved. Moreover, the LOS algorithm can reduce the traditional control states from three degrees of freedom (ship position (x, y) and course) to two degrees of freedom (course and speed), which is particularly important for the control of underactuated ships.

As shown in Figure 3, it was assumed that the current desired position of track tracking is $P_{k+1} = [x_{k+1}, y_{k+1}]^T$; the last expected position is $P_k = [x_k, y_k]^T$; the current position of the ship is $P = [x, y]^T$. Then, the LOS angle can be calculated by Equation (1) [18]:

$$\psi_{LOS} = \arctan\left(\frac{y_{k+1} - y}{x_{k+1} - x}\right), \quad \psi_{LOS} \in (-\pi, \pi) . \tag{1}$$

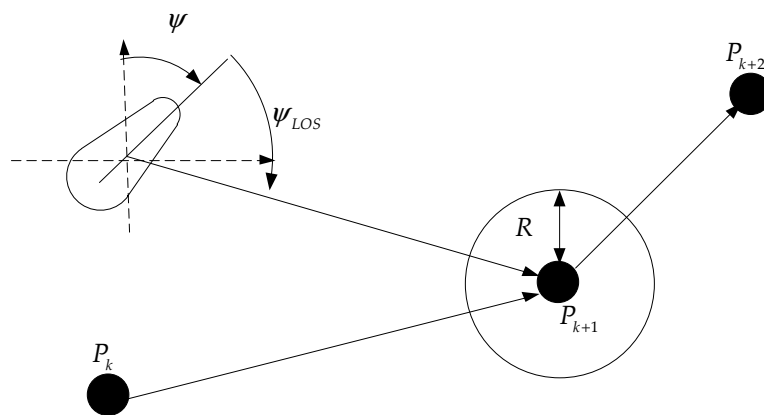


Figure 3. Principle diagram of the line of sight (LOS) navigation algorithm.

When the LOS algorithm is applied to ship trajectory tracking, Figure 4 shows that the expected heading point $P_{LOS} = [x_{LOS}, y_{LOS}]^T$ is $\Delta = nL$ away from the projection point of the current position of the ship on the desired track. L is the length of the ship, and Δ is the visual distance of the ship. At this time, the LOS control algorithm can be written as Equation (2):

$$\alpha_\phi = \alpha_k + \arctan\left(\frac{-e}{\Delta}\right), \tag{2}$$

where e represents the lateral tracking error of the controlled ship, and α_k represents the angle between the northward direction of the geodetic coordinate system and the desired track. This paper assumes that the heading angle deviation $\psi_e = \alpha_\phi - \psi$, and when ψ_e tends to zero, the self-propelled model travels to the target heading point and finally reaches the desired track to complete the trajectory tracking.

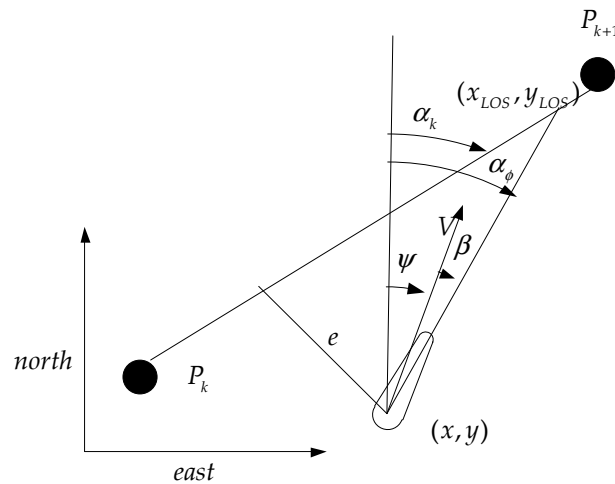


Figure 4. Principle diagram of the LOS trajectory tracking algorithm.

Suppose that there is a circle with the center of the current heading-looking point (x_{k+1}, y_{k+1}) , and the radius is R . At a certain moment, the current position (x_k, y_k) of the controlled ship satisfies the following Equation (3):

$$(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2 \leq R^2. \tag{3}$$

2.2. Mathematical Model of Unmanned Ship

There are two kinds of mathematical models of ship motion, separated and integral. At the end of the 1970s, the Japanese ship maneuvering mathematical model group (MMG) proposed a set of ship motion mathematical models, often referred to as the MMG model. Its main feature is the role of ship hydrodynamic forces and moments according to the physical significance, decomposing hydrodynamic forces and moments in the bare hull, propeller, rudder, and interference with each other. The MMG equation with wind force and the moment of the unmanned ship is Equation (4) [19]:

$$\begin{cases} (m + m_x)\dot{u} - (m + m_y)vr = X_H + X_P + X_R + X_{wind} \\ (m + m_y)\dot{v} + (m + m_x)ur = Y_H + Y_P + Y_R + Y_{wind} \\ (I_{zz} + J_{zz})\dot{r} = N_H + N_P + N_R + N_{wind} \\ \dot{\psi} = r \\ \dot{x}_0 = u \cos \psi - v \sin \psi \\ \dot{y}_0 = v \cos \psi + u \sin \psi \end{cases} \tag{4}$$

In Equation (4), m is the mass of ship hull; m_x and m_y are longitudinal and transverse added mass, respectively; I_{zz} and J_{zz} are the moment of inertia and additional moment of inertia; X , Y , and N represent vertical force, lateral force, and turning moment, respectively. The subscript H is hull viscous fluid force and moment; the subscript for P represents force and moment produced by the propeller; R represents force and moment produced by rudder; the wind represents disturbance of ocean wind; x_0 and y_0 are for the longitudinal and transverse position of the ship; v is the lateral velocity of the ship; u is the longitudinal velocity of the ship; Ψ is the heading angle; and r is the turning angular velocity.

The integral model considers the separation mechanism of the hull as a whole, and the influence between the various mechanisms has been taken into account by default in the test. After the relevant

parameters of the ship are known, the definitions for V, L, x_c are speed, ship length, and the coordinates of the center of gravity. Dimensionless quantities are in Equation (5) [20]:

$$\begin{cases} m' = m / (\frac{1}{2}\rho L^3), m = \rho \nabla \\ x_c' = x_c / L \\ v' = v / V \\ r' = rL / V \\ I_{ZZ}' = I_{ZZ} / (\frac{1}{2}\rho L), I_{ZZ} = \frac{mL^2}{16} \end{cases} \quad (5)$$

In the upper Equation (5), ρ is the water density, and ∇ is the displacement.

The hydrodynamic derivative is the most important part of the mathematical model of ship motion. Clarke collated a large number of ship model experimental data and obtained 10 regression equations of hydrodynamic derivatives by the main parameters of the ship, which can be written as Equation (6) [21]:

$$\begin{cases} Y_{\dot{v}}' = -[1 + 0.16C_b B/T - 5.1(B/L)^2] \cdot \pi(T/L)^2 \\ Y_{\dot{r}}' = -[0.67B/L - 0.0033(B/T)^2] \cdot \pi(T/L)^2 \\ N_{\dot{v}}' = -[1.1B/L - 0.041B/T] \cdot \pi(T/L)^2 \\ N_{\dot{r}}' = -[1/12 + 0.017C_b B/T - 0.33B/L] \cdot \pi(T/L)^2 \\ Y_v' = -[1 + 0.40C_b B/T] \cdot \pi(T/L)^2 \\ Y_r' = -[-1/2 + 2.2B/L - 0.080B/T] \cdot \pi(T/L)^2 \\ N_v' = -[1/2 + 2.4T/L] \cdot \pi(T/L)^2 \\ N_r' = -[1/4 + 0.039B/T - 0.56B/L] \cdot \pi(T/L)^2 \\ Y_{\delta}' = 3.0A_{\delta} / L^2 \\ N_{\delta}' = -(1/2)Y_{\delta}' \end{cases} \quad (6)$$

In the upper Equation (6), B, T, C_b , and A_{δ} are the breadth of the ship, draft, square coefficient, and rudder angle; Y_v', Y_r', N_v' , and N_r' are the hydrodynamic derivatives of the hull itself. According to practical experience, it is necessary to revise them according to Equation (7) [20]:

$$\begin{cases} \Delta Y'v = -\gamma Y_{\delta}' \\ \Delta Y'r = -1/2(\Delta Y'v) \\ \Delta N'v = -1/2(\Delta Y'v) \\ \Delta N'r = 1/4(\Delta Y'v) \\ t\gamma = 0.30 \end{cases} \quad (7)$$

Thus, $Y'v = Y_v' + \Delta Y'v, Y'r = Y_r' + \Delta Y'r, N'v = N_v' + \Delta N'v, N'r = N_r' + \Delta N'r$ allow the ship to be modeled as a standard, dimensionless state space equation, Equation (8) [22]:

$$\begin{bmatrix} m' - Y_{\dot{v}}' & L(m'x'_c - Y_{\dot{r}}') \\ m'x'_c - N_{\dot{v}}' & L(I'_{zz} - N_{\dot{r}}') \end{bmatrix} \begin{bmatrix} \dot{v} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{V}{L} Y'_{\dot{v}} & V(Y'_r - m') \\ \frac{V}{L} N'_{\dot{v}} & V(N'_r - m'x'_c) \end{bmatrix} \begin{bmatrix} v \\ r \end{bmatrix} + \begin{bmatrix} \frac{V^2}{L} Y'_{\delta} \\ \frac{V^2}{L} Y'_{\delta} \end{bmatrix} \delta \quad (8)$$

Equation (8) can be simplified to Equation (9):

$$\dot{X}_{(2)} = A_{(2)}X_{(2)} + B_{(2)}\delta, A_{(2)} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, B_{(2)} = \begin{bmatrix} b_{11} \\ b_{21} \end{bmatrix} \quad (9)$$

$$\begin{cases} a_{11} = [(I'_z - N'_r)Y'_v - (m'x'_c - Y'_r)N'_v]V/S_1 \\ a_{12} = [(I'_z - N'_r)(Y'_r - m') - (m'x'_c - Y'_r)(N'_r - m'x'_c)]LV/S_1 \\ a_{21} = [-(m'x'_c - N'_v)Y'_v + (m' - Y'_v)N'_v]V/L/S_1 \\ a_{22} = [-(m'x'_c - N'_v)(Y'_r - m') + (m' - Y'_v)(N'_r - m'x'_c)]V/S_1 \\ b_{11} = [(I'_z - N'_r)Y'_\delta - (m'x'_c - Y'_r)N'_\delta]V^2/S_1 \\ b_{21} = [-(m'x'_c - N'_v)Y'_\delta + (m' - Y'_v)N'_\delta]V^2/L/S_1 \\ S_1 = [(I'_z - N'_r)(m' - Y'_v) - (m'x'_c - N'_v)(m'x'_c - Y'_r)]L \end{cases} \quad (10)$$

The integral model is:

$$G_{\psi\delta}(s) = \frac{K}{s(Ts + 1)}. \quad (11)$$

Among it, $K = \frac{b_{11}a_{21} - b_{21}a_{11}}{a_{11}a_{22} - a_{12}a_{21}}$, $T = -(\frac{a_{11} + a_{22}}{a_{11}a_{22} - a_{12}a_{21}} + \frac{b_{21}}{b_{11}a_{21} - b_{21}a_{11}})$.

The wind force and moment can be calculated by Equation (12) [23]:

$$\begin{cases} X_{wind} = 0.5\rho_a U_R^2 A_f C_{wx} \\ Y_{wind} = 0.5\rho_a U_R^2 A_s C_{wy} \\ N_{wind} = 0.5\rho_a U_R^2 A_s L C_{wn} \end{cases} \quad (12)$$

In Equation (12), ρ_a is the air density; U_R is the relative wind velocity; A_f is the orthogonal projection area of the hull surface; A_s is the side of the projection area on the water line; and C_{wx} , C_{wy} , and C_{wn} are the lateral and longitudinal direction of the wind pressure force coefficient and wind pressure moment coefficient, whose calculations are in Equation (13):

$$\begin{cases} C_{wx} = X_0 + X_1 \cos \psi_a + X_3 \cos 3\psi_a + X_5 \cos 5\psi_a \\ C_{wy} = Y_1 \sin \psi_a + Y_3 \sin 3\psi_a + Y_5 \sin 5\psi_a \\ C_{wn} = N_1 \sin \psi_a + N_2 \sin 2\psi_a + N_3 \sin 3\psi_a \end{cases} \quad (13)$$

In Equation (13), ψ_a is the attack angle. The serial numbers of X_i , Y_i , and N_i range from 1 to 5. Fujiwara obtained the parameter threshold for the calculation result after several experiments. The parameter exceeding the threshold is: X_0 , X_1 , X_3 , X_5 , Y_1 , Y_3 , Y_5 , N_1 , N_2 , and N_3 , calculated as Equations (14)–(16). L is the length of the ship; B is the breadth of the ship; A_T is the transverse projected area of the ship; A_L is the lateral projected area of the ship; A_{OD} is the lateral projected area of the superstructure; C is the distance from midship to the center of the lateral projected area; C_{BR} is the distance from midship to the center of the lateral projected area of superstructure; H_{BR} is the height to the top of the superstructure; H_C is the height to the lateral projected area; x_{nm} , y_{nm} , and n_{nm} are a recursive coefficient obtained by the experiment and can be determined based on Table 1.

$$\begin{cases} X_0 = x_{00} + x_{01}(BH_{BR})/A_T + x_{02}C/H_C + x_{03}A_{OD}/(L^2) \\ X_1 = x_{10} + x_{11}A_L/(LB) + x_{12}LH_C/A_L + x_{13}LH_{BR}/A_L \\ \quad + x_{14}A_{OD}/A_L + x_{15}A_T/(LB) + x_{16}L^2/A_T + x_{17}L/H_C \\ X_3 = x_{30} + x_{31}L/(LH_{BR}) + x_{32}A_L/A_T + x_{33}LH_C/A_L \\ \quad + x_{34}A_{OD}/A_L + x_{35}A_{OD}/(L^2) + x_{36}C/H_C + x_{37}C_{BR}/L \\ X_5 = x_{50} + x_{51}A_L/A_{OD} + x_{52}C_{BR}/L + x_{53}A_L/(LB) \end{cases} \quad (14)$$

$$\begin{cases} Y_1 = y_{10} + y_{11}C_{BR}/L + y_{12}C/L + y_{13}A_L/A_{OD} + y_{14}C/H_C + y_{15}A_T/(BH_{BR}) \\ Y_3 = y_{30} + y_{31}A_L/(LB) + y_{32}LH_C/A_L + y_{33}C_{BR}/L + y_{34}B/H_{BR} + y_{35}A_{OD}/A_L + y_{36}A_T/(BH_{BR}) \\ Y_5 = y_{50} + y_{51}A_L/(LB) + y_{52}L/H_{BR} + y_{53}C_{BR}/L + y_{54}B^2/A_T + y_{55}C/L + y_{56}LH_C/A_L \end{cases} \quad (15)$$

$$\begin{cases} N_1 = n_{10} + n_{11}C/L + n_{12}LH_C/A_L + n_{13}A_T/A_L + n_{14}C/H_C \\ \quad + n_{15}A_L/(LB) + n_{16}A_T/(L^2) + n_{17}B^2/A_T + n_{18}C_{BR}/L \\ N_2 = n_{20} + n_{21}C_{BR}/L + n_{22}C/L + n_{23}A_L/A_{OD} + n_{24}A_T/(B^2) \\ \quad + n_{25}L/H_{BR} + n_{26}A_T/(BH_{BR}) + n_{27}A_L/(LB) + n_{28}A_L/(L^2) \\ N_3 = n_{30} + n_{31} * C_{BR}/L + n_{32} * A_T/(B * H_{BR}) + n_{33} * A_L/A_T \end{cases} \quad (16)$$

Table 1. Each coefficient of independent variables.

m=	0	1	2	3	4	5	6	7	8
x_{0m}	-0.330	0.293	0.0193	0.682					
x_{1m}	-1.353	1.700	2.87	-0.463	-0.570	-6.640	-0.0123	0.0202	
x_{3m}	0.830	-0.413	-0.0827	-0.563	0.804	-5.67	0.0401	-0.132	
x_{5m}	0.0372	-0.0075	-0.103	0.0921					
y_{1m}	0.684	0.717	-3.22	0.0281	0.0661	0.298			
y_{3m}	-0.400	0.282	0.307	0.0519	0.0526	-0.0814	0.0582		
y_{5m}	0.122	-0.166	-0.0054	-0.0481	-0.0136	-0.0864	-0.0297		
n_{1m}	0.299	1.71	0.183	-1.09	-0.0442	-0.289	4.24	-0.0646	0.0306
n_{2m}	0.117	0.123	-0.323	0.0041	-0.166	-0.0109	0.174	0.214	-1.06
n_{3m}	0.0230	0.0385	-0.0339	0.0023					

3. Controller Design and Parameter Tuning

In the experiment, the PID control algorithm and the ADRC algorithm were used. The two algorithms and their parameters' respective tuning methods are introduced.

3.1. PID Algorithm

According to the closed-loop spectrum requirement of the system, this paper presents a method of PID parameter tuning based on closed-loop gain shaping. The solution process is simple, and the physical meaning is obvious. The designed PID controller has good robustness. The second-order object is Equation (17):

$$G = \frac{b_1s + b_0}{a_2s^2 + a_1s + a_0}. \quad (17)$$

A practical engineering object can be transformed into a second-order object by model reduction or Bode graph approximation. According to the first-order closed-loop gain shaping algorithm, control K_C can be calculated as Equation (18) [24]:

$$K_C = \frac{a_2s^2 + a_1s + a_0}{b_0T_{01}s} = \frac{a_1}{b_0T_{01}} + \frac{a_0}{b_0T_{01}s} + \frac{a_2s}{b_0T_{01}} \quad (18)$$

In Equation (18), T_{01} is time constant designed for the time based on the bandwidth frequency of the controlled object. It can be seen that Equation (18) is a standard PID controller, k_p , k_i , and k_d are in Equation (19):

$$k_p = \frac{a_1}{b_0T_{01}}, k_i = \frac{a_0}{b_0T_{01}}, k_d = \frac{a_2}{b_0T_{01}}. \quad (19)$$

3.2. ADRC Algorithm

The active disturbance rejection control (ADRC) is not very strict with the mathematical model of the controlled object during disturbances, and it can, through the special structure of nonlinear feedback, compensate for the external and internal error of the system in time [25]. ADRC is mainly composed of a tracking differentiator (TD), extended state observer (ESO), and nonlinear state error feedback control law (NLSEF); the structure is shown in Figure 5.

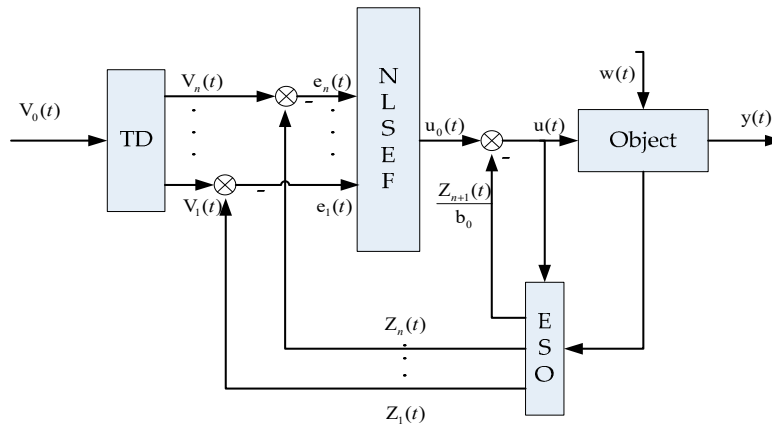


Figure 5. Structural block diagram of the active disturbance rejection control (ADRC) algorithm.

Based on the second-order system as an example, this paper introduces the principle of ADRC, where the second-order controlled system state equation is Equation (20):

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f(x_1, x_2) + bu \\ y = x_1 \end{cases} \quad (20)$$

In the second-order tracking differentiator, for example, differential Equation (21) is as follows:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -fst(x_1, x_2, x, r, h_0) \end{cases} \quad (21)$$

Among them, v is the target signal; v_1 and v_2 are estimates of the target signal and its differential signal; h_0 is the filtering factor; r is the tracking speed factor; the expressions of nonlinear function $fst(v_1, v_2, v, r, h_0)$ are in Equation (22) [26]:

$$\begin{cases} d = r \cdot h; d_0 = d \cdot h; y = x_1 - u + h \cdot x_2; a_0 = \sqrt{d^2 + 8r|y|} \\ fst = \begin{cases} \frac{-r \cdot a}{d} & |a| \leq d \\ -r \cdot \text{sgn}(a) & |a| > d \end{cases} \quad a = \begin{cases} x_2 + \frac{y}{h} & |y| \leq d_0 \\ x_2 + 0.5(a_0 - d) \cdot \text{sgn}(y) & |y| > d_0 \end{cases} \end{cases} \quad (22)$$

For the second-order system, expand the real-time action of the open-loop system acceleration $f(x_1(t), x_2(t))$ for the new state variables, and $\dot{x}_3 = \varphi(t)$ so that Equation (20) turns to Equation (23):

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f(x_1, x_2) + bu \\ \dot{x}_3 = \varphi(t) \\ y = x_1 \end{cases} \quad (23)$$

$e = z_1 - y$, the nonlinear continuous observer of the system, is Equation (24):

$$\begin{cases} \dot{z}_1 = z_2 - \beta_{01}e \\ \dot{z}_2 = z_3 - \beta_{02}fal(e, a_{01}, \delta_{01}) + bu \\ \dot{z}_3 = -\beta_{03}fal(e, a_{02}, \delta_{02}) \end{cases} \quad (24)$$

Among them, z_1 and z_2 , respectively, represent the output of the object and its first-order differential; z_3 is for internal and external disturbance estimation; β_{01} , β_{02} , and β_{03} are for the output error correction gain; the *fal* function is Equation (25) [26]:

$$fal(e, a, \delta) = \begin{cases} |e|^a \operatorname{sgn}(e), & |e| > \delta \\ e/\delta^{1-a}, & |e| \leq \delta \end{cases} \quad \delta > 0. \tag{25}$$

Set $e_1 = x_1 - z_1; e_2 = x_2 - z_2$, and the nonlinear state error feedback control law is Equation (26):

$$u_0 = k_1 fal(e_1, a_1, \delta_1) + k_2 fal(e_2, a_2, \delta_2). \tag{26}$$

$\beta_{01}, \beta_{02}, a_1, a_2, \delta_1$, and δ_2 are all adjustable parameters, while e_1 and e_2 represent the state error.

The basis calculation method of the ADRC parameters can be carried out as follows:

(1) For TD, r determines the tracking speed of the tracking signal. The smaller the tracking speed, the better it can suppress overshoot. If r is too small, it will affect the response speed of the system. For h , when the simulation step size is determined, the steady-state flutter can be eliminated as long as h is larger than the simulation step.

(2) ESO stability is the main objective of parameter adjustment, generally taking $a_{01} = 0.5, a_{02} = 0.25, \beta_{01}h = 1, \beta_{02} = \frac{\beta_{01}}{h}$, and $\beta_{03} = \frac{\beta_{02}}{h}$.

(3) The two parameters of NLSEF have relatively clear meanings. Compared with PID, k_1 is equivalent to the proportional coefficient, k_2 almost equivalent to the differential coefficient, and its function is similar to that of PID.

(4) The function of parameter b is to produce a large error control signal when the system has a delay, and the corresponding system response becomes slower. It is similar to the integral coefficient. A system with a b value that is too small may be unstable, and a b value that is too large will slow down the response of the system.

In the specific application process, the discussed methods need to be debugged many times, because the control parameters of ADRC do not depend on the mathematical model, and the actual nonlinear factors on the effect of the controller need to be avoided. Next, a more general method for calculating parameters is proposed. In summary, the controller equations of ADRC are in Equation (27):

$$\begin{cases} e = z_1 - y \\ \dot{z}_1 = z_2 - \beta_{01}e \\ \dot{z}_2 = z_3 - \beta_{02}fal(e, a_{01}, \delta_{01}) + bu \\ \dot{z}_3 = -\beta_{03}fal(e, a_{02}, \delta_{02}) \\ e_1 = x_1 - z_1; e_2 = x_2 - z_2 \\ u_0 = k_1 fal(e_1, a_1, \delta_1) + k_2 fal(e_2, a_2, \delta_2) \\ u = u_0 - z_3/b \end{cases} . \tag{27}$$

The parameters that need to be tuned are the following: ESO: $b, \beta_{01}, \beta_{02}, \beta_{03}$; NLSEF: k_1, k_2 . The rest of the parameters can be taken as follows: TD: $r = 100; h = 2$, ESO: $a_{01} = 0.5, a_{02} = 0.25, \delta_{01} = \delta_{02} = 0.1$; NLSEF: $a_1 = 0.75, a_2 = 2.75, \delta_1 = \delta_2 = 0.1$.

According to gain K of the ship integral model, it can be assumed that: $p = 1/\sqrt{K}$; next, a second-order controlled system is taken as an example to study the ADRC parameter tuning method based on manipulability index p .

There is a standard system Equation (28):

$$S1 : \begin{cases} \dot{x}_{11} = x_{12} \\ \dot{x}_{12} = f_1(x_{11}, x_{12}, t_1) + b_1^* u_1 \\ y_1 = x_{11} \end{cases} \quad ADRC_1 : \begin{cases} e_{01} = z_{11} - y_1 \\ \dot{z}_{11} = z_{12} - \beta_{11} e_{01} \\ \dot{z}_{12} = z_{13} - \beta_{12} fal(e_{01}, a_{01}, \delta_{01}) + b_1 u_1 \\ \dot{z}_{13} = -\beta_{13} fal(e_{01}, a_{02}, \delta_{02}) \\ e_{11} = x_{11} - z_{11}; e_{12} = x_{12} - z_{12} \\ u_{01} = k_{11} fal(e_{11}, a_1, \delta_1) \\ \quad + k_{12} fal(e_{12}, a_2, \delta_2) \\ u_1 = u_{01} - z_{13} / b_1 \end{cases} \quad (28)$$

The manipulability index of S1 is p_1 , and the parameters of the controller have been set. Now the following system parameters need to be tuned in Equation (29):

$$S2 : \begin{cases} \dot{x}_{21} = x_{22} \\ \dot{x}_{22} = f_2(x_{21}, x_{22}, t_2) + b_2^* u_2 \\ y_2 = x_{21} \end{cases} \quad ADRC_2 : \begin{cases} e_{02} = z_{21} - y_2 \\ \dot{z}_{21} = z_{22} - \beta_{21} e_{02} \\ \dot{z}_{22} = z_{23} - \beta_{22} fal(e_{02}, a_{01}, \delta_{01}) + b_2 u_2 \\ \dot{z}_{23} = -\beta_{23} fal(e_{02}, a_{02}, \delta_{02}) \\ e_{21} = x_{21} - z_{21}; e_{22} = x_{22} - z_{22} \\ u_{02} = k_{21} fal(e_{21}, a_1, \delta_1) \\ \quad + k_{22} fal(e_{22}, a_2, \delta_2) \\ u_2 = u_{02} - z_{23} / b_2 \end{cases} \quad (29)$$

The manipulability index of S2 is p_2 , $m = \frac{p_1}{p_2}$, according to the concept of time scale proposed by Professor Han, and Equation (27) is changed to Equation (30):

$$\begin{cases} e_{01} = z_{11} - y_1 \\ \dot{z}_{11} = z_{12} - \beta_{21} e_{01} / m \\ \dot{z}_{12} = z_{13} - \beta_{22} fal(e_{01}, a_{01}, \delta_{01}) / m^2 + b_2 u_2 / m^2 \\ \dot{z}_{13} = -\beta_{23} fal(e_{01}, a_{02}, \delta_{02}) / m^3 \\ e_{21} = x_{11} - z_{11}; e_{12} = x_{12} - z_{12} \\ u_{02} = k_{21} fal(e_{11}, a_1, \delta_1) + k_{22} m fal(e_{12}, a_2, \delta_2) \\ u_2 = u_{02} - z_{13} / (b_2 / m^2) \end{cases} \quad (30)$$

Then the parameter setting equations are obtained in Equation (31):

$$\beta_{21} = m\beta_{11}, \beta_{22} = m^2\beta_{12}, \beta_{23} = m^3\beta_{13}, k_{21} = k_{11}, k_{22} = 1 / mk_{12}, b_2 = m^2 b_1. \quad (31)$$

4. Automatic Berthing Simulation

The tank experiment was carried out at the Japan National Research Institute of Fisheries Engineering, so the simulation is aimed at this experiment. The experiment mode is shown in Figure 6. The client and the server communicate with each other through UDP, and the server communicates with the ship through Wi-Fi. Because the server is written in python, in the simulation, the server was also written in Python to publish and receive data, and the client was written in MATLAB/Simulink to achieve ship motion control.

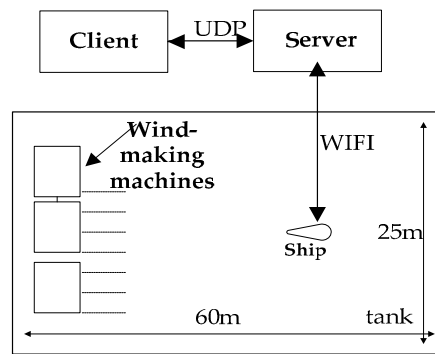


Figure 6. Tank experiment model.

An unmanned ship has a unique communication system. Because there is no personnel on board, it is very important to exchange information with the shore. Common communication protocols include the transmission control protocol/internet protocol (TCP/IP), UDP, serial communication protocol, wireless network protocol, etc. [27–29]. TCP is a reliable communication protocol, that is, before formal communication between the two sides, it is necessary to establish a connection to ensure that data are not lost. Therefore, the transmission speed of data is relatively slow, and UDP only needs to set a port and IP address to send data, which does not guarantee the reliability of data (i.e., the possibility of packet loss), so it is an unreliable communication protocol, but UDP has a transmission speed that TCP cannot match.

As a general control system simulation software, MATLAB has been widely used for its powerful functions. In the research of control theory, the design and simulation of control law can be conveniently carried out using MATLAB/Simulink. In simulation tests, it is often necessary to connect external hardware simulation equipment for real-time simulation research. However, in the past, MATLAB/Simulink lacked the ability of real-time simulation. Usually, we need to write another application program that can simulate real time to do this work. In this way, the function of real-time simulation software is relatively singular, involving substantial repetitive work, and debugging is difficult. In addition, the communication between MATLAB and other platforms has not been studied much, which is not conducive to the direct application of MATLAB for actual control. Therefore, it is necessary to explore how to carry out real-time control in MATLAB/Simulink. Using the interface of MATLAB/Simulink and an external program to exchange data with the external control program through shared memory, real-time communication across platforms, and real-time simulation of ship motion control with high accuracy can be realized.

A 7.2-scale fishing ship named Kosoko was used in the experiment. The dimensions of the ship are shown in Table 2.

Table 2. Ship dimensions.

Description	Ship Scale	Model Scale
Length	12.00 m	1.667 m
Breadth	5.10 m	0.701 m
Draft	1.00 m	0.138 m

Firstly, a ship maneuverability simulation was carried out to verify the accuracy of the ship mathematical model. The turning experiment and zig-zag experiment are the two most important experiments to test ship maneuverability [30,31]. The simulation results of the +35 deg turning test are shown in Figure 7.

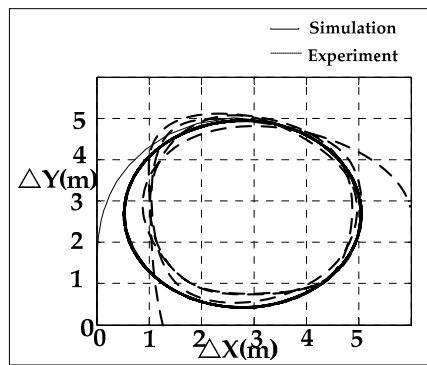


Figure 7. +35 deg turning test: comparisons between simulated and experimental data.

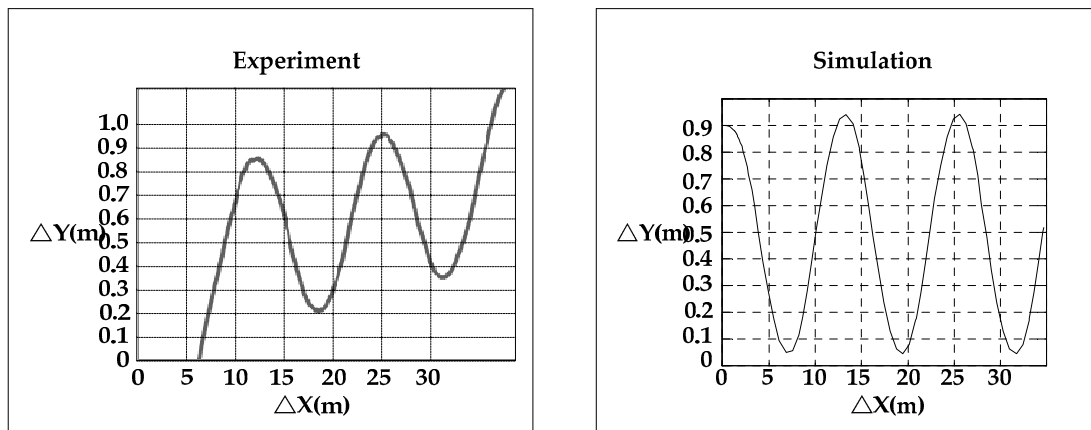
In Figure 7, ΔX and ΔY is the change of position. Set the current position to (x,y) and the initial position to (x_0,y_0) , so $\Delta X = x - x_0$, $\Delta Y = y - y_0$; the simulation data are analyzed in Table 3.

Table 3. Compare for results of the +35 deg turning test.

+35 deg Turning Test	Experiment	Simulation	$error = \left \frac{d_{\text{experiment}} - d_{\text{simulation}}}{d_{\text{experiment}}} \right \times 100\%$
Advance (m)	4.32	4.51	4.39%
Transfer (m)	1.13	1.24	9.73%
Tactical diameter (m)	3.91	4.23	8.18%
Final diameter (m)	4.24	4.53	6.84%

From the results of Table 3, the errors of the ship’s advance, transfer, tactical diameter, and final diameter with real ship data are very small. The simulation results can basically meet the requirements of engineering accuracy and lay a foundation for automatic berthing simulation.

Figures 8 and 9 are the simulation results of the +10 deg zig-zag test.



(a) Experimental data

(b) Simulation data

Figure 8. +10 deg zig-zag test: comparisons between simulated and experimental data.

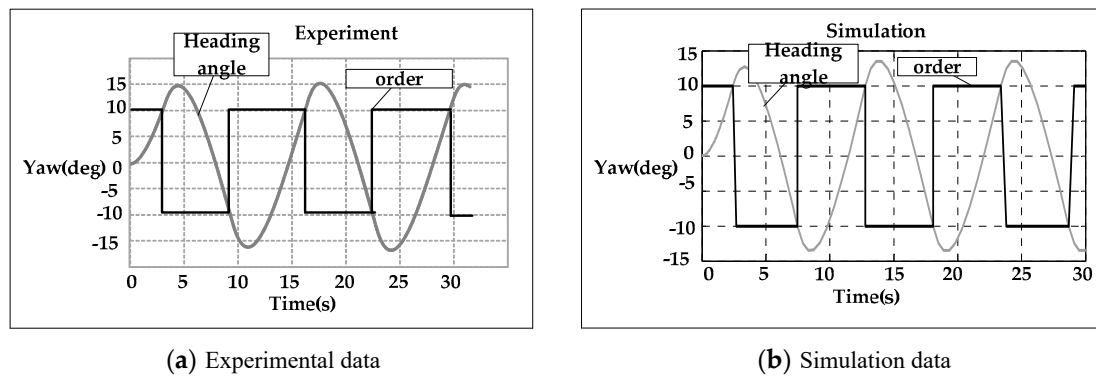


Figure 9. Yaw of +10 deg zig-zag test: comparisons between simulated and experimental data.

From Figure 8, it can be seen that there are some differences in ship trajectories due to the neglect of some nonlinear factors in the simulation process. There is drift in the actual experiment, but the motion characteristics and steering speed are very close. The analysis of the overshoot angle is shown in Table 4.

Table 4. Analysis of the overshoot angle of the +10 deg zig-zag test.

Overshoot Angle (deg)	Experiment	Simulation	$error = \left \frac{d_{\text{experiment}} - d_{\text{simulation}}}{d_{\text{experiment}}} \right \times 100\%$
First time	4.81	3.95	17.88%
Second time	4.93	4.26	13.59%
Third time	4.84	4.18	13.64%

Through the zig-zag experiment, it can be seen that the maneuverability and steering performance of the ship model are similar to those of the actual ship. The first and second overshoot angles are very close. The international maritime organization (IMO) standard for maneuverability of the zig-zag experiment is that the first and second overshoot angles (δ_1, δ_2) shall satisfy the following conditions (32):

$$\delta_1 = \begin{cases} < 10^\circ & L/V < 10s \\ < 20^\circ & L/V \geq 30s \\ < \left(5 + \frac{1}{2}(L/V)\right)^\circ & 10s \leq L/V < 30s \end{cases} \quad \delta_2 = \begin{cases} < 25^\circ & L/V < 10s \\ < 40^\circ & L/V \geq 30s \\ < \left(17.5 + \frac{3}{4}(L/V)\right)^\circ & 10s \leq L/V < 30s \end{cases} \quad (32)$$

In (32), L is the length of the ship and V is the velocity. In this test, $L/V < 10s$, so both of them meet the IMO standard. In summary, the ship model can be used in simulation calculation and has a strong substitution.

In the process of ship motion simulation, two computers were used, one functioning as a server to receive and publish ship data, and the other serving as a ship motion controller. In the actual communication process, the communication frequency of both sides is a very important factor. If the communication frequency is not appropriate, it will cause packet loss and control effect weakening. The communication frequency was tested as follows: The server issued data to the network in a period of 0.05 s, and the client was tested with 0.025 s, 0.05 s, 0.1 s, and 0.125 s, respectively. The numbers of data received are shown in Figure 10:

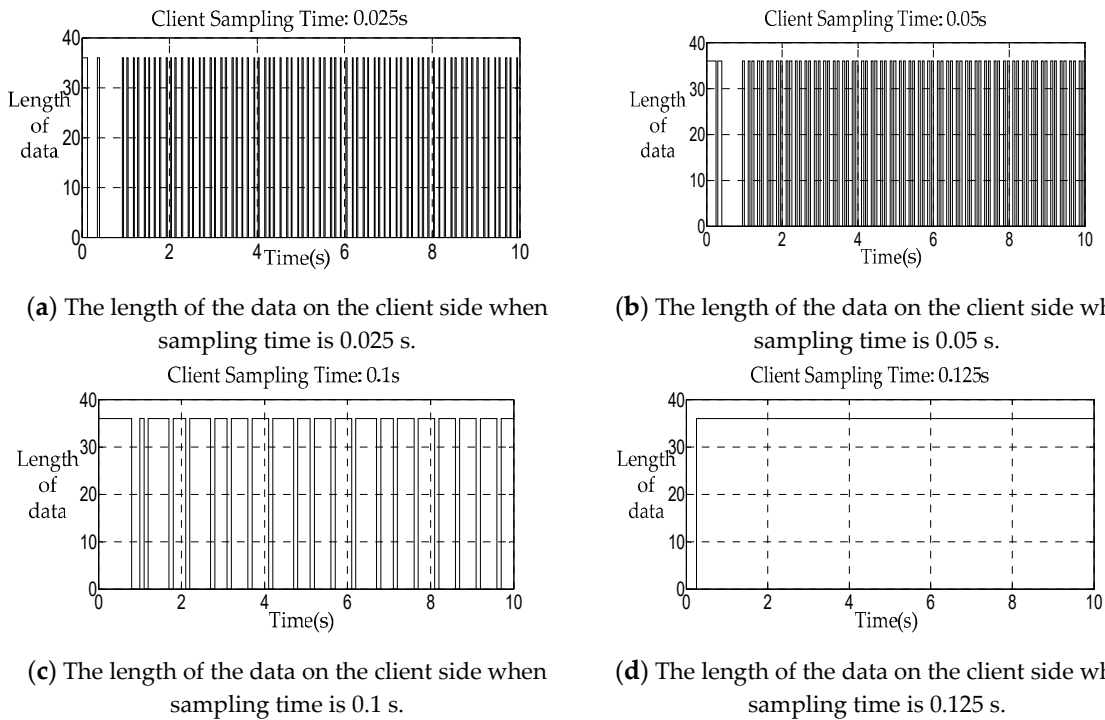


Figure 10. Communication frequency test before simulation.

As shown in Figure 10, the larger the client sampling time, the smaller the data loss. When at 0.125 s, there is no loss of data, so this sampling time is used in the simulation. In addition, for the sake of insurance, a strategy is adopted in the test, that is, if there is a packet loss, the data received at the previous moment will be maintained, so as to ensure the real-time status and effectiveness of the control.

In the next step, the PID controller and ADRC controller were applied to simulate the automatic berthing of an unmanned ship. The speed control adopts the step-by-step descending mode based on actual navigation experience, and the speed changes to 0 when the specified berth arrives. Parameters of PID: 0.84; 0.001; 1.56. Parameters of ADRC: $r = 100$; $h = 0.1$; $a_{01} = 0.5$; $a_{02} = 0.25$; $\delta_1 = 0.1$; $\beta_{01} = 10$; $\beta_{02} = 100$; $\beta_{03} = 500$; $b_0 = 1$; $a_1 = 0.75$; $a_2 = 2.75$; $\delta_2 = 0.1$; $k_1 = 0.35$; $k_2 = 0.01$; $\delta_{01} = 0.1$; $\delta_{02} = 0.1$. The wind speed is 1.5 m/s (model scale), representing actual five-level wind.

The results are shown in Figures 11–14. Figure 11 shows ship trajectories in automatic berthing simulation when berthing without wind using PID and ADRC and under wind using PID and ADRC; Figure 12 shows ship heading angle; Figure 13 gives the ship rudder angle; the ship RPMs are in Figure 14.

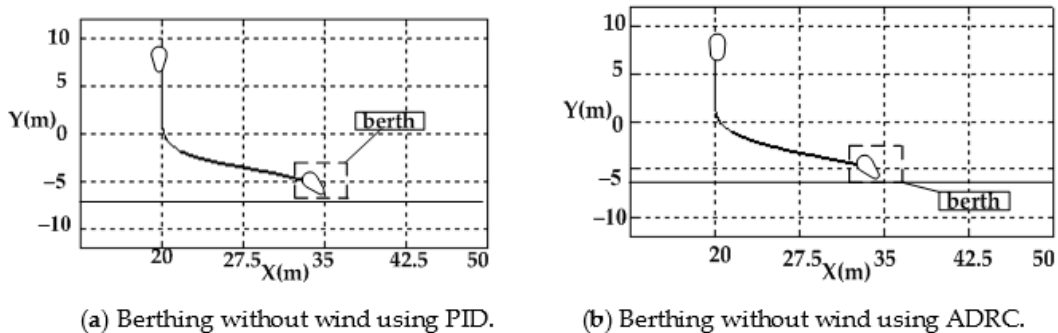
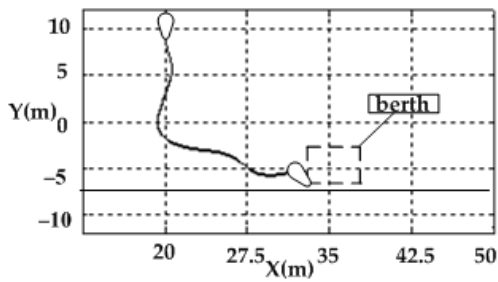
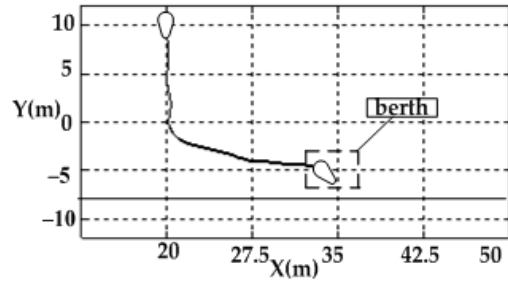


Figure 11. Cont.

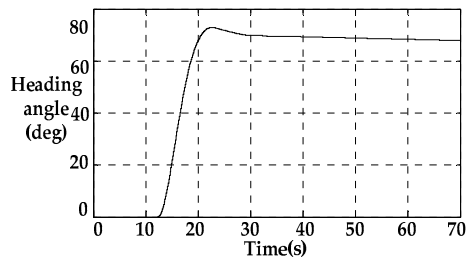


(c) Berthing under wind using PID.

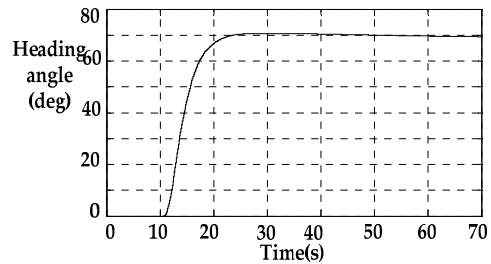


(d) Berthing under wind using ADRC.

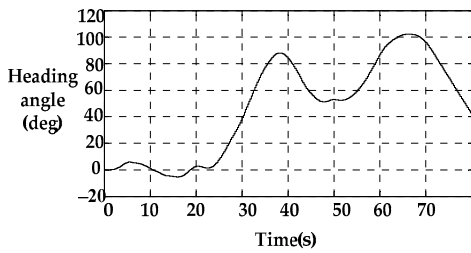
Figure 11. Ship trajectory in the simulation of ship automatic berthing.



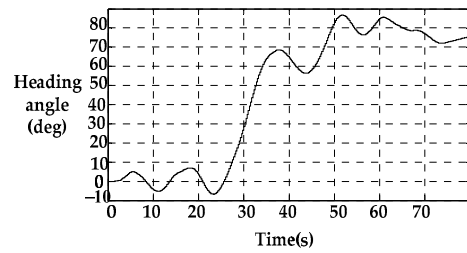
(a) Berthing without wind using PID.



(b) Berthing without wind using ADRC.

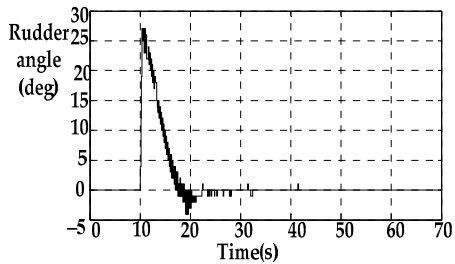


(c) Berthing under wind using PID.

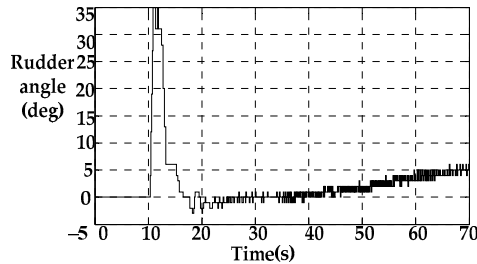


(d) Berthing under wind using ADRC.

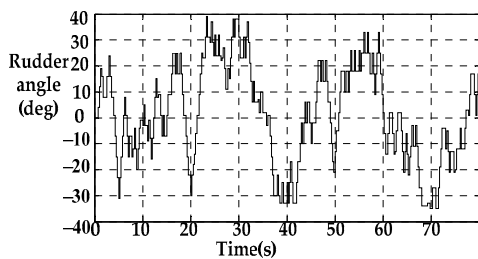
Figure 12. Ship heading angle in the simulation of ship automatic berthing.



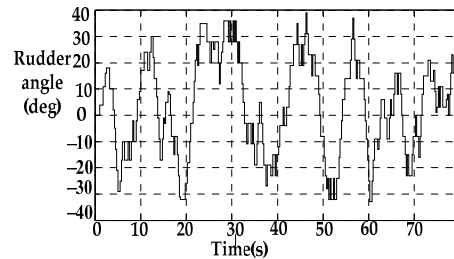
(a) Berthing without wind using PID.



(b) Berthing without wind using ADRC.



(c) Berthing under wind using PID.



(d) Berthing under wind using ADRC.

Figure 13. Ship rudder angle in the simulation of ship automatic berthing.

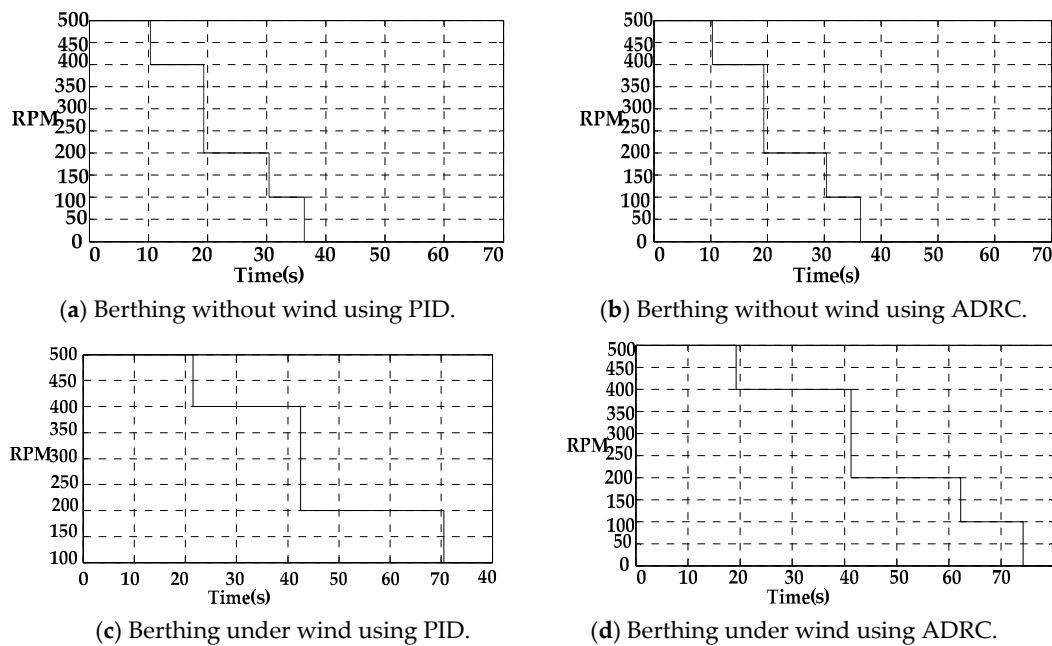


Figure 14. Ship revolutions per minute (RPMs) in the simulation of ship automatic berthing.

In Figure 11, it can be seen that in windless conditions, the PID and ADRC controllers can make the ship reach berth, and the control effect is excellent. However, in the case of wind with a speed of 1.5 m/s (model scale), the ADRC controller makes the ship trajectory more stable, and the ship arrives at berth smoothly. The PID controller has a poor control effect, which makes the ship only reach the edge of the berth. In Figure 12; Figure 13, it can be seen that ADRC has a more sensitive steering ability than PID. In Figure 14, we can see that the speed can be adjusted according to the distance, and the ship can be parked in the designated berth. Detailed analysis is in Table 5, where (a) is for berthing without wind using PID, (b) is for berthing without wind using ADRC, (c) is for berthing under wind using PID, (d) is for berthing under wind using ADRC. D is the distance between the final position and the center of the berth, while Y/N is whether to complete automatic berthing.

Table 5. Analysis of the results of simulation of ship automatic berthing.

	Heading Adjustment Time	RPM Zero Time	D	Y/N
(a)	29.5 s	36.2 s	0.4 m	Y
(b)	22.3 s	35.8 s	0.5 m	Y
(c)	55.4 s	71.4 s	1.7 m	N
(d)	52.3 s	73.5 s	0.6 m	Y

As can be seen from Table 5, the heading adjustment time is the time when the course reaches relative stability, and that of ADRC is faster than that of PID in each case. It should be considered that ADRC can make the ship react more quickly. RPM Zero Time is the time when the ship arrives at the berth. In the case of wind, the speed of the ship is slowed down and the adjustment time is longer, and the PID controller makes the ship reach the edge of the berth. Although the speed can be 0 at this time in the simulation strategy, this situation is considered to be the failure of automatic berthing in practice. Whether with an ADRC or PID controller, the strategy of frequent steering is used to resist the influence of wind on ship motion in the case of wind, and the effect of ADRC is better.

5. Automatic Berthing Experiment

The tank experiment was carried out at the Japan National Research Institute of Fisheries Engineering, and the plane scale coordinate schematic diagram of the experiment is shown in Figure 6.

The experimental system of a self-propelled ship model includes a scale ship model, self-propelled mechanism, communication system, camera, positioning and orientation system, speed sensor, etc. In this experimental system, the ship position was obtained by the ultrasonic positioning system, and the heading angle was measured by the fiber optic gyroscope on the ship, and the speed of the ship was measured by the anemometer at the bottom of the ship. All ship navigation data were stored in the ship's microcomputer and sent to the shore data exchange server (DES), which is similar to the function of the ship's automatic identification system (AIS). An initial setting server sends initial configuration information to ships through Bluetooth. DES sends ship dynamic data to the control system and receives control commands (rudder angle and propeller speed) and sends them to the corresponding ship model. Figure 15 is a photograph of the experiment tank, wind-making machine, and ship model.

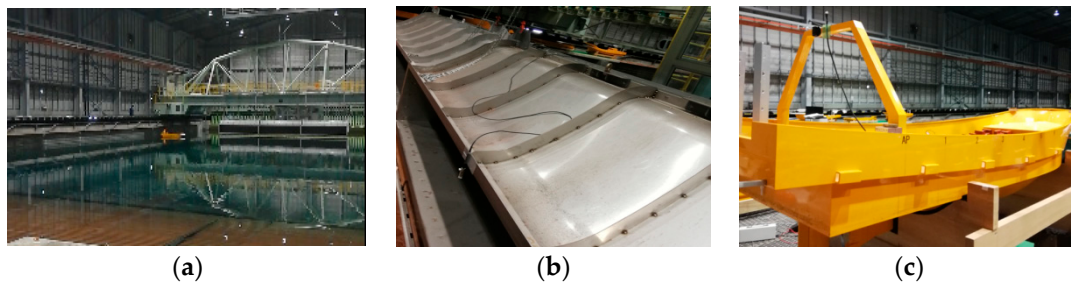


Figure 15. Photographs of the (a) experiment tank, (b) wind-making machine, and (c) ship model.

Because speed control is easy to achieve, in this experiment, the speed of the propeller was kept at a very low speed. The speed of the propeller was maintained at 60 rpm, and the speed of the propeller was 0 after the ship reached the target berth. According to the parameters used in the simulation, the parameters of the controller were adjusted through field experiments. The results of PID: 0.6; 0.001; 1.2. ADRC are as follows: $r = 100$; $h = 0.2$; $a_{01} = 0.5$; $a_{02} = 0.25$; $\delta_1 = 0.1$; $\beta_{01} = 5$; $\beta_{02} = 30$; $\beta_{03} = 70$; $b_0 = 1$; $a_1 = 0.75$; $a_2 = 2.75$; $\delta_2 = 0.1$; $k_1 = 0.35$; $k_2 = 0.01$; $\delta_{01} = 0.1$; $\delta_{02} = 0.1$. In this experiment, there are four typical cases for comparative analysis.

Case 1: The controller uses the PID algorithm, and the experiment was completed without wind disturbance.

Case 2: The controller uses the ADRC algorithm, and the experiment was completed without wind disturbance.

Case 3: The controller uses the PID algorithm, and the experiment was completed with wind whose speed was 1.5 m/s (model scale), representing actual five-level wind.

Case 4: The controller uses the ADRC algorithm, and the experiment was completed with wind whose speed was 1.5 m/s (model scale), representing actual five-level wind.

The experimental results are shown in Figures 16–18. Figure 16 shows ship trajectories in the tank experiment of ship automatic berthing when berthing without wind using PID and ADRC and under wind using PID and ADRC; Figure 17 gives the ship heading angle; Figure 18 shows the ship rudder angle.

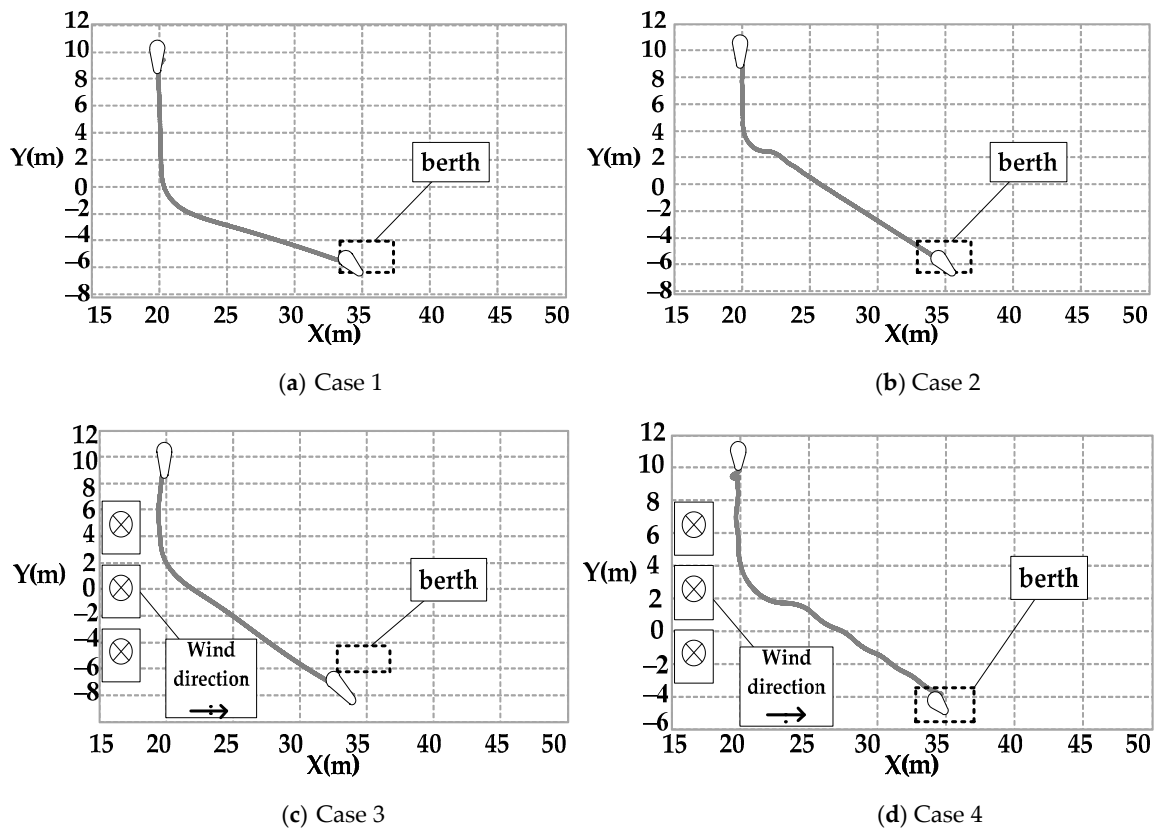


Figure 16. Ship trajectory in the tank experiment of ship automatic berthing.

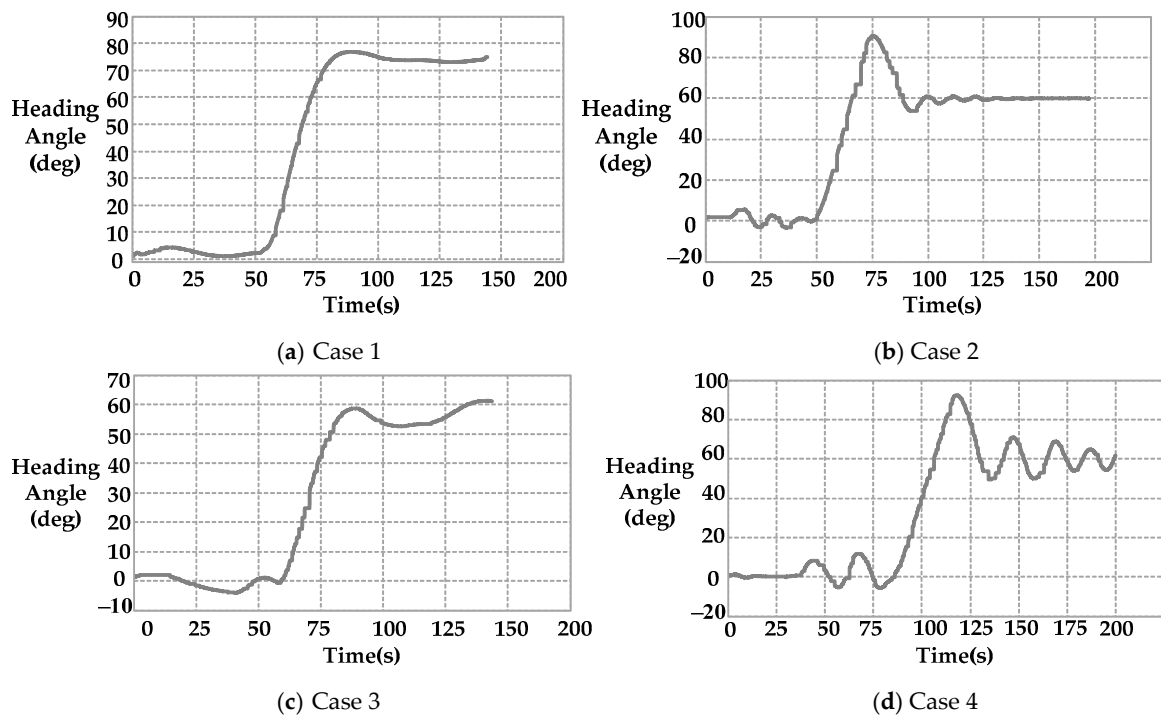


Figure 17. Ship heading angle in the tank experiment of ship automatic berthing.

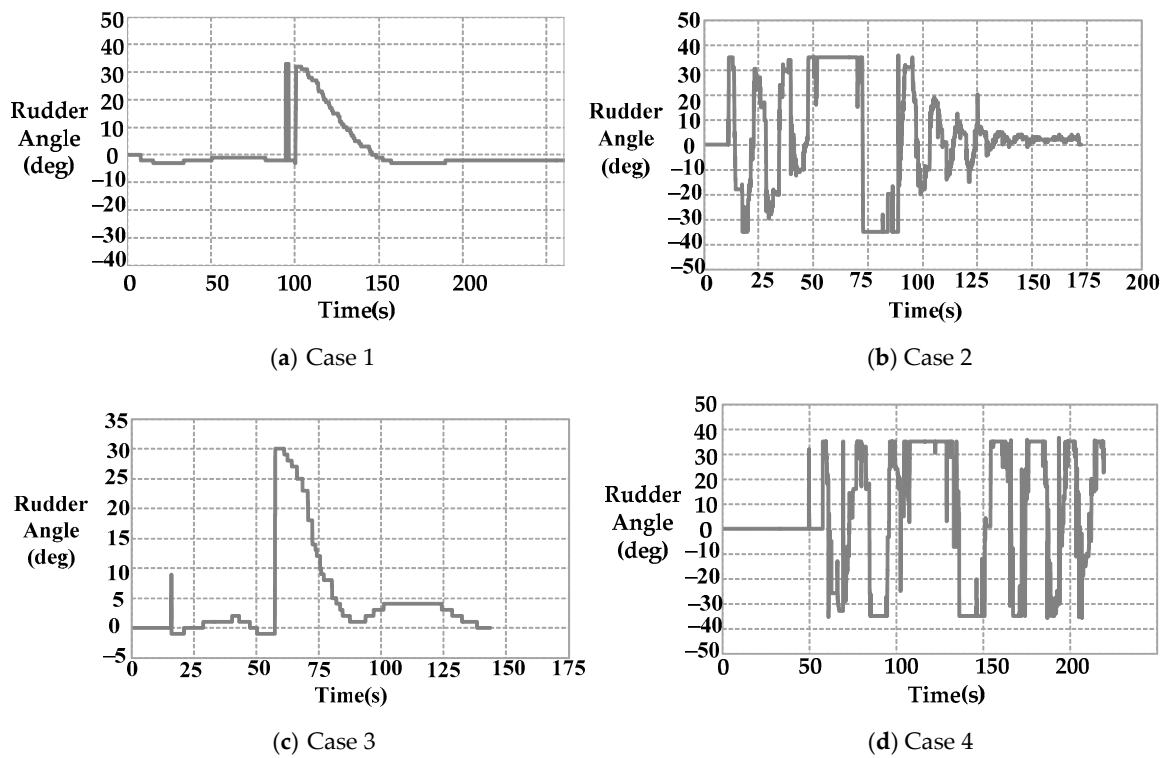


Figure 18. Ship rudder angle in the tank experiment of ship automatic berthing.

From the above results of Figures 16–18, it can be seen that the unmanned ship can reach the target berth smoothly without wind using the PID controller. From the above results of case 2, it can be seen that the ADRC controller can also achieve the desired control effect, but the nonlinear compensation structure of the controller makes the rudder angle change frequently because it is sensitive to the surrounding conditions. The effect of this wind level on the autonomous navigation of unmanned ships is more serious, especially at low speed, because the effect of the rudder is weakened. From the above results of case 3, due to the effect of wind, it is difficult to steer, the ship reacts slowly, and the rudder’s function is mostly due to disturbance resistance, which results in the unmanned ship deviating from the target course and not reaching the designated berth completely. From the above results of case 4, under the action of wind, ADRC has a faster response ability. Because of the existence of TD and ESO, unmanned ships can make faster turns due to the prejudgment and excessive disturbance. Despite the frequent changes of rudder angle, the scheduled berth is reached, which solves the problem that the rudder effect of the unmanned ship is weakened and deviates from the course due to strong wind. Detailed analyses are in Table 6. D is the nearest distance to the center of the berth, and Y/N is whether to complete automatic berthing.

Table 6. Analysis of the results of the experiment of ship automatic berthing.

	Heading Adjustment Time	Time of Arrival at Berth	D	Y/N
Case 1	105.2 s	130.3 s	0.8 m	Y
Case 2	98.4 s	134.5 s	0.3 m	Y
Case 3	110.5 s		2.7 m	N
Case 4	123.5 s	148.1 s	0.6 m	Y

From Table 6, we can see that the results of the experiment are close to those of the simulation, but the adjustment time becomes longer because of the low-speed navigation and the lack of control over RPM in the experiment. In the experiment, it was found that in the case of wind, ship steering becomes more difficult, especially in the case of low speed, and when the wind speed is high to a

certain extent, the ship cannot even steer, so it is of great value to design a better controller under reasonable circumstances. In case 3, the ship fails to pass the berth, and the effect of wind on the result is greater than that in simulation. The control results of cases 1 and 2 are similar, and the ship has successful automatic berthing. The experiment results show that the berthing controller using ADRC is effective, with more appropriate initial parameters and a better control effect. Moreover, the control is still effective in the presence of wind disturbance, which shows that the controller is insensitive to external disturbance and has a certain robustness. In general, the disturbance of wind in the berth of the wharf is relatively small, so the controller can resist such disturbance to achieve the control task.

6. Conclusions

Through this experiment, the effectiveness of the ADRC algorithm, when used in the automatic berthing of an underactuated unmanned ship under wind loads, was verified. This paper mainly draws the following conclusions:

(1) In the process of automatic berthing of unmanned ships, the LOS algorithm is a reliable algorithm. The algorithm is simple and easy to implement, which lays a foundation for further ship motion control.

(2) The mathematical model of ship motion established in this paper has been verified by maneuverability simulation tests. The results obtained by simulation were basically similar to those obtained by experiments and have reference value.

(3) The ADRC algorithm accelerates the ship's response to the disturbances by changing the rudder angle frequently when permitted. It further proves that the intelligent control algorithm can improve the control effect of an unmanned ship.

(4) Through the comparative analysis of experimental results and simulation results, we can see that the situation in the experiment and simulation is not exactly the same. In the actual experiment, there are more nonlinear and unknown factors affecting ship motion. In future simulation analysis, more detailed and comprehensive considerations are needed.

Author Contributions: Conceptualization, C.G. and Z.P.; methodology, Z.P.; software, S.S.; validation, Z.P. and S.S.; formal analysis, Z.P.; investigation, C.G.; resources, C.G.; data curation, Z.P.; writing—original draft preparation, Z.P.; writing—review and editing, C.G.; visualization, S.S.; supervision, Z.P.; project administration, Z.P.; funding acquisition, C.G.

Funding: This research was supported by the National Natural Science Foundation of China (Nos. 51879027, 51579024, 61374114, 51809028) and the fundamental research funds for the central universities (DMU Nos.3132016311, 3132018154).

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

References

1. Rødseth, Ø.J.; Burmeister, H.C. Developments toward the unmanned ship. In Proceedings of the International Symposium Information on Ships–ISIS 2012, Hamburg, Germany, 30–31 August 2012; Volume 201, pp. 30–31.
2. Wróbel, K.; Montewka, J.; Kujala, P. Towards the assessment of potential impact of unmanned vessels on maritime transportation safety. *Reliab. Eng. Syst. Saf.* **2017**, *165*, 155–169. [[CrossRef](#)]
3. Yan, R.; Pang, S.; Sun, H.B.; Pang, Y.J. Development and missions of unmanned surface vehicle. *J. Mar. Sci. Appl.* **2010**, *9*, 451–457. [[CrossRef](#)]
4. Hogg, T.; Ghosh, S. Autonomous merchant vessels: Examination of factors that impact the effective implementation of unmanned ships. *Aust. J. Marit. Ocean Aff.* **2016**, *8*, 206–222. [[CrossRef](#)]
5. Martin, A.Y. Unmanned maritime vehicles: Technology evolution and implications. *Mar. Technol. Soc. J.* **2013**, *47*, 72–83. [[CrossRef](#)]
6. 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](#)]
7. 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](#)]

8. Park, J.Y.; Kim, N. Design of an adaptive backstepping controller for auto-berthing a cruise ship under wind loads. *Int. J. Nav. Archit. Ocean Eng.* **2014**, *6*, 347–360. [[CrossRef](#)]
9. Tamaru, H.; Hagiwara, H.; Yoshida, H.; Tasaki, T.; Miyabe, H. Development of automatic berthing system for kaisho maru and its performance evaluation. *J. Jpn. Inst. Navig.* **2005**, *113*, 157–164. [[CrossRef](#)]
10. Kim, T.H.; Basar, T.; Ha, I.J. Asymptotic stabilization of an underactuated surface vessel via logic-based control. In Proceedings of the 2002 American Control Conference (IEEE Cat. No. CH37301), Anchorage, AK, USA, 8–10 May 2002; Volume 6, pp. 4678–4683.
11. Yang, L.; Chen, G. Automatic berthing control of underactuated surface ships in restricted waters based on nonlinear adaptive control method. In Proceedings of the 31st Chinese Control Conference, Hefei, China, 25–27 July 2012; pp. 939–944.
12. Namkyun, I.M.; Hasegawa, K. A study on automatic ship berthing using parallel neural controller. *J. Kansai Soc. Nav. Archit. Jpn.* **2002**, *237*, 127–132.
13. Moreira, L.; Fossen, T.I.; Soares, C.G. Path following control system for a tanker ship model. *Ocean Eng.* **2007**, *34*, 2074–2085. [[CrossRef](#)]
14. Jin, X. Fault tolerant finite-time leader–follower formation control for autonomous surface vessels with LOS range and angle constraints. *Automatica* **2016**, *68*, 228–236. [[CrossRef](#)]
15. Liu, Y.; Bu, R.; Gao, X. Ship Trajectory Tracking Control System Design Based on Sliding Mode Control Algorithm. *Pol. Marit. Res.* **2018**, *25*, 26–34. [[CrossRef](#)]
16. Garcia-Aunon, P.; Peñas, M.S. A new UAV ship-tracking algorithm. *IFAC-PapersOnLine* **2017**, *50*, 13090–13095. [[CrossRef](#)]
17. Johansen, T.A.; Cristofaro, A.; Perez, T. Ship collision avoidance using scenario-based model predictive control. *IFAC-PapersOnLine* **2016**, *49*, 14–21. [[CrossRef](#)]
18. Wilson, P.A.; Harris, C.J.; Hong, X. A line of sight counteraction navigation algorithm for ship encounter collision avoidance. *J. Navig.* **2003**, *56*, 111–121. [[CrossRef](#)]
19. Yasukawa, H.; Yoshimura, Y. Introduction of MMG standard method for ship maneuvering predictions. *J. Mar. Sci. Technol.* **2015**, *20*, 37–52. [[CrossRef](#)]
20. Zhang, X.K. *Ship Motion Control*; National Defense Industry Press: Beijing, China, 2006; pp. 11–17.
21. Clarke, D. New non-linear Equation for ship manoeuvring. *Int. Shipbuild. Prog.* **1971**, *18*, 181–197. [[CrossRef](#)]
22. Fang, M.C.; Luo, J.H.; Lee, M.L. A nonlinear mathematical model for ship turning circle simulation in waves. *J. Ship Res.* **2005**, *49*, 69–79.
23. Fujiwara, T.; Ueno, M.; Nimura, T. Estimation of wind forces and moments acting on ships. *J. Soc. Nav. Archit. Jpn.* **1998**, *1998*, 77–90. [[CrossRef](#)]
24. Zhang, X.K.; Zhang, Q.; Ren, H.X.; Yang, G.P. Linear reduction of backstepping algorithm based on nonlinear decoration for ship course-keeping control system. *Ocean Eng.* **2018**, *147*, 1–8. [[CrossRef](#)]
25. Han, J. From PID to active disturbance rejection control. *IEEE Trans. Ind. Electron.* **2009**, *56*, 900–906. [[CrossRef](#)]
26. Xue, W.; Bai, W.; Yang, S. ADRC with adaptive extended state observer and its application to air–fuel ratio control in gasoline engines. *IEEE Trans. Ind. Electron.* **2015**, *62*, 5847–5857. [[CrossRef](#)]
27. Qian, M.; Wu, Z.; Han, J. Real-time simulation and analyses of ship energy management system network. *Energy Procedia* **2012**, *16*, 1972–1978. [[CrossRef](#)]
28. Liu, X.Z.; Zhou, J.; Liang, D.Q. Huge Data Blocks Transmission Based on UDP. *Dianxun Jishu Telecommun. Eng.* **2012**, *52*, 96–100.
29. Rodseth, O.J.; Christensen, M.J.; Lee, K. Design challenges and decisions for a new ship data network. *ISIS* **2011**, 15–16.
30. Kim, Y.G.; Kim, S.Y.; Kim, H.T.; Lee, S.W.; Yu, B.S. Prediction of the maneuverability of a large container ship with twin propellers and twin rudders. *J. Mar. Sci. Technol.* **2007**, *12*, 130–138. [[CrossRef](#)]
31. Wang, X.; Liu, Z.; Cai, Y. The ship maneuverability based collision avoidance dynamic support system in close-quarters situation. *Ocean Eng.* **2017**, *146*, 486–497. [[CrossRef](#)]

