

Article

Fractional-Order $PI^{\lambda}D^{\mu}$ Controller Using Adaptive Neural Fuzzy Model for Course Control of Underactuated Ships

Guangyu Li ¹, Baojie Chen ², Huayue Chen ^{3,*} and Wu Deng ^{4,5,*}¹ School of Software, Dalian Jiaotong University, Dalian 116026, China; ligyu@djtu.edu.cn² Haifeng General Aviation Technology Company Ltd., Beijing 100070, China; chenbaojie@hfga.com.cn³ School of Computer Science, China West Normal University, Nanchong 637002, China⁴ School of Electronic Information and Automation, Civil Aviation University of China, Tianjin 300300, China⁵ Traction Power State Key Laboratory of Southwest Jiaotong University, Chengdu 610031, China

* Correspondence: sunnyxiaoyue20@cwnu.edu.cn (H.C.); wdeng@cauc.edu.cn (W.D.)

Abstract: For the uncertainty caused by the time-varying modeling parameters with the sailing speed in the course control of underactuated ships, a novel identification method based on an adaptive neural fuzzy model (ANFM) is proposed to approximate the inverse dynamic characteristics of the ship in this paper. This model adjusts both its own structure and parameters as it learns, and is able to automatically partition the input space, determine the number of membership functions and the number of fuzzy rules. The trained ANFM is used as an inverse controller, in parallel with a fractional-order $PI^{\lambda}D^{\mu}$ controller for the course control of underactuated ships. Meanwhile, the sine wave curve and the sawtooth wave curve are considered as the input learning samples of ANFM, respectively, and the inverse dynamics simulation experiments of the ship are carried out. Two different ANFM structures are obtained, which are connected in parallel with the fractional-order $PI^{\lambda}D^{\mu}$ controller respectively to control the course of ship. The simulation results show that the proposed method can effectively overcome the influence of uncertainty of ship modeling parameters, track the desired course quickly and effectively, and has a good control effect. Finally, comparative experiments of four different controllers are carried out, and the results show that the FO $PI^{\lambda}D^{\mu}$ controller using ANFM has the advantages of small overshoot, short adjustment time, and precise control.

Keywords: adaptive neural fuzzy model; system identification; inverse dynamic characteristics; fractional-order $PI^{\lambda}D^{\mu}$ controller; underactuated ships; course control



Citation: Li, G.; Chen, B.; Chen, H.; Deng, W. Fractional-Order $PI^{\lambda}D^{\mu}$ Controller Using Adaptive Neural Fuzzy Model for Course Control of Underactuated Ships. *Appl. Sci.* **2022**, *12*, 5604. <https://doi.org/10.3390/app12115604>

Academic Editor: Federico Divina

Received: 24 April 2022

Accepted: 30 May 2022

Published: 31 May 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. 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 (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, the problem of motion control of underactuated ships has been a hot research topic [1–6]. Ships rely on two control quantities, the turning torque generated by the rudder device and the longitudinal propulsion force generated by the main propeller, to simultaneously control the movement of the ship in three degrees of freedom in both horizontal plane position and course, which constitute an underactuated system [7]. The dynamics of underactuated ships have the characteristics of large inertia, large time delay, nonlinearity, and the uncertainty problems which have always been concerned [8–11]. Changes in navigation conditions and the influence of external disturbances have caused uncertainty in the parameters of the ship's mathematical model and even structural perturbations, which have always been a problem for researchers [12–16]. However, most of the existing linear or nonlinear ship's motion control methods are required to be based on accurate mathematical models, which affect the performance of the control system. However, due to the uncertainties inherent in the controlled object of a ship, these methods usually fail to achieve the expected research results. Therefore, it is very important to explore new control methods to solve the uncertainty problem in ship motion control.

Due to the complexity and uncertainty of the model of underactuated ships, artificial neural networks can be used to approximate their inverse dynamic characteristics. Artificial neural networks have approximation ability of any smooth nonlinear function with the required accuracy. The unique advantages of artificial neural networks make them powerful tools for solving such problems. Many methods based on artificial neural networks can be used in the design of autopilot for ships. These methods can be divided into two kinds. The first one is to train the artificial neural networks controller using another controller [17–20]. The main problems with this method are that another well-known controller is required and the lack of adaptation, because a supervisor controller based on a concrete regime is designed. To improve this situation, the controllers designed for different operating conditions can provide an alternative to adaptive control or gain scheduling in this application. The second one is to use the artificial neural networks identification approach [21–28]. An approximator based on artificial neural networks is proposed to evaluate the uncertain parameters of the mathematical model of the ship for further controller design using other control techniques. For example, adaptive backstepping and backstepping-based integral sliding mode control were combined with artificial neural networks and achieved steering control over the ship. A Radian Basic Function (RBF) neural network was proposed with fuzzy logic control [29–35].

Taking advantage of the functional equivalence of neural networks and fuzzy logic systems, the neural fuzzy stable adaptive control method can inherit the equivalent results in adaptive control of neural networks, and the converse is also true. Neural fuzzy systems are mainly divided into two categories: static neuro-fuzzy systems and dynamic neuro-fuzzy systems. The static neural fuzzy system is a neural network implementation of the conventional fuzzy logic system, and there are three main types of representative ones. The first one, fuzzy adaptive learning control network (FALCON) [36], was first proposed by Lin to study the hybrid structural, parametric learning strategy. FALCON integrates the basic elements and functions of conventional fuzzy control into a connectionist structure, constituting a forward multilayer neural network with a distributed learning structure. The second one, a fuzzy basis function network (FBFN) [37] was proposed by Wang and Menel. In a fuzzy basis function network, the fuzzy system is represented by a level expansion of fuzzy basis functions, which is an algebraic iteration of the affiliation function. Each fuzzy basis function represents a fuzzy rule. The third one, adaptive network-based fuzzy inference system (ANFIS) [38] proposed by Jang, is similar to the T-S fuzzy model in structure and function, and its fuzzy rule posterior is a linear combination of anterior states. The cooperative neural fuzzy Inference System (CANFIS) is a generalization of the single-output ANFIS with multi-output nonlinear fuzzy rules. Due to the functional equivalence and the complementarity of the mechanism between the neural network and the fuzzy system, the combination of the two, the neuro-fuzzy system, is widely used in the field of ship motion control and has achieved certain results [39–46]. In [39], a feed-forward multilayered architecture of artificial neural networks was proposed and applied to approximate the inverse model of the ship, which used a cost function to generate artificial neural network training data. In [40], a controller based on an artificial neural network was proposed for automatic ship docking. Furthermore, the structure of the artificial neural network was optimized by genetic algorithm. By generating reliable docking data, the simulation module of ship docking was established. In [41], a deep learning method was applied in the ship trajectory data restoration of an automatic identification system. By using bi-directional long short-term memory recurrent neural networks, when multiple points were missing, ship trajectories could be restored. In [42], a corrective system based on artificial neural networks was used for the coordinates of the ship's position and predicting ship trajectory. In [43], a deep neural network based on long-short-term-memory was proposed to identify a ship motion model and predict ship maneuvering motion. In [44], a course controller based on two-multilayered feed-forward neural network was proposed. The first neural network was used for the ship forward dynamic approximator, and the second neural network was used for the ship's course. In [47], a fuzzy PID control

was applied to the ship's autopilot. In [48], three fuzzy controllers were proposed to ship berthing. The first one was used to control the longitudinal direction movement of ship toward to the normal of wharf by propeller. The second one was used to compute the relative bearing error with tugboat. The last one was designed for bringing the ship into wharf. In [49], a scheme based on observer-based adaptive fuzzy-output feedback was developed and applied for ship autopilot. In [50], to adjust the transient performance, a fuzzy logic system was proposed. In addition, a linear course-keeping controller based on second-order closed-loop gain shaping algorithm was designed. However, these methods are generally based on static networks; the network structure is first determined or the number of fuzzy rules is pre-given, then the modeling accuracy is evaluated using generalization errors and then directly used for ship motion control. Such algorithms inherently have large uncertainties and limitations due to the use of trial-and-error methods or relying on the knowledge of domain experts to determine structures or rules.

Digital PID autopilot is a commonly used controller for ship course control. However, this automatic rudder is too sensitive for ship models and high-frequency disturbances, which leads to frequent steering, and it is difficult to achieve a good control effect [47]. With the development of control theory, fractional calculus theory [51] was used in controller design, resulting in a fractional-order $PI^\lambda D^\mu$ controller [52]. In 1999, Podlubny I first proposed a fractional-order $PI^\lambda D^\mu$ controller. The introduction of the integral order λ and the differential order μ makes the controller have two more adjustable parameters, making the controller more flexible and robust. In [53], Pritesh Shah introduced the design and tuning methods of the fractional-order $PI^\lambda D^\mu$ controller, and software tools were also discussed. In [54], different types of variable-order fractional operators were defined and fractional-order $PI^\lambda D^\mu$ controllers for linear dynamical systems was designed. In [55], Davut Izci introduced fractional-order $PI^\lambda D^\mu$ controllers based on an opposition-based hunger games search algorithm, and applied it to a magnetic ball suspension system. In [56], a fractional-order $PI^\lambda D^\mu$ controllers based on combination of Lévy flight distribution and simulated annealing algorithms was applied for buck converters.

Considering the discussion provided above, in this article, we aim to propose a network structure of adaptive neural fuzzy models to identify the inverse dynamics in the course control of underactuated ships, taking the underactuated ships as the research objects and addressing the uncertainty problem caused by the modeling parameters in the ship's course control with the time-varying sailing speed. The trained adaptive neural fuzzy model is also used as an inverse controller in parallel with a fractional-order $PI^\lambda D^\mu$ controllers for ships' course control. In order to illustrate the effectiveness of the proposed design, comparative simulations are conducted to verify the advantages of the proposed design. Above all, the main contributions can be summarized as follows:

- (i) A novel identification method based on adaptive neural fuzzy model (ANFM) was proposed to approximate the inverse dynamic characteristics of an underactuated ship. The model had good generalization ability and could automatically divide the input space, determine the number of membership functions and the number of fuzzy rules.
- (ii) Sine wave curve and sawtooth wave curve were used as input learning samples of ANFM, respectively, and inverse dynamics simulation experiments of ship were carried out to obtain two different sets of ANFM structures.
- (iii) The trained ANFM was used as an inverse controller in parallel with a fractional-order $PI^\lambda D^\mu$ controllers. The ANFM-FOPID controller for course control was constructed and the effectiveness was evaluated by simulation.
- (iv) The effectiveness was further evaluated by comparative experiments of four different controllers, which are the PID controller [57], the PID controller based on PSO algorithm [58], the FO $PI^\lambda D^\mu$ controller based on GA algorithm [59] and the FO $PI^\lambda D^\mu$ controller based on ANFM (proposed).

2. Motion Model for Underactuated Ships

The ship movement has six degrees of freedom (DOF), which denote position and direction. In the inertial frame, (x, y, z) denotes the position, and first-order differentiation of (x, y, z) denotes translational motion along x , y and z . Surge, sway and heave are the first three DOFs, (φ, θ, ψ) denotes the directions of ship, and first-order differentiation of (φ, θ, ψ) denotes rotation of the ship. Roll, pitch and yaw are the last three DOF. The ship motion description is shown in Figure 1.

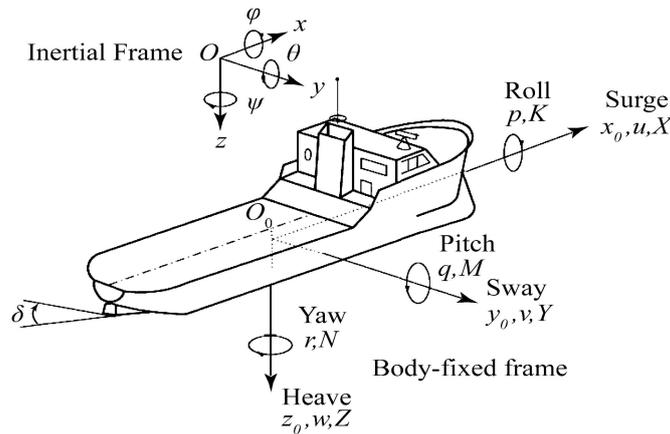


Figure 1. Ship motion model.

For most control problems of ships, only sway, surge, and yaw are discussed. Therefore, the ship’s motion is regarded as a plane motion with three degrees of freedom. The maneuvering model of the ship [6] can be shown as:

$$\begin{cases} \dot{x} = u \cos \psi - v \sin \psi \\ \dot{y} = u \sin \psi + v \cos \psi \\ \dot{\psi} = r \end{cases} \tag{1}$$

$$\begin{cases} \dot{u} = \frac{m_{22}}{m_{11}}vr - \frac{d_{11}}{m_{11}}u + \frac{1}{m_{11}}\tau_u \\ \dot{v} = -\frac{m_{11}}{m_{22}}ur - \frac{d_{22}}{m_{22}}v \\ \dot{r} = \frac{m_{11}-m_{22}}{m_{33}}uv - \frac{d_{33}}{m_{33}}r + \frac{1}{m_{33}}\tau_r \end{cases} \tag{2}$$

where x -surge, y -sway, ψ -course angle, u -surge velocity, v -sway velocity. r -yaw velocity; τ_u -longitudinal force of propeller, τ_r -the torque of propeller; m_{11} , m_{22} , m_{33} , d_{11} , d_{22} and d_{33} are uncertainty parameters.

For course control of ships, the nonlinear model is a dynamic system. The responding nonlinear ship model is shown in Figure 2. Input is rudder angle δ , output is course angle ψ . By capturing the ship dynamics from $\delta \rightarrow \dot{\psi} \rightarrow \psi$, the differential equation retains nonlinear influences and allows wind and wave disturbances to be converted into a disturbing rudder angle δ_D as an input signal that enters the ship model with the actual rudder angle δ .

The Norrbin nonlinear model [57] for course control of underactuated ships can be expressed as

$$\ddot{\psi} = -a_1\dot{\psi} - a_2\dot{\psi}^3 + b\delta \tag{3}$$

where ψ and δ are the course angle and the control rudder angle respectively. $a_1 = \alpha b$, $a_2 = \beta b$, $b = K/T$ are model parameters, K and T are ship indices, α and β are nonlinear coefficients, parameter K is proportional to speed V , while parameter T is inversely proportional to speed V .

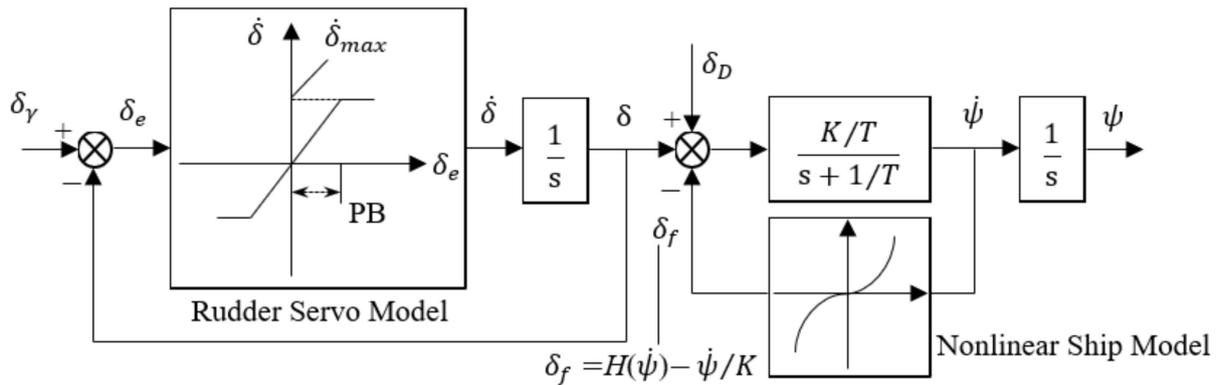


Figure 2. Responding nonlinear ship model.

The model uncertainty conditions discussed in this section refer to the problem of uncertainty in the four parameters K, T, α, β mentioned above that vary with the speed V , thus causing a time-varying uncertainty in the modeling parameters.

Based on the NARX model [60], without considering the noise, Equation (3) can be discretized and transformed as follows,

$$\psi(k + 1) = c_1\psi(k) + c_2\psi(k - 1) + c_3\psi(k - 2) + c_4[\psi(k - 1) - \psi(k - 2)]^3 + c_5\delta(k - 2) \tag{4}$$

where $c_1 = 2, c_2 = -(1 + a_1h), c_3 = a_1h, c_4 = -a_2/h, c_5 = bh^2, h$ is the sampling period and k is the sampling moment. Obviously, the variation of parameters K, T, α, β lead to the variation of parameter $c_i (i = 1, \dots, 5)$, which makes the mathematical model of ship course control uncertain.

By deforming Equation (4) and simplifying the problem, the inverse model of the ship’s motion can be deduced as

$$\delta(k + 1) = d_1\psi(k) + d_2\psi(k - 1) + d_3\psi(k - 2) + d_4[\psi(k) - \psi(k - 1)]^3 \tag{5}$$

where $d_1 = (1 + a_1h)/bh^2, d_2 = -(2 + a_1h)/bh^2, d_3 = 1/bh^2, d_4 = a_2/bh^3$.

3. Adaptive Neural Fuzzy Model

The adaptive neural fuzzy model (ANFM) proposed in this paper is a neural network implementation of a dynamic fuzzy logic system. Because dynamic fuzzy logic systems contain feedback of states or output quantities, their connectionist structure corresponds to recurrent neural networks (RRN). Therefore, ANFM is more suitable for modeling and control of nonlinear dynamic systems. This paper proposes an ANFM for approximating the inverse dynamics of the ship’s course. The “adaptive” nature of ANFM means that the structure of the model is not pre-determined, but adjusted simultaneously with the parameters as it is learned. Meanwhile, the number of fuzzy rules is also uncertain, the fuzzy rules grow gradually during the learning process, the number of rules does not increase with the number of input variables, and automatic modeling can be achieved without the need for expert knowledge.

3.1. ANFM Structure

In Figure 3, the structure of ANFM is a “deformed” RBF neural network, which is functionally equivalent to a T-S fuzzy model. The “deformation” mainly refers to the fact that the network has more than three layers, and the weights are a function of the input and not constants.

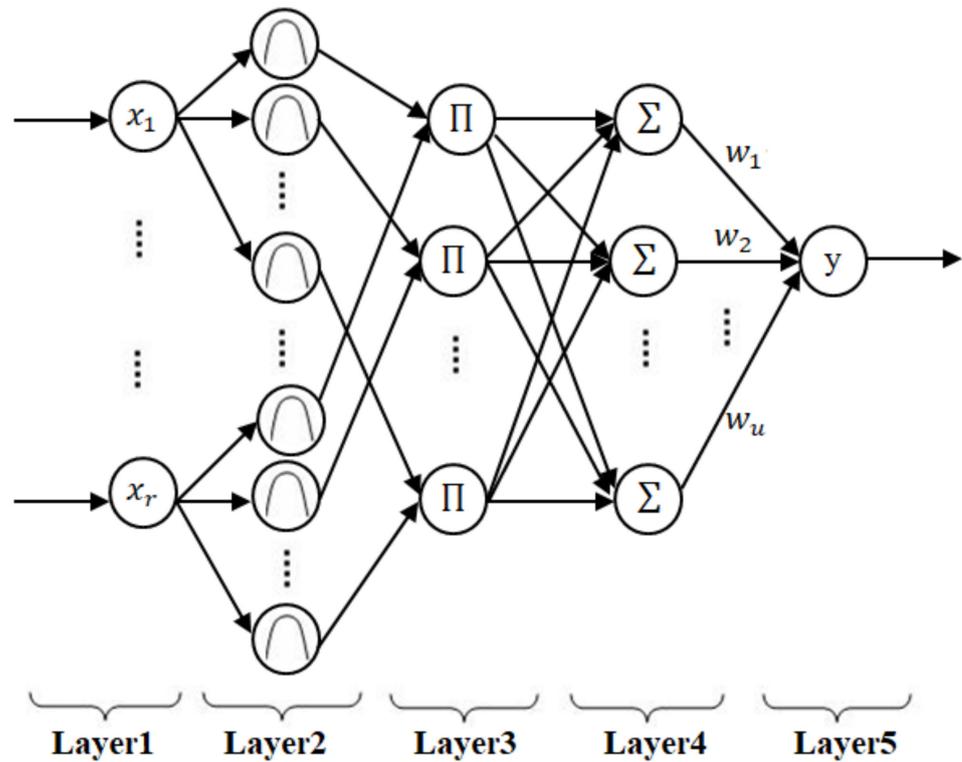


Figure 3. ANFM structure.

Layer 1: The input layer, where each node represents an input linguistic variable respectively, and the total number of input variables is r . In this paper, the output course angle $\psi(k)$ generated by the ship, its previous step $\psi(k - 1)$ and rudder angle $\delta(k)$ are used as sample data for the training of ANFM.

Layer 2: The membership function layer, where each node represents a respective RBF membership function. In this paper, the Gaussian function is chosen as the membership function of ANFM.

$$\mu_{ij}(x_i) = \exp\left[-\frac{(x_i - c_{ij})^2}{\sigma_j^2}\right] \tag{6}$$

where, μ_{ij} is the j th membership function of x_i , c_{ij} is the center of the j th Gaussian membership function of x_i , σ_j is the width of the j th Gaussian membership function of x_i , and u is the number of membership functions for each variable.

Layer 3: the fuzzy rule layer, where each node represents a possible antecedent of a fuzzy rule, namely the IF part. Here, the T-norm product operator is used, which is multiplied by the Gaussian membership function of each input variable, and the width of the membership function of each variable corresponding to each rule is the same.

$$\varphi_j = \exp\left[-\frac{\sum_{i=1}^r (x_i - c_{ij})^2}{\sigma_j^2}\right] \tag{7}$$

Which is

$$\varphi_j = \exp\left[-\frac{\|\mathbf{X} - \mathbf{C}_j\|^2}{\sigma_j^2}\right] \tag{8}$$

where, $\mathbf{X} = (x_1, x_2, \dots, x_r) \in \mathbb{R}^r$, $\mathbf{C}_j = (c_{1j}, c_{2j}, \dots, c_{rj}) \in \mathbb{R}^r$ is the center of the j th RBF neural unit. In terms of neural networks, each node in the layer is an RBF neural unit node and the number of neural units is time-varying in the dynamic process. Therefore, in

ANFM, the number of RBF neural unit nodes is equal to the number of rules of the fuzzy system, and the two concepts are interchangeable.

Layer 4: Normalization layer, where each node represents an N node respectively. Each N node is the output of its corresponding previous layer of rules as a proportion of the sum of the outputs of all rules. It is easy to know that the number of N nodes is equal to the number of fuzzy rules. The output of the j th node N_j is

$$\phi_j = \frac{\varphi_j}{\sum_{k=1}^u \varphi_k} \tag{9}$$

Layer 5: The output layer, where each node represents a respective output variable. This layer represents the latter part of the fuzzy rule, namely the THEN part. In this paper, only the single output mode is used as an example for derivation. The output of ANFM is a linear superposition of all the input signals of the previous layer according to the connection weights.

$$y(X) = \sum_{k=1}^u \omega_k \phi_k \tag{10}$$

where, y is the output variable and ω_k is the connection weight of the k th ($k = 1, 2, \dots, u$) rule. In ANFM, the connection weights are not simply real constants, but a function of all input variables about the whole ANFM, expressed as a linear combination of each input variable according to a corresponding set of weight coefficients, with the following expression.

$$\omega_k = a_{k0} + a_{k1}x_1 + a_{k2}x_2 + \dots + a_{kr}x_r \tag{11}$$

Substituting Equations (8), (9) and (11) into Equation (10), the detailed expression for the output variable is obtained as

$$y(X) = \frac{\sum_{i=1}^u [(a_{i0} + a_{i1}x_1 + \dots + a_{ir}x_r) \exp(-\frac{\|X-C_i\|^2}{\sigma_i^2})]}{\sum_{i=1}^u \exp(-\frac{\|X-C_i\|^2}{\sigma_i^2})} \tag{12}$$

The “adaptive” nature of ANFM focuses on the number of membership functions for each input variable in layer 2, the number of fuzzy rules or RBF neural units in layer 3, the number of normalized nodes in layer 4 and the number of connection weights in layer 5, all of which are equal and time-varying, denoted here by u . During the modeling or control process, u is constantly changing according to a specific learning algorithm, so that the entire middle three layers of the ANFM structure is also constantly changing, thus being able to learn the uncertainty of the ship model and thus produce the appropriate output to obtain a satisfactory modeling result.

3.2. ANFM Learning Algorithm

Because the structure of the 2nd, 3rd and 4th layers of ANFM changes all the time, the learning algorithm of ANFM in this paper identifies the structure of ANFM from three aspects: generating fuzzy rules, determining weights and pruning fuzzy rules, so that it can automatically to determine the fuzzy rules and achieve the specific performance of the system.

3.2.1. Generation of Fuzzy Rules

In ANFM, the number of fuzzy rules directly affects the identification ability of the system. Too few fuzzy rules will not fully encompass the input/output state space and the performance of ANFM will deteriorate; too many fuzzy rules will increase the complexity of the system and lead to poor generalization of ANFM. Therefore, output error is an important factor in determining whether a new rule should be generated.

For the i th observation (X_i, o_i) , where X_i is the input vector and o_i is the desired output, the full output y_i of the existing structure of ANFM can be calculated according to Equation (12).

$$\|e_i\| = \|o_i - y_i\| \tag{13}$$

If $\|e_i\| > k_e$, add a new rule. Here k_e is the error index, the value of which is predetermined according to the desired accuracy of ANFM.

In addition, the coverage of the Gaussian function is another important factor in determining whether a new rule should be added.

For the i th observation (X_i, o_i) , calculate the distance $d_i(j)$ between the input value X_i and the centre C_j of the existing RBF neural unit, that is

$$d_i(j) = \|X_i - C_j\| \tag{14}$$

Find out

$$d_{min} = \operatorname{argmin}(d_i(j)) \tag{15}$$

and this d_{min} is the coverage scope. If $d_{min} > k_d$, a new fuzzy rule should be considered. Otherwise, the observed data X_i can then be represented by the nearest available RBF neural unit. Here k_d denotes the effective radius of d_{min} .

Thus, the error index k_e and the effective radius k_d play important roles in determining the generation of new rules. In this paper, we adopt the idea of “graded learning” [61], based on a monotonically decreasing function, gradually reducing the error index k_e and the effective radius k_d of each RBF neural unit as follows.

$$k_e = \max[e_{max} \times \beta^i, e_{min}] \tag{16}$$

$$k_d = \max[d_{max} \times \gamma^i, d_{min}] \tag{17}$$

where e_{max} is the predefined maximum error, e_{min} is the desired accuracy of ANFM, and $0 < \beta < 1$ is the convergence constant; d_{max} is the maximum length of the input space, d_{min} is the minimum length, and $0 < \gamma < 1$ is the decay constant. These parameters will be set prior to the ANFM learning. The key idea of graded learning is to first determine the position that generates a large output error without being covered by the existing fuzzy rules. This stage is called rough learning. When k_e and k_d reaches e_{min} and d_{min} , respectively, this stage is called detailed learning. Because the width of the RBF unit is very important for the generalization capacity of the system, the width is too small to fully divide the input space, making the system’s generalization ability worse. The width is too large and easily falls into saturation, and the correct output cannot be produced. Therefore, the initial parameters of the new rules are determined by the formulas.

$$C_i = X_i \tag{18}$$

$$\sigma_i = k_s \times d_{min} \tag{19}$$

where, k_s is a predetermined overlap factor; the width of the first rules is also a pre-set constant. From (16) and (17), it can be seen that graded learning is the transition process from rough to detailed learning. Only when $\|e_i\| > k_e$ and $d_{min} > k_d$, an additional fuzzy rule is needed. For the other three cases, the followings are discussed:

- (i) $\|e_i\| \leq k_e$ and $d_{min} \leq k_d$, ANFM can fully accommodate the observation data, no need to do anything;
- (ii) $\|e_i\| \leq k_e$ and $d_{min} > k_d$, this situation indicates that the establishment of ANFM has a good generalization ability, and only the parameters of the post-rules need to be adjusted;
- (iii) $\|e_i\| > k_e$ and $d_{min} \leq k_d$, this situation indicates that the generalization ability of the RBF neural unit covering X_i is not very good. Therefore, the RBF neural unit and

results parameters will be updated at the same time. For the k th RBF neural unit closest to X_i , adjust by the following equation.

$$\sigma_k^j = k_\omega \times \sigma_k^{j-1} \tag{20}$$

where, k_ω is a pre-set constant and $k_\omega > 1$.

3.2.2. Determination of Weights

Assuming that n observations produce u fuzzy rules, the network output of the fourth layer can be obtained from Equation (9), which can be expressed in matrix form as follows.

$$\phi = \begin{bmatrix} \phi_{11} & \cdots & \phi_{1n} \\ \vdots & \vdots & \vdots \\ \phi_{u1} & \cdots & \phi_{un} \end{bmatrix} \tag{21}$$

For any input $X_j(x_{1j}, x_{2j}, \dots, x_{rj})$, the output y_j of the system is calculated from Equation (10), which can be rewritten in compact form as follows.

$$Y = W\Psi \tag{22}$$

$$\tilde{E} = ||O - Y|| \tag{23}$$

where W and Ψ are given by the following equation,

$$W = (a_{10} \ \cdots \ a_{u0} \ a_{11} \ \cdots \ a_{u1} \ \cdots \ a_{1r} \ \cdots \ a_{ur}) \tag{24}$$

$$\Psi = \begin{bmatrix} \phi_{11} & \cdots & \phi_{1n} \\ \vdots & \vdots & \vdots \\ \phi_{u1} & \cdots & \phi_{un} \\ \phi_{11} \cdot x_{11} & \cdots & \phi_{1n} \cdot x_{1n} \\ \vdots & \vdots & \vdots \\ \phi_{u1} \cdot x_{11} & \cdots & \phi_{un} \cdot x_{1n} \\ \vdots & \vdots & \vdots \\ \phi_{11} \cdot x_{r1} & \cdots & \phi_{1n} \cdot x_{rn} \\ \vdots & \vdots & \vdots \\ \phi_{u1} \cdot x_{r1} & \cdots & \phi_{un} \cdot x_{rn} \end{bmatrix} \tag{25}$$

The ideal output is assumed to be $O = (o_1, o_2, \dots, o_n) \in \mathfrak{R}^n$. The Linear least-square is used to approximate an optimal weight vector $\Psi^* \in \mathfrak{R}^{u \times (r+1)}$, such that the error is minimized.

$$W^* = O(\Psi^T\Psi)^{-1}\Psi^T \tag{26}$$

3.2.3. Pruning of Fuzzy Rules

Pruning techniques are necessary for the identification of dynamic time-varying nonlinear systems in order to avoid overfitting. In this section, the error rate of decline method [62] is used to prune the fuzzy rules, and the basis for pruning is the importance of the fuzzy rules.

Given n pairs of input/output data $(X_i, o_i), i = 1, 2, \dots, n$, Equations (22) and (23) are used as a special case of the linear regression model.

$$D = H\theta + E \tag{27}$$

where, $D = O^T \in \mathfrak{R}^n$ is the desired output, $H = \Psi^T = (h_1 \cdots h_v) \in \mathfrak{R}^{n \times v}, v = u \times (r + 1)$ is called the regression vector, $\theta = W^T \in R^v$ contains the parameters and $E \in R^n$ is

assumed to be an error vector uncorrelated with the regression quantity h_i . For a matrix H , by QR decomposition.

$$H = QA \tag{28}$$

Substituting Equation (28) into Equation (27) gives

$$D = QA\theta + E = QG + E \tag{29}$$

The linear least-squares solution for G is

$$G = (Q^T Q)^{-1} Q^T D \tag{30}$$

Or

$$g_i = \frac{q_i^T D}{q_i^T q_i} \tag{31}$$

where, $i = 1, 2, \dots, v$ and G and θ satisfy the following equations.

$$A\theta = G \tag{32}$$

when $i \neq j$, since q_i and q_j are orthogonal, the sum of squares of D is given by

$$D^T D = \sum_{i=1}^v g_i^2 q_i^T q_i + E^T E \tag{33}$$

After removing the mean, the variance of D is given by the following equation.

$$n^{-1} D^T D = n^{-1} \sum_{i=1}^v g_i^2 q_i^T q_i + n^{-1} E^T E \tag{34}$$

Therefore, the rate of error decline can be defined as

$$err_i = \frac{g_i^2 q_i^T q_i}{D^T D} \tag{35}$$

Substituting Equation (31) into Equation (35), we get

$$err_i = \frac{(q_i^T D)^2}{q_i^T q_i D^T D} \tag{36}$$

Let ϕ_i be the angle between vector q_i and D . Then

$$\cos^2 \phi_i = err_i = \frac{(q_i^T D)^2}{q_i^T q_i D^T D} \tag{37}$$

If $\phi_i = 0$ and $err_i = 1$, it means that the influence is maximum. Otherwise $\phi_i = 90^\circ$, if $err_i = 0$, the two vectors are orthogonal, indicating that q_i has no effect on D .

Rearrange the newly obtain $err_i (i = 1, 2, \dots, (r + 1)u)$ into a matrix $\Delta = (\delta_1, \delta_2, \dots, \delta_u) \in \mathbb{R}^{(r+1) \times u}$, the i th column δ_i of Δ is the $(r + 1)$ error rate of decline associated with the i th rule. Further definition is

$$\eta_i = \sqrt{\frac{\delta_i^T \delta_i}{r + 1}} \tag{38}$$

It can be seen that η_i reflects the importance of the i th rule: the larger the value of η_i , the more important the i th rule is. If $\eta_i < k_{err}$ then the i th rule can be eliminated. k_{err} is a pre-defined threshold value.

The flowchart of ANFM algorithm is shown in Figure 4.

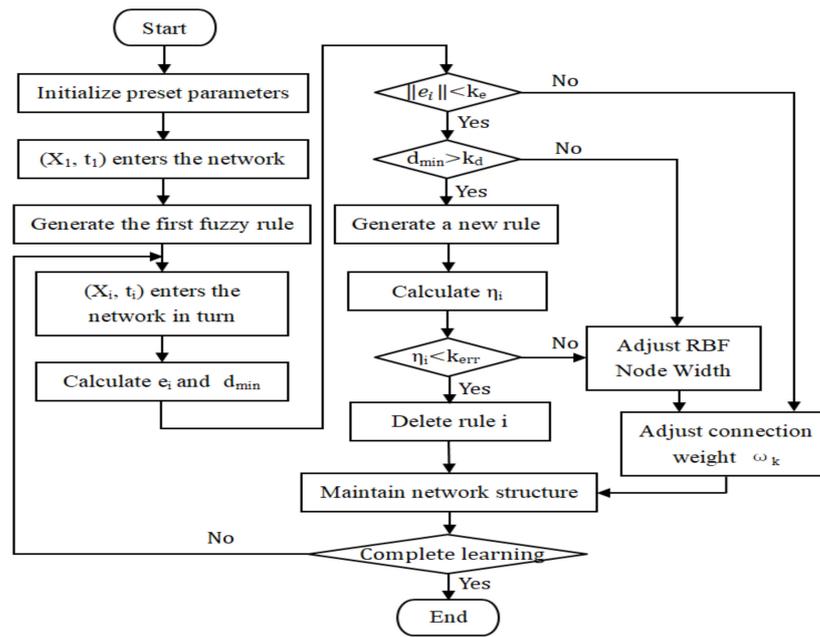


Figure 4. Flow of ANFM learning algorithm.

4. Inverse Model Identification of Ship Motion

According to Equation (5), the process of ANFM learning is shown in Figure 5. ANFM is used to learn the inverse model of ship course control with time-varying uncertainty in the modelling parameters, namely the dynamics of $\psi \rightarrow \delta$.

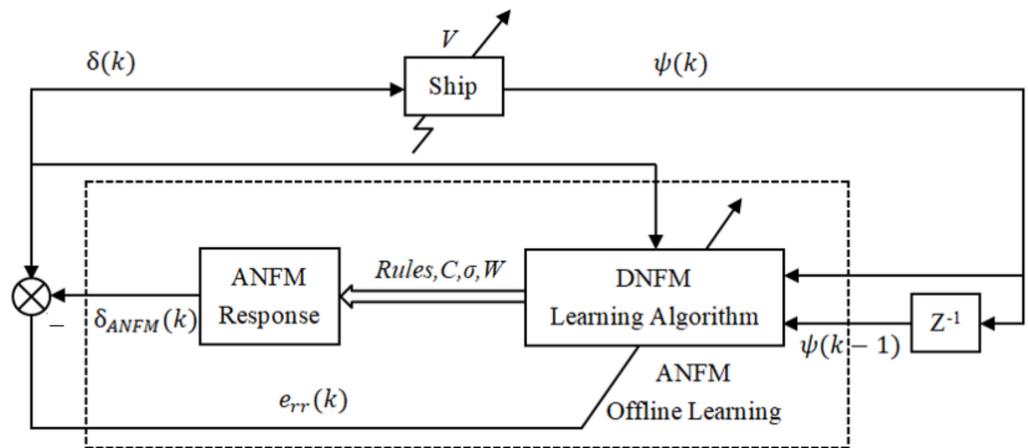


Figure 5. Progress of ANFM learning.

When learning, the system first generates suitable sample data (X_i, o_i) . Here X_i and o_i denote the input and output values of the ANFM corresponding to the i th sample data, respectively. For the generation of sample data, sine and sawtooth waves can be used in this paper. The ship module in the figure is set to model conditions where the parameters K, T, α, β are time-varying with speed V . Let the input data of the ship be $\delta(k)$, and each input rudder angle $\delta(k)$ produces an output course response $\psi(k)$ within a certain learning time N_k . $\psi(k)$ together with its previous step $\psi(k-1)$ and $\delta(k)$ are used as the input quantity of the ANFM learning algorithm module for determining the structure of ANFM, namely the number of fuzzy rules at each moment, and further from which the center C_j , the width σ_j and the weight matrix W of the output layer of each fuzzy rule, namely the RBF neural unit, are determined. The identification of ANFM as $\psi(k)$ and $\psi(k-1)$ as two input quantities is to expand the sample learning range of ANFM, thus effectively improving its

generalization ability. Once the learning algorithm has determined these parameters, the ANFM response module produces the corresponding network output values $\delta_{ANFM}(k)$. The difference between the input rudder angle of the ship $\delta(k)$ and the output rudder angle of ANFM $\delta_{ANFM}(k)$ is the discrimination error $err(k)$, which characterizes the learning efficiency and approximation performance of ANFM. ANFM is considered to be able to approximate the inverse dynamics of $\psi \rightarrow \delta$ for ship course control during this learning process when $err(k)$ enters within the set desired error range $[-err_d, err_d]$ and can be maintained until the end of learning. The trained ANFM can fix the network structure, that is, the number of RBF nodes in the third layer, and further use it in the online control system in the next stage.

The flowchart of the dynamic approximation inverse dynamics model is shown in Figure 6. After the training, the model can be obtained and the number of fuzzy rules and the initial weights, centers and widths are obtained for the following work.

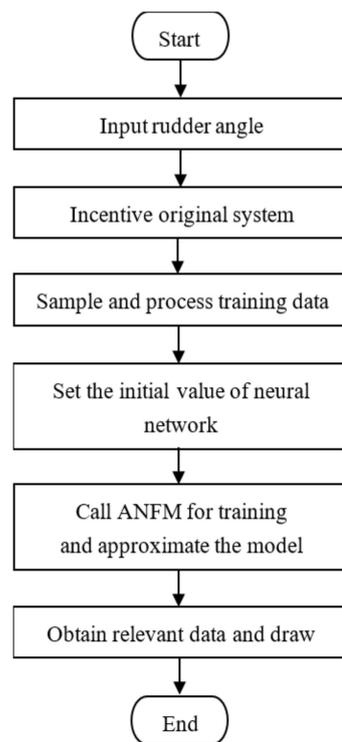


Figure 6. Flow of the approximation process.

5. Course Control Based on ANF-FOPID Controller

5.1. FO $PI^\lambda D^\mu$ Controller

There are four main types of representative fractional-order controllers: the TID controller [63], the CRONE controller [64], the fractional-order $PI^\lambda D^\mu$ controller and the lead lag compensator [65]. This paper adopts the fractional-order $PI^\lambda D^\mu$ controller proposed by Podlubny I. Compared with the traditional PID controller, the fractional-order $PI^\lambda D^\mu$ controller has two more parameters, namely the integral order λ and differential order μ make the control system more flexible and robust. The fractional-order $PI^\lambda D^\mu$ controller transfer function is

$$G_c(s) = K_p + \frac{K_i}{s^\lambda} + K_d s^\mu \tag{39}$$

$\lambda > 0, \mu > 0$ is the order of the controller, taking any real number.

In order to apply the theory of integer-order controllers to fractional-order controllers, the fractional-order calculus needs to be approximated to integer order. This paper uses a

modified Oustaloup approximation [66] to construct a continuous rational transfer function model for the fractional order calculus operator s^α , in the fitted frequency range (ω_b, ω_h) .

$$G(s) = K \left(\frac{ds^2 + b\omega_h s}{d(1-\alpha)s^2 + b\omega_h s + d\alpha} \right) \prod_{k=-N}^N \frac{1 + s/\omega'_k}{1 + s/\omega_k} \tag{40}$$

where $K = (\omega_b\omega_h)^\alpha$, ω_k and ω'_k are given by

$$\omega'_k = \left(\frac{b}{d} \right)^{\frac{2k-\alpha}{2N+1}} \omega_h^{\frac{N+k+\frac{1}{2}(1-\alpha)}{2N+1}} \omega_b^{\frac{N-k+\frac{1}{2}(1+\alpha)}{2N+1}} \tag{41}$$

$$\omega_k = \left(\frac{b}{d} \right)^{\frac{2k+\alpha}{2N+1}} \omega_h^{\frac{N+k+\frac{1}{2}(1+\alpha)}{2N+1}} \omega_b^{\frac{N-k+\frac{1}{2}(1-\alpha)}{2N+1}} \tag{42}$$

5.2. Course Control Based on ANF-FOPID Controller

The ANFM trained in Section 4 is connected in parallel with the FO PI^λD^μ controller to construct ANF-FOPID controller for ship course control. The structure of ANF-FOPID controller is shown in Figure 7.

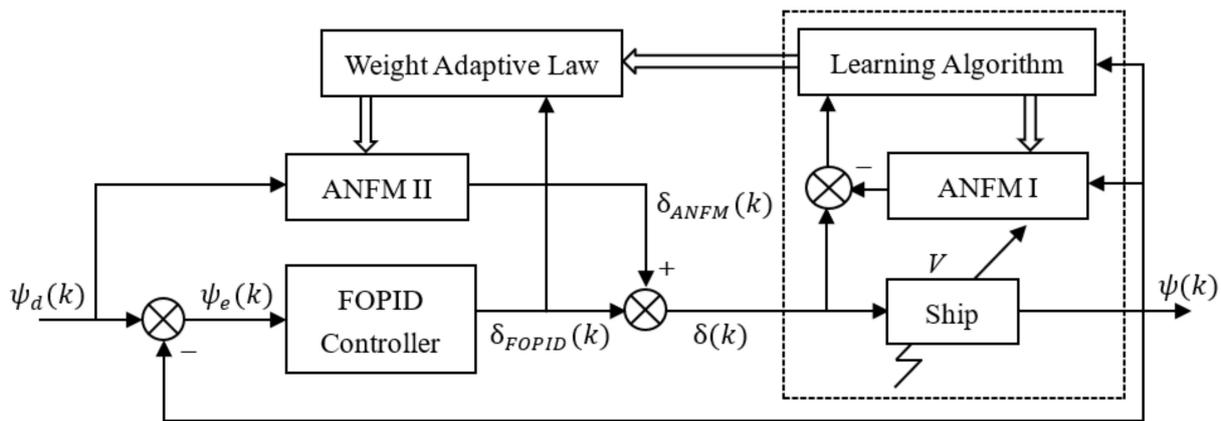


Figure 7. ANF-FOPID control system.

After learning the inverse dynamics model of the ship course, the ANFM I with trained structure and parameters is obtained and copied into ANFM II, as shown in the dashed box in the Figure 7, and connected in parallel with the FO PI^λD^μ controller (ANF-FOPID controller) to achieve course control of ship.

In the control process, an adaptive law with the output of the FO PI^λD^μ controller as the independent variable is introduced to update the ANFM output layer weight matrix W online to compensate for modelling errors and to cope with the effects of time-varying uncertainty in the modelling parameters. ANF-FOPID controller generates the control input rudder angle δ as

$$\delta = \delta_{FOPID} + \delta_{ANFM} \tag{43}$$

In which, ANFM is used to generate the control compensation signal and is the master controller, the detailed expression of which is shown in Equation (22). The FO PI^λD^μ controller is introduced to achieve faster and more accurate tracking performance and to ensure reliable control.

The squared error term between the expected control quantity and the actual control quantity of ANFM is

$$E(k) = \frac{1}{2} [\delta(k) - \delta_{ANFM}(k)]^2 \tag{44}$$

Substituting Equation (44), that is

$$E(k) = \frac{1}{2} \delta_{FOPID}^2(k) \tag{45}$$

The adaptive law for designing the online adjustment weight matrix W using the gradient descent method is as follows,

$$W(k + 1) = W(k) - \eta \frac{\partial E(k)}{\partial W(k)} \tag{46}$$

where, $\eta > 0$ is the learning rate.

From Equations (22), (44) and (46), it follows that

$$\frac{\partial E(k)}{\partial W(k)} = \frac{\partial E(k)}{\partial \delta_{ANFM}(k)} \cdot \frac{\partial \delta_{ANFM}(k)}{\partial W(k)} \tag{47}$$

that is

$$\frac{\partial E(k)}{\partial W(k)} = -[\delta(k) - \delta_{ANFM}(k)] \cdot \Psi(k) \tag{48}$$

The adaptive law is as follows, which is expressed as a function of the output of the fractional $PI^\lambda D^\mu$ controller and the output matrix vector of the ANFM normalization layer:

$$W(k + 1) = W(k) + \eta \delta_{FOPID}(k) \cdot \Psi(k) \tag{49}$$

Therefore, the ANF-FOPID controller achieves online control of the course by adjusting the output layer weight matrix of the ANFM that meets the training requirements, in conjunction with the FO $PI^\lambda D^\mu$ controller.

6. Simulations

In this paper, the simulation is conducted with the real ship data of COSCO 5446 TEU, a large container ship. As the ship uncertainty parameters K, T, α, β vary with the speed V , the values of parameters K, T, α, β corresponding to different speeds between $V = 16.3$ knots and $V = 27.1$ knots (1 knot = 1.852 km/h) can be derived from the ship state space model and the maneuverability of the ship [67], as shown in Table 1.

Table 1. Uncertain model parameters under different ship speed V .

V (Konts)	K	T	α	β
27.1	0.2676	186.9556	10.4915	7.5343
26.5	0.2617	191.1885	10.7291	8.0578
24.5	0.2419	206.7958	11.6049	10.1966
22.5	0.2222	225.1776	12.6366	13.1644
19.8	0.1955	255.8837	14.3601	19.3172
17.8	0.1758	284.6346	15.9748	26.5858
16.3	0.1609	310.8280	17.4475	34.6171

To identify the inverse dynamics of the ship’s course according to Figure 5, the simulation time is taken to be $N_k = 600$ s, and the ship’s working condition is assumed to be a smooth acceleration state, namely the speed is kept at $V = 16.3$ knot during the initial 120 s of the simulation, and as the speed is constant, the modeling parameters at this time are taken to correspond to the constant values in Table 1. During the 121~480 s of the simulation, the ship gradually accelerates from $V = 16.3$ knot to $V = 27.1$ knot. The change in speed causes a change in the modelling parameters, and this section linearizes the trend according to the values of each parameter corresponding to the speed $V = 16.3$ knot and $V = 27.1$ knot in Table 1. And during the period 481~600 s of the simulation, the speed was

kept at $V = 27.1$ knot. The speed change curve is shown in Figure 8, and the uncertainty parameters K, T, α, β at different speeds are shown in Figure 9.

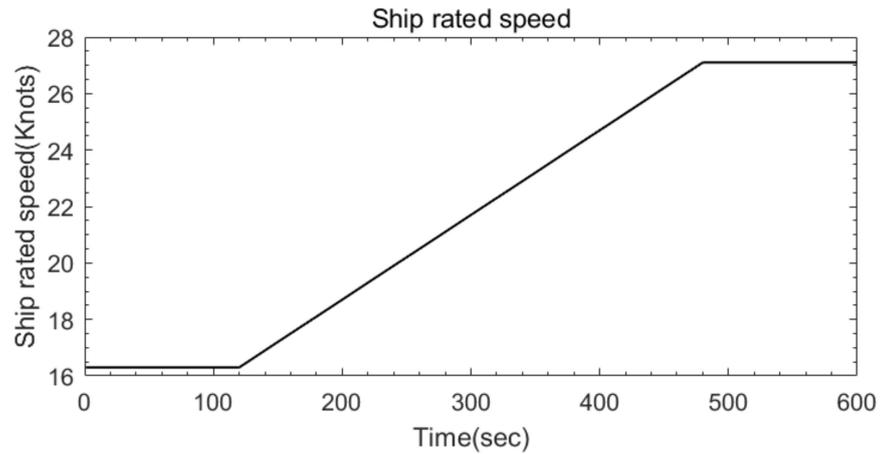


Figure 8. Changing curve of ship speed V .

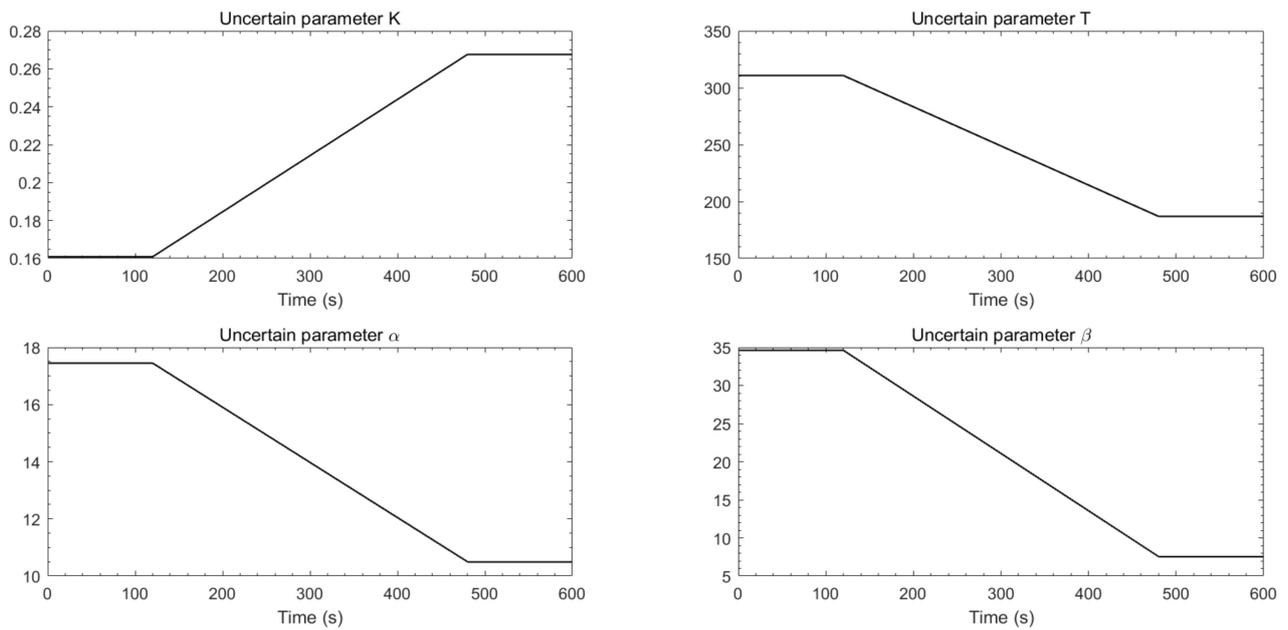


Figure 9. Changing curve of uncertain parameters $T, \alpha, \beta V$.

After the uncertainty condition is set, the input rudder angle signal is generated in Figure 5. As the rudder angle variation range is generally between $[-35^\circ, 35^\circ]$, in order to ensure the training data contain as wide a range of inputs as possible, the control rudder angle is first taken to be a sin wave curve as follows,

$$\delta(k) = 35\sin(2\pi k/600) \tag{50}$$

where $k \in [0, 600]$ is the training time, and let the simulation step $h = 1$ s. The output $\psi(k)$ generated by the ship, its previous step $\psi(k - 1)$ and $\delta(k)$ are used as sample data for the training of ANFM. That is

$$(X_i, o_i) = ((\psi(k), \psi(k - 1)), \delta(k)) \tag{51}$$

Take the pre-set parameters for ANFM as follows: $e_{max} = 5, e_{min} = 0.1, \beta = 0.9764, d_{max} = 251.2381, d_{min} = 0.2, \gamma = 0.9809, \sigma_0 = 106.1584, k_s = 2, k_\omega = 1.05, k_{err} = 0.0001$.

And the learning accuracy is set to $err_d = 0.01$. In doing so, ANFM continuously approximates the inverse dynamics of the ship under uncertainty.

The input sine wave curve is shown in Figure 10 and the simulation results are shown in Figures 11–14.

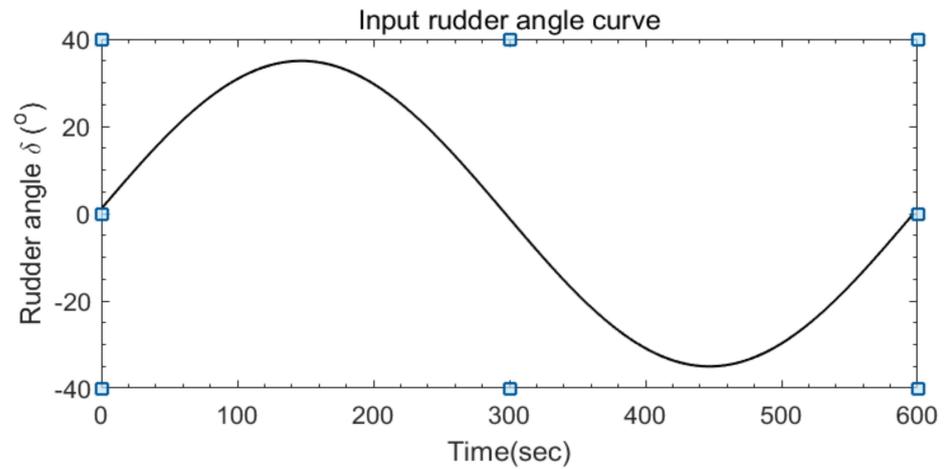


Figure 10. Input samples curve of sine wave.

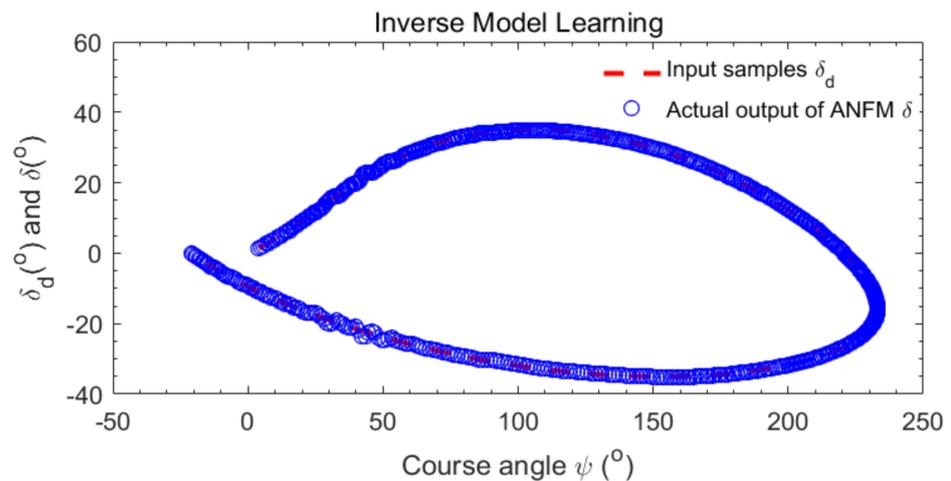


Figure 11. Comparison curve between input samples and ANFM actual outputs.

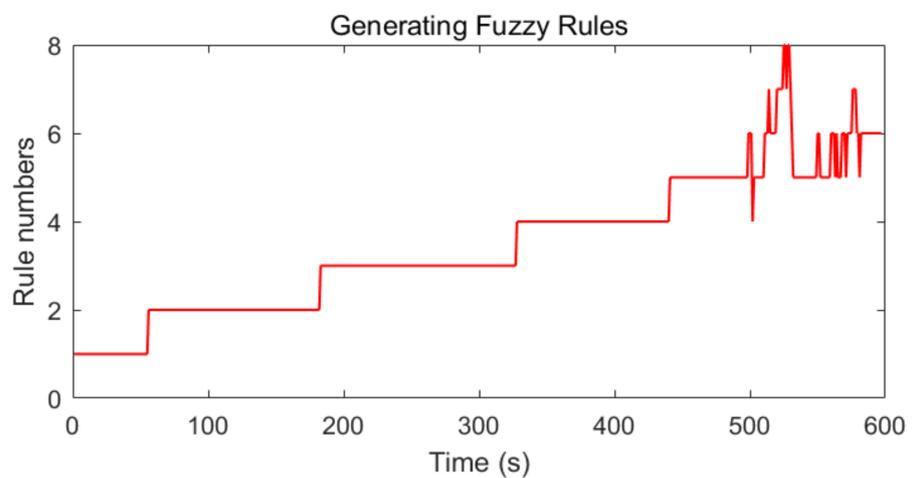


Figure 12. Generating fuzzy rules in training.

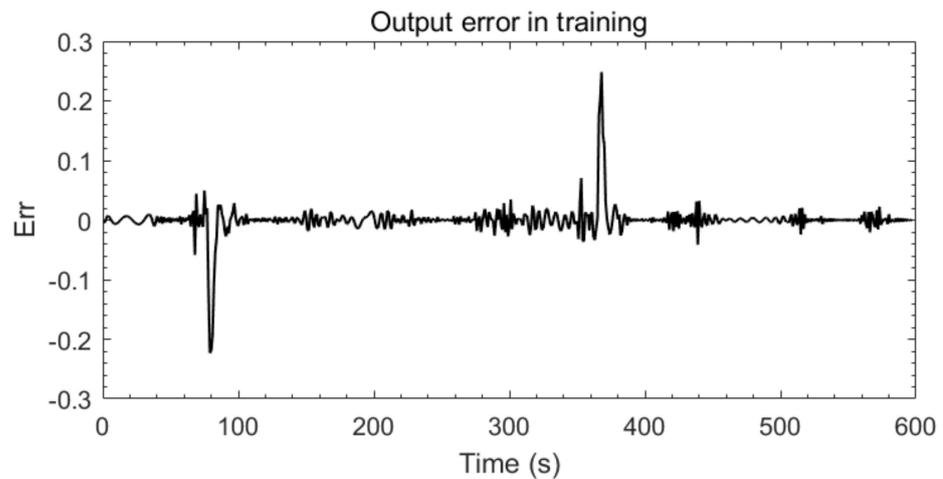


Figure 13. Output error in training.

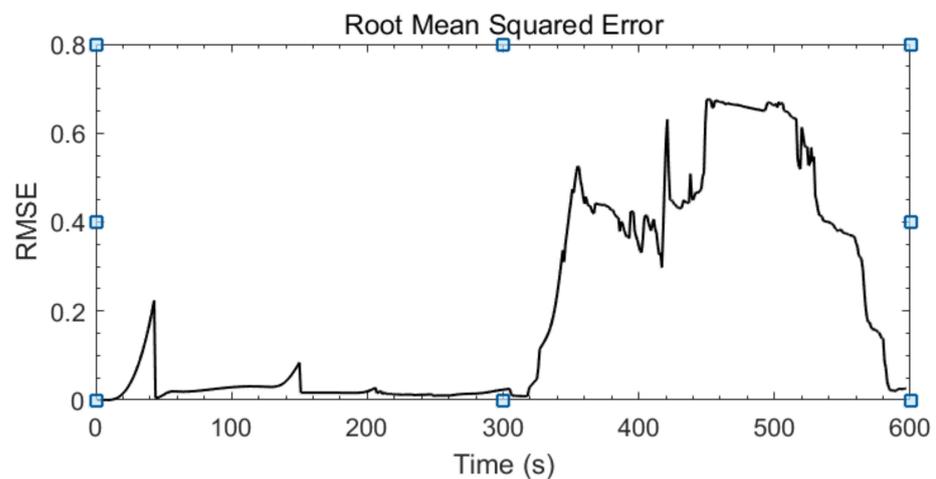


Figure 14. Root mean squared error in training.

The approximation between the input rudder angle and the output rudder angle of the ANFM is given in Figure 11. From Figure 11, the ANFM shows a strong approximation capability, with its output rudder angle almost always covering the ideal rudder angle. Figure 12 gives the changes in the structure of ANFM during the learning process. Rule numbers indicate the number of fuzzy rules in the third layer of ANFM or the number of RBF neural units in that layer. In Figure 12, the number of fuzzy rules gradually increases from 1 to 8, and then is pruned to 6 and saved until the end of the final learning, from which it can be argued that ANFM requires only six fuzzy rules, or only six RBF neural units to describe the inverse dynamics of $\psi \rightarrow \delta$ when the learning is complete. This result largely reflects the simplicity of the ANFM structure. From Figures 13 and 14, the errors are always kept within a reasonable error margin, indicating that ANFM can adequately learn the $\psi \rightarrow \delta$ inverse dynamics of ship course control. Through offline learning, the initial values of the center C , width σ and weight matrix W of each ANFM rule at this point are the corresponding values at the end of offline learning, respectively, as follows

$$C = \begin{bmatrix} 3.9551 & 43.3226 & 138.7384 & 321.2632 & 304.0374 & 294.7533 \\ 2.9910 & 42.7975 & 137.8617 & 322.2676 & 305.1702 & 295.9340 \end{bmatrix} \quad (52)$$

$$\sigma = [405.39 \quad 111.97 \quad 269.37 \quad 38.477 \quad 14.531 \quad 26.192] \quad (53)$$

$$W = \begin{bmatrix} -562.799 & 89.394 & -156.248 & 1154.4 & 46.487 & 48.629 \\ 86.734 & -35.134 & -75.125 & 20.336 & 20.771 & 21.047 \\ -85.027 & 35.127 & -86.289 & 71.988 & -20.496 & -20.938 \end{bmatrix} \quad (54)$$

To further verify the effectiveness of the ANFM structure, the paper then takes the input rudder angle as a sawtooth wave curve, with the rudder angle varying between $[-35^\circ, 35^\circ]$.

$$\delta(k) = \begin{cases} 7k/24, & 0 \leq k < 120 \\ -7k/24 + 70, & 120 \leq k < 360 \\ 7k/24 + 140, & 360 \leq k \leq 600 \end{cases} \quad (55)$$

The preset parameters of ANFM are kept constant and the input sawtooth wave curve is shown in Figure 15 and the simulation results are shown in Figures 16–19.

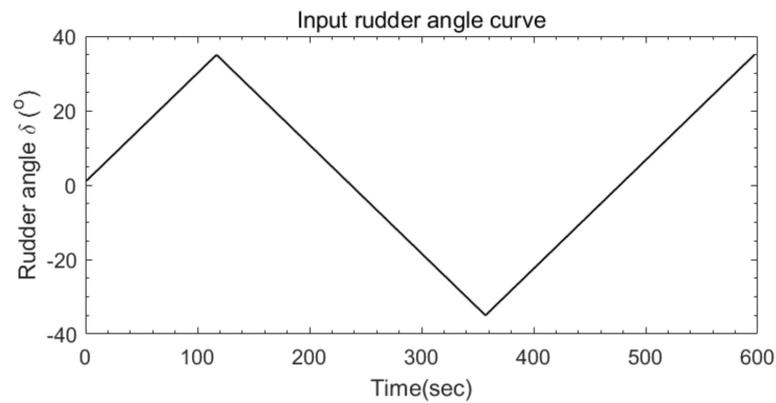


Figure 15. Input samples curve of sawtooth wave.

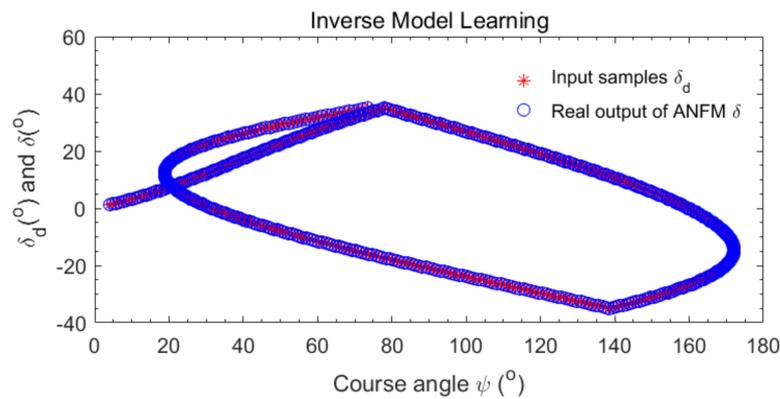


Figure 16. Comparison curve between input samples and ANFM actual outputs.

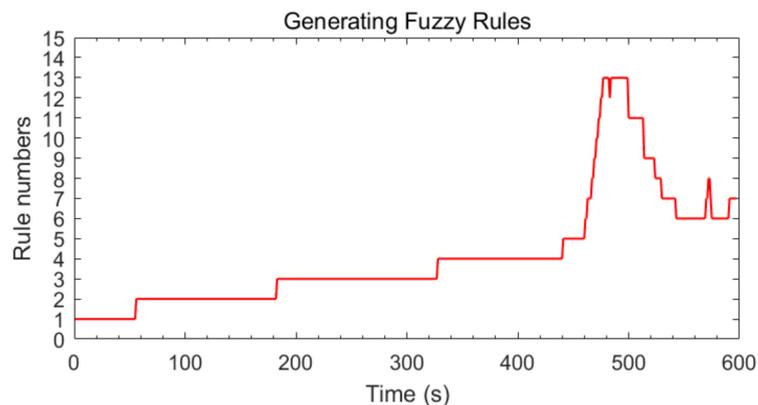


Figure 17. Generating fuzzy rules in training.

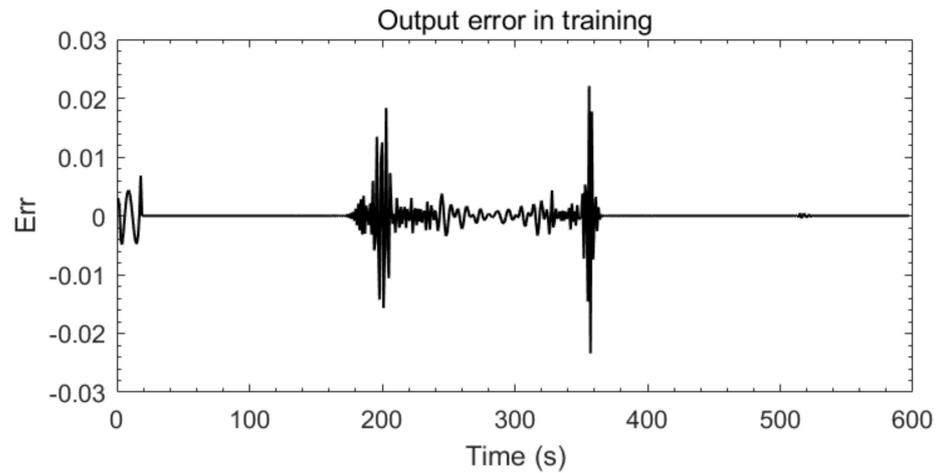


Figure 18. Output error in training.

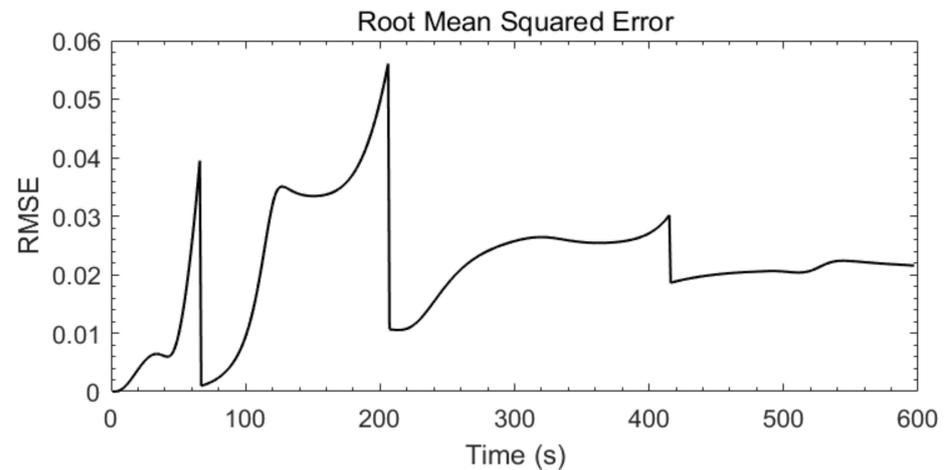


Figure 19. Root-mean-squared error in training.

As can be seen from the figure, the number of fuzzy rules gradually increases from 1 to 13 and then is pruned to 7 and saved until the end of the final learning, from which it can be argued that ANFM requires only seven fuzzy rules, or only seven RBF neural units, to describe the inverse dynamics of $\psi \rightarrow \delta$ after the learning is complete. Through offline learning, the initial values of the center C , width σ and weight matrix W of each ANFM rule at this point are the corresponding values at the end of offline learning, respectively, as follows

$$C = \begin{bmatrix} 3.9551 & 40.4649 & 135.5329 & 231.7632 & 304.0374 & 78.9814 & 84.1322 \\ 2.9910 & 39.8792 & 134.6732 & 231.5733 & 305.1702 & 80.3030 & 85.4312 \end{bmatrix} \quad (56)$$

$$\sigma = [205.68 \quad 103.80 \quad 268.51 \quad 273.13 \quad 111.67 \quad 20.016 \quad 13.031] \quad (57)$$

$$W = \begin{bmatrix} -584.013 & 87.696 & -215.734 & 1280.1 & 37.295 & 35.594 & 43.125 \\ 90.3475 & -34.491 & 85.637 & -79.283 & 21.341 & 21.262 & 25.173 \\ -88.312 & 34.463 & -85.267 & 75.568 & -21.474 & -21.387 & -25.323 \end{bmatrix} \quad (58)$$

The parameters of the fractional-order $PI^\lambda D^\mu$ controller are taken to be: $K_p = 0.083$, $K_i = 0.02$, $K_d = 1.23$, $\lambda = 0.012$, $\mu = 0.93$, learning rate $\eta = 0.04$, and the fractional-order calculus operator fitting frequency range (ω_b, ω_h) is set to $(0.001, 1000)$, while the weight matrix W of ANFM is adjusted online by Equation (49). The initial values of the different the center C , width σ and weight matrix W obtained from the two different input learning

samples mentioned above are used as structural parameters for ANFM respectively. Set the desired course angle to be tracked to

$$\psi_d(k) = \begin{cases} 30^\circ, & 0 \leq k < 120 \\ 45^\circ, & 120 \leq k < 360 \\ 60^\circ, & 360 \leq k \leq 600 \end{cases} \quad (59)$$

The initial course angle is taken as 0° and the left rudder is positive. It can be seen from Equation (59) that, while the modelling parameters are time-varying, the desired course angle is not perfectly synchronized with the speed and modeling parameters and is also time-varying in segments. The setting of this variation rule is to appropriately increase the difficulty of the control problem and to verify the adaptation of the ANF-FOPID controller to uncertainty.

The course tracking control curves with different learning samples are shown in Figures 20 and 21 and the control effects are shown in Tables 2 and 3.

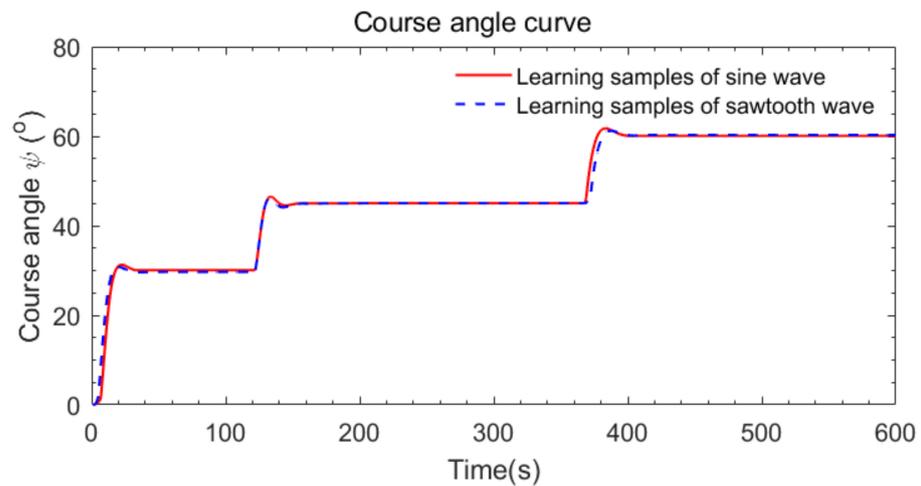


Figure 20. Course-tracking curves in two different learning samples.

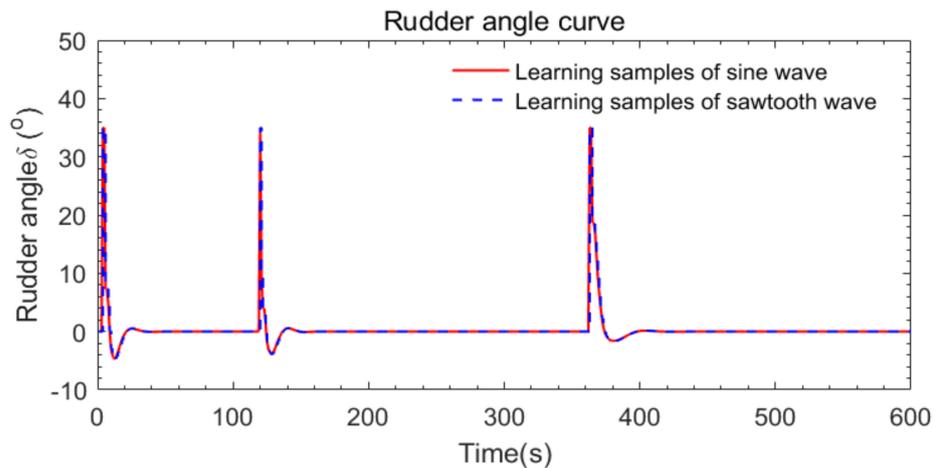


Figure 21. Rudder curves in two different learning samples.

Table 2. Performance in learning sample of Sine wave.

Sine Wave	Adjust Time t_s	Rise Time t_r	Overshoot M	E_{ss}
30°	32	18	4.27%	0.09
45°	52	16	3.31%	0.02
60°	43	18	2.93%	0.08

Table 3. Performance in learning sample of sawtooth wave.

Sawtooth Wave	Adjust Time t_s	Rise Time t_r	Overshoot M	E_{ss}
30°	57	17	2.64%	0.31
45°	55	16	2.49%	0.018
60°	51	23	2.02%	0.051

It can be seen from Figures 20 and 21 and Tables 2 and 3, the ANF-FOPID controller can drive the ship to respond quickly to the desired course angle by first generating a maximum control rudder angle and then gradually reducing it and enabling the ship to accurately track the desired course ψ_d . When the desired course angle setting is changed, the control effect has some oscillation, but the actual course of the ship can achieve fast and dynamic tracking with little overshoot. It is shown that the ANF-FOPID controller based on the ANFM offline learning mode can overcome the influence of uncertainty conditions arising from the variation of modeling parameters with speed and achieve satisfactory control results.

In order to fully validate the effectiveness of the ANF-FOPID controller, four different control strategies are used for ship course, which are the PID controller [57], the PID controller based on PSO algorithm [58], the FO PI^λD^μ controller based on GA algorithm [59] and the FO PI^λD^μ controller based on ANFM (proposed). The ship’s speed increases from $V = 19.8$ knots to $V = 24.5$ knots at a constant speed, the maximum rudder angle restricts to -35° to $+35^\circ$, and desired course angle is 30° when time is 0–600 s. The comparison results can be seen in Figures 22 and 23 and Table 4.

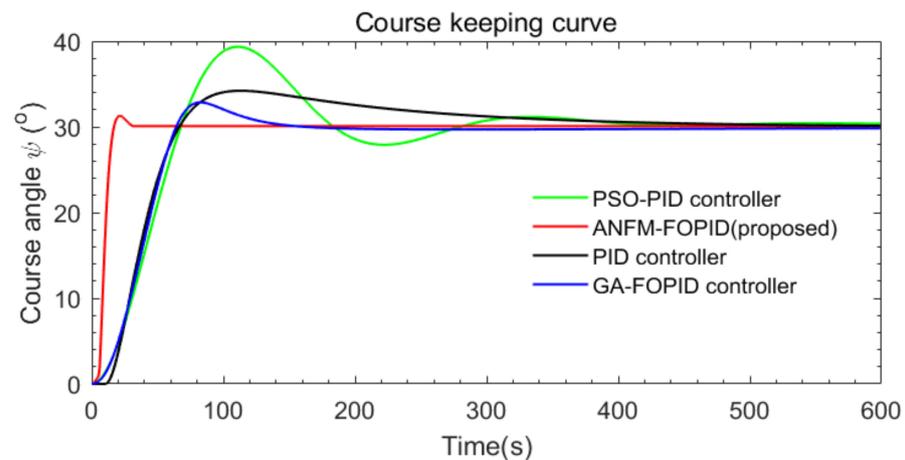


Figure 22. Comparison of course keeping using four different controllers.

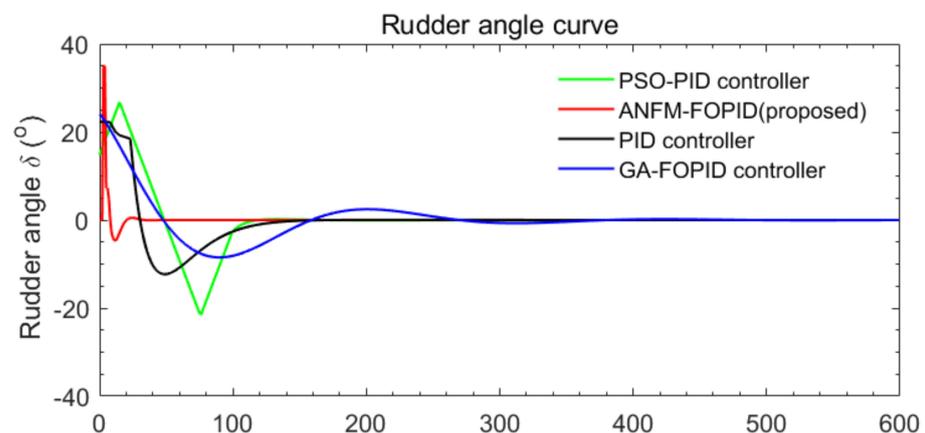


Figure 23. Comparison of rudder angle using four different controllers.

Table 4. Performance of four different controllers.

Controller Type	Adjust Time t_s	Rise Time t_r	Overshoot M	E_{ss}
ANFM -FOPID (proposed)	32	18	4.27%	0.009
PSO-PID	416	67	31.16%	0.024
PID	519	71	14.25%	0.007
GA-FOPID	163	65	9.53%	0.086

The simulation results show that the FO $PI^\lambda D^\mu$ controller based on ANFM (proposed) has significantly better control effect than the automatic rudder PID controller, PID controller based on PSO algorithm and FO $PI^\lambda D^\mu$ controller based on GA algorithm. The FO $PI^\lambda D^\mu$ controller based on ANFM has the advantages of small overshoot, short adjustment time, and accurate control.

7. Conclusions

This paper has discussed the construction of a novel identification method based on ANFM for approximating the inverse dynamics of the ship's course motion. Firstly, the simulation experiments of ship inverse dynamics have been carried out and obtained two groups of different ANFM structures by using sine wave curve and sawtooth wave curve as input learning samples of ANFM, respectively. The results have shown that the ANFM had a strong approximation capability, with its output rudder angle almost always covering the ideal rudder angle. Secondly, the trained ANFM has been used as inverse controller, and the fractional-order $PI^\lambda D^\mu$ controller has been connected in parallel for course tracking control of underactuated ships. The simulation results have shown that the proposed method can effectively overcome the influence of uncertainty of ship modeling parameters, track the desired course quickly and effectively, and had a good control effect. Finally, in ship course control, the effectiveness has been further evaluated by comparative experiments of four different controllers, which are the PID controller, the PID controller based on PSO algorithm, the FO $PI^\lambda D^\mu$ controller based on GA algorithm and the FO $PI^\lambda D^\mu$ controller based on ANFM (proposed). In terms of future works, because the system needs to set the trend of uncertain parameters artificially, and the control conditions and training conditions should be the same, but tend to be idealized, the online control mode after offline learning shows certain limitations. This method does not guarantee good control when the uncertainties no longer vary according to the established working conditions. To solve this problem, the next step is to change to an online control model, where modeling and control are carried out simultaneously to achieve full adaptivity of the intelligent controller in order to improve the generalization capability and control efficiency of ANFM and to cope with the effects of non-normative uncertainties.

Author Contributions: Conceptualization, G.L.; methodology, G.L.; software, H.C.; validation, G.L. and B.C.; data curation, G.L.; writing—original draft preparation, G.L.; writing—review and editing, W.D.; project administration, B.C.; funding acquisition, G.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Science Researching Plans of Liaoning Province Education Department under Grant JDL2019003, the National Natural Science Foundation of China under Grant 61803066 and U2133205, and Research Foundation for Civil Aviation University of China (3122022PT02).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Dai, S.L.; He, S.; Wang, M.; Yuan, C. Adaptive neural control of underactuated surface vessels with prescribed performance guarantees. *IEEE transactions on neural networks and learning systems*. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *30*, 3686–3698. [[CrossRef](#)] [[PubMed](#)]
2. Chen, L.; Cui, R.; Yang, C.; Yan, W. Adaptive Neural Network Control of Underactuated Surface Vessels with Guaranteed Transient Performance: Theory and Experimental Results. *IEEE Trans. Ind. Electron.* **2019**, *67*, 4024–4035. [[CrossRef](#)]
3. Jia, Z.; Hu, Z.; Zhang, W. Adaptive output-feedback control with prescribed performance for trajectory tracking of underactuated surface vessels. *ISA Trans.* **2019**, *95*, 18–26. [[CrossRef](#)] [[PubMed](#)]
4. Lin, X.; Nie, J.; Jiao, Y.; Liang, K.; Li, H. Adaptive fuzzy output feedback stabilization control for the underactuated surface vessel. *Appl. Ocean Res.* **2018**, *74*, 40–48. [[CrossRef](#)]
5. Ghommam, J.; Saad, M.; Mnif, F. Prescribed performances based fuzzy-adaptive output feedback containment control for multiple underactuated surface vessels. *Ocean Eng.* **2022**, *249*, 110898. [[CrossRef](#)]
6. Li, G.; Li, Y.; Chen, H.; Deng, W. Fractional-Order Controller for Course-Keeping of Underactuated Surface Vessels Based on Frequency Domain Specification and Improved Particle Swarm Optimization Algorithm. *Appl. Sci.* **2022**, *12*, 3139. [[CrossRef](#)]
7. Pettersen, K.Y.; Egeland, O. Exponential stabilization of an underactuated surface vessel. In Proceedings of the 35th IEEE Conference on Decision and Control, Kobe, Japan, 13 December 1996; Volume 1, pp. 967–972.
8. Zhang, Y.; Li, S.; Liu, X. Adaptive Near-Optimal Control of Uncertain Systems with Application to Underactuated Surface Vessels. *IEEE Trans. Control Syst. Technol.* **2017**, *26*, 1204–1218. [[CrossRef](#)]
9. Jin, T.; Xia, H.; Gao, S. Reliability analysis of the uncertain fractional-order dynamic system with state constraint. *Math. Methods Appl. Sci.* **2021**, *45*, 2615–2637. [[CrossRef](#)]
10. Deng, W.; Xu, J.; Gao, X.-Z.; Zhao, H. An enhanced MSIQDE algorithm with novel multiple strategies for global optimization problems. *IEEE Trans. Syst. Man Cybern. Syst.* **2022**, *52*, 1578–1587. [[CrossRef](#)]
11. Zhang, Z.-H.; Min, F.; Chen, G.-S.; Shen, S.-P.; Wen, Z.-C.; Zhou, X.-B. Tri-Partition State Alphabet-Based Sequential Pattern for Multivariate Time Series. *Cogn. Comput.* **2021**, 1–19. [[CrossRef](#)]
12. Wu, X.; Wang, Z.C.; Wu, T.H.; Bao, X.G. Solving the Family Traveling Salesperson Problem in the Adleman–Lipton Model Based on DNA Computing. *IEEE Trans. NanoBioscience* **2021**, *21*, 75–85. [[CrossRef](#)] [[PubMed](#)]
13. Chen, H.; Zhang, Q.; Luo, J.; Xu, Y.; Zhang, X. An enhanced Bacterial Foraging Optimization and its application for training kernel extreme learning machine. *Appl. Soft Comput.* **2019**, *86*, 105884. [[CrossRef](#)]
14. Cui, H.; Guan, Y.; Chen, H. Rolling Element Fault Diagnosis Based on VMD and Sensitivity MCKD. *IEEE Access* **2021**, *9*, 120297–120308. [[CrossRef](#)]
15. Li, T.; Shi, J.; Deng, W.; Hu, Z. Pyramid particle swarm optimization with novel strategies of competition and cooperation. *Appl. Soft Comput.* **2022**, *121*, 108731. [[CrossRef](#)]
16. Park, B.S.; Yoo, S.J. Adaptive-observer-based formation tracking of networked uncertain underactuated surface vessels with connectivity preservation and collision avoidance. *J. Frankl. Inst.* **2019**, *356*, 7947–7966. [[CrossRef](#)]
17. Guan, W.; Zhou, H.; Su, Z.; Zhang, X.; Zhao, C. Ship Steering Control Based on Quantum Neural Network. *Complexity* **2019**, *2019*, 3821048. [[CrossRef](#)]
18. Pathan, D.M.; Hussain, T.; Daudpoto, J.; Memon, I.A. Neural network controller for a ship. *Sindh. Univ. Res. J.* **2012**, *44*, 399–404.
19. Khizer, A.N.; Yaping, D.; Unar, M.A. Design of Heading Controller for Cargo Ship using Feed Forward Artificial Neural Network. *Int. J. Adv. Comput. Technol.* **2013**, *5*, 556–566. [[CrossRef](#)]
20. Pathan, D.M.; Abbassi, A.F.; Memon, Z.A. Neural network course changing and track keeping controller for a submarine. *Mehran. Univ. Res. J. Eng. Technol.* **2012**, *31*, 711–722.
21. Duan, H.Q.; Sun, H.F. Adaptive backstepping neural network algorithm of ship line-course control. *J. Nanjing Univ. Sci. Technol.* **2012**, *36*, 427–431.
22. Xia, G.; Wu, H.; Shao, X. Adaptive Filtering Backstepping for Ships Steering Control without Velocity Measurements and with Input Constraints. *Math. Probl. Eng.* **2014**, *2014*, 218585. [[CrossRef](#)]
23. Shenreng, J.; Liu, L. Adaptive neural network control for ship steering system using filtered backstepping design. *J. Appl. Sci.* **2013**, *13*, 1691–1697.
24. Wang, R.; Li, D.; Miao, K. Optimized Radial Basis Function Neural Network Based Intelligent Control Algorithm of Unmanned Surface Vehicles. *J. Mar. Sci. Eng.* **2020**, *8*, 210. [[CrossRef](#)]
25. Wanga, R.; Deng, H.; Miao, K.; Zhao, Y.; Du, J. Backstepping based integral sliding mode control with neural network for ship steering control. *MATEC Web Conf.* **2017**, *139*, 141. [[CrossRef](#)]
26. Velagic, J. Design of Ship Controller and Ship Model Based on Neural Network Identification Structures. In Proceedings of the 2006 World Automation Congress, Budapest, Hungary, 24–26 July 2006. [[CrossRef](#)]
27. Al-Hussein, A.B. Combined neural network and PD adaptive tracking controller for ship steering system. *Iraq J. Electr. Electron. Eng.* **2017**, *13*, 59–66. [[CrossRef](#)]
28. Wang, W.; Liu, C. An efficient ship autopilot design using observer-based model predictive control. *Proc. Inst. Mech. Eng. Part M J. Eng. Marit. Environ.* **2020**, *235*, 203–212. [[CrossRef](#)]
29. Shao, H.; Lin, J.; Zhang, L.; Galar, D.; Kumar, U. A novel approach of multisensory fusion to collaborative fault diagnosis in maintenance. *Inf. Fusion* **2021**, *74*, 65–76. [[CrossRef](#)]

30. Zhang, X.; Wang, H.; Du, C.; Fan, X.; Cui, L.; Chen, H.; Deng, F.; Tong, Q.; He, M.; Yang, M.; et al. Custom-Molded Offloading Footwear Effectively Prevents Recurrence and Amputation, and Lowers Mortality Rates in High-Risk Diabetic Foot Patients: A Multicenter, Prospective Observational Study. *Diabetes Metab. Syndr. Obes. Targets Ther.* **2022**, *15*, 103–109. [[CrossRef](#)]
31. He, Z.; Shao, H.; Zhong, X.; Zhao, X. Ensemble transfer CNNs driven by multi-channel signals for fault diagnosis of rotating machinery cross working conditions. *Knowl.-Based Syst.* **2020**, *207*, 106396. [[CrossRef](#)]
32. Wei, Y.; Zhou, Y.; Luo, Q.; Deng, W. Optimal reactive power dispatch using an improved slime mould algorithm. *Energy Rep.* **2021**, *7*, 8742–8759. [[CrossRef](#)]
33. Deng, W.; Zhang, X.; Zhou, Y.; Liu, Y.; Zhou, X.; Chen, H.; Zhao, H. An enhanced fast non-dominated solution sorting genetic algorithm for multi-objective problems. *Inf. Sci.* **2021**, *585*, 441–453. [[CrossRef](#)]
34. Wu, J.; Wang, Z. A Hybrid Model for Water Quality Prediction Based on an Artificial Neural Network, Wavelet Transform, and Long Short-Term Memory. *Water* **2022**, *14*, 610. [[CrossRef](#)]
35. Jin, T.; Xia, H.; Deng, W.; Li, Y.; Chen, H. Uncertain Fractional-Order Multi-Objective Optimization Based on Reliability Analysis and Application to Fractional-Order Circuit with Caputo Type. *Circuits Syst. Signal Process.* **2021**, *40*, 5955–5982. [[CrossRef](#)]
36. Lin, C.T.; Lee, C.S.G. *Neural Fuzzy Systems—A Neuro-Fuzzy Synergism to Intelligent Systems*; Prentice-Hall: Hoboken, NJ, USA, 1996.
37. Wang, L.X.; Mendel, J.M. Fuzzy Basis Functions, Universal Approximation and Orthogonal Least-Squares Learning. *IEEE Trans. Neural Netw.* **1992**, *3*, 807–814. [[CrossRef](#)]
38. Jang, J.S.R. ANFIS: Adaptive-Network-based Fuzzy Inference Systems. *IEEE Trans. Syst. Man Cybern.* **1993**, *23*, 665–685. [[CrossRef](#)]
39. Tung, L.T. Design a ship autopilot using neural network. *J. Ship Prod. Design* **2017**, *33*, 192–196. [[CrossRef](#)]
40. 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](#)]
41. Zhong, C.; Jiang, Z.; Chu, X.; Liu, L. Inland Ship Trajectory Restoration by Recurrent Neural Network. *J. Navig.* **2019**, *72*, 1359–1377. [[CrossRef](#)]
42. Volkova, T.A.; Balykina, Y.E.; Bespalov, A. Predicting ship trajectory based on neural networks using AIS data. *J. Marine Sci. Eng.* **2021**, *9*, 254. [[CrossRef](#)]
43. Jiang, Y.; Hou, X.R.; Wang, X.G.; Wang, Z.H.; Yang, Z.L.; Zou, Z.J. Identification modeling and prediction of ship maneuvering motion based on LSTM deep neural network. *J. Mar. Sci. Technol.* **2022**, *27*, 125–137. [[CrossRef](#)]
44. Le, T.T. Ship heading control system using neural network. *J. Mar. Sci. Technol.* **2021**, *26*, 963–972. [[CrossRef](#)]
45. Deng, W.; Li, Z.; Li, X.; Chen, H.; Zhao, H. Compound Fault Diagnosis Using Optimized MCKD and Sparse Representation for Rolling Bearings. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 3508509. [[CrossRef](#)]
46. Jin, T.; Xia, H. Lookback option pricing models based on the uncertain fractional-order differential equation with Caputo type. *J. Ambient Intell. Humaniz. Comput.* **2021**, 1–14. [[CrossRef](#)]
47. Zhang, C.; Wan, L.; Liu, Y. Ship heading control based on fuzzy PID control. In Proceedings of the IEEE 2019 34rd Youth Academic Annual Conference of Chinese Association of Automation (YAC), Jinzhou, China, 6–8 June 2019; pp. 607–612.
48. Nguyen, V.S.; Im, N.K. Automatic ship berthing based on fuzzy logic. *Int. J. Fuzzy Log. Intell. Syst.* **2019**, *19*, 163–171. [[CrossRef](#)]
49. Zhu, L.; Li, T. Observer-based adaptive fuzzy prescribed performance control for intelligent ship autopilot. *Syst. Sci. Control Eng.* **2021**, *9*, 489–496. [[CrossRef](#)]
50. Min, B.; Zhang, X. Concise robust fuzzy nonlinear feedback track keeping control for ships using multi-technique improved LOS guidance. *Ocean Eng.* **2021**, *224*, 108734. [[CrossRef](#)]
51. Oldham, K.B.; Spanier, J. *The Fractional Calculus*; Academic Press: New York, NY, USA, 1974.
52. Podlubny, I. Fractional-order System and Fractional-order Control. *IEEE Trans. Autom. Control* **1999**, *44*, 208–214. [[CrossRef](#)]
53. Shah, P.; Agashe, S. Review of fractional PID controller. *Mechatronics* **2016**, *38*, 29–41. [[CrossRef](#)]
54. Dabiri, A.; Moghaddam, B.P.; Machado, J.A.T. Optimal variable-order fractional PID controllers for dynamical systems. *J. Comput. Appl. Math.* **2018**, *339*, 40–48. [[CrossRef](#)]
55. Izci, D.; Ekinci, S.; Eker, E.; Kayri, M. A Novel Modified Opposition-based Hunger Games Search Algorithm to Design FOPID Controller for Magnetic Ball Suspension System. *Adv. Control. Appl. Eng. Ind. Syst.* **2022**, *4*, e96.
56. Izci, D.; Ekinci, S.; Hekimoğlu, B. Fractional-Order PID Controller Design for Buck Converter System via Hybrid Lévy Flight Distribution and Simulated Annealing Algorithm. *Arab. J. Sci. Eng.* **2022**, 1–19. [[CrossRef](#)]
57. Fossen, T.I. *Guidance and Control of Ocean Vehicle*; John Wiley & Sons Ltd.: London, UK, 1994.
58. Dlabáč, T.; Čalasan, M.; Krčum, M.; Marvučić, N. PSO-based PID controller design for ship course-keeping autopilot. *Brodogradnja* **2019**, *70*, 1–15. [[CrossRef](#)]
59. Bingül, Z.A.F.E.R.; Karahan, O.Ğ.U.Z.H.A.N. Fractional PID controllers tuned by evolutionary algorithms for robot trajectory control. *Turk J. Elec. Eng. Comp. Sci.* **2012**, *20*, 1123–1136.
60. Lee, S.; Kil, R.M. A Gaussian potential function network with hierarchically self-organizing learning. *Neural Netw.* **1991**, *4*, 207–224. [[CrossRef](#)]
61. Ho, S.J.; Shu, L.S.; Ho, S.Y. Optimizing fuzzy neural networks for tuning PID controllers using an orthogonal simulated annealing algorithm OSA. *IEEE Trans. Fuzzy Syst.* **2006**, *14*, 421–434. [[CrossRef](#)]
62. Lune, B.J. Three-Parameter Tunable Tilt-Integral Derivative (TID) Controller. US Patent US5371670, 6 December 1994.

63. Oustaloup, A.; Mathieu, B.; Lanusse, P. The CRONE Control of Resonant Plants: Application to a Flexible Transmission. *Eur. J. Control* **1995**, *1*, 113–121. [[CrossRef](#)]
64. Raynaud, H.F.; Zergamoh, A. State-space representation for fractional order controllers. *Automatica* **2000**, *36*, 1017–1021. [[CrossRef](#)]
65. Oustaloup, A.; Levron, F.; Mathieu, B.; Nanot, F.M. Frequency-band complex non-integer differentiator: Characterization and synthesis. *IEEE Trans. Circuits Syst.-I Fundam. Theory Appl.* **2000**, *47*, 25–39. [[CrossRef](#)]
66. Leontaritis, I.J.; Billings, S.A. Input-output parametric models for non-linear systems part I: Deterministic non-linear systems. *Int. J. Control.* **1985**, *41*, 303–328. [[CrossRef](#)]
67. Shen, Z. *Study on General Fuzzified CMAC Based Ship Motion Intelligent Control and its Distributed Simulation*; Dalian Maritime University: Dalian, China, 2005.