applied sciences

MDPI

Article

Fractional-Order Controller for Course-Keeping of Underactuated Surface Vessels Based on Frequency Domain Specification and Improved Particle Swarm Optimization Algorithm

Guangyu Li [1], Yanxin Li [1], Huayue Chen [2,*] and Wu Deng [3,*]

[1] School of Software, Dalian Jiaotong University, Dalian 116026, China; ligyu@djtu.edu.cn (G.L.); leeyx@djtu.edu.cn (Y.L.)
[2] School of Computer Science, China West Normal University, Nanchong 637002, China
[3] School of Electronic Information and Automation, Civil Aviation University of China, Tianjin 300300, China
* Correspondence: sunnyxiaoyue20@cwnu.edu.cn (H.C.); wdeng@cauc.edu.cn (W.D.)

Abstract: In this paper, a new fractional-order (FO) $PI^{\lambda}D^{\mu}$ controller is designed with the desired gain and phase margin for the automatic rudder of underactuated surface vessels (USVs). The integral order λ and the differential order μ are introduced in the controller, and the two additional adjustable factors make the FO $PI^{\lambda}D^{\mu}$ controller have better accuracy and robustness. Simulations are carried out for comparison with a ship's digital PID autopilot. The results show that the FO $PI^{\lambda}D^{\mu}$ controller has the advantages of a small overshoot, short adjustment time, and precise control. Due to the uncertainty of the model parameters of USVs and two extra parameters, it is difficult to compute the parameters of an FO $PI^{\lambda}D^{\mu}$ controller. Secondly, this paper proposes a novel particle swarm optimization (PSO) algorithm for dynamic adjustment of the FO $PI^{\lambda}D^{\mu}$ controller parameters. By dynamically changing the learning factor, the particles carefully search in their own neighborhoods at the early stage of the algorithm to prevent them from missing the global optimum and converging on the local optimum, while at the later stage of evolution, the particles converge on the global optimal solution quickly and accurately to speed up PSO convergence. Finally, comparative experiments of four different controllers under different sailing conditions are carried out, and the results show that the FO $PI^{\lambda}D^{\mu}$ controller based on the IPSO algorithm has the advantages of a small overshoot, short adjustment time, precise control, and strong anti-disturbance control.

Keywords: underactuated surface vessels; fractional order $PI^{\lambda}D^{\mu}$ controller; course-keeping; improved particle swarm optimization algorithm; autopilot

Citation: 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. https://doi.org/10.3390/app12063139

Academic Editor: Giancarlo Mauri

Received: 16 February 2022
Accepted: 16 March 2022
Published: 18 March 2022

1. Introduction

In recent years, the maneuverability of underactuated surface vessels (USVs) has become a hot topic of ship control [1–5]. An underactuated system is one in which the number of degrees of freedom of the system is greater than the number of control inputs. An underactuated system refers to a situation where the dimension of the system control input vector space is smaller than the dimension of the system's generalized coordinate vector space [6]. In the study of ship maneuverability, the effectiveness of ship course control has a direct impact on ship maneuverability and the safety of navigation at sea. For vessels at sea, to keep the ship on the expected course, the rudder must be controlled by a control system to correct for the yaw caused by disturbances. Digital PID rudders are widely used as course controllers for ships at sea, but they are too sensitive to disturbances such as wind and waves, which can lead to frequent steering of the rudder. A great deal of research has been carried out on automatic rudder steering for ships in order to achieve satisfactory maneuvering performance, and many results have been obtained [7–12]. In [7],

Le proposed to combine fuzzy logic control with linear control theory for the design of a ship autopilot controller and used the controller for ship course control, achieving good control results. In [8], optimal control based on the linear quadratic optimal was proposed and applied to a ship's autopilot. In [9], the PSO method for an optimal autopilot PID controller was designed. In [10], the $H\infty$ controller was solved by the "2-Riccati" equation, and it was used for the rudder and flap of a ship's course under system uncertainty. In [11], a new robust adaptive nonlinear feedback algorithm based on error-driven function was proposed, and it was used for ship course-keeping. In [12], an indirect adaptive robust controller was proposed. First, the parameter estimation law under system uncertainty was studied for a ship model. Secondly, an indirect adaptive robust controller was applied for a ship's heading.

With the continuous improvement of precision control theory, fractional order calculus theory [13–15] and fractional order $PI^{\lambda}D^{\mu}$ controllers [16–18] have been proposed one after another. In 1999, Podlubny proposed fractional order $PI^{\lambda}D^{\mu}$ controllers on the basis of fractional order calculus theory [16]. Compared with the traditional PID controller, the introduction of integral order λ and differential order μ makes the PID controller have two more control parameters, expanding the original "point" control to "surface" control, further improving the control accuracy and making the controller more flexible [19–27]. In [19], a two-degrees-of-freedom planar robot was controlled by an FO PID controller tuned with particle swarm optimization (PSO) and a genetic algorithm (GA). In [20], an FO $PI^{\lambda}D^{\mu}$ controller was applied to the speed control of subway trains, and optimization studies were carried out to make the train achieve better speed control and stability. In [21], a new FO $PI^{\lambda}D^{\mu}$ controller was proposed, and a structural transformation method was designed to obtain the controller parameters, which were used to improve the anti-interference of the controller. In [22], a special case of fractional-order systems was first investigated, and on this basis, the FOPD controller and the FOPID controller were designed, which were used in the control of fractional-order systems to achieve a closed-loop system with a monotonic amplitude-frequency response. In [23], for a three-link rigid manipulator system with coupled multiple-input multiple-output characteristics and high nonlinearity, a fractional-order self-tuned fuzzy PID controller was proposed, which was applied to a robotic manipulator system and achieved good control results. In [24], an adaptive fractional order PID controller was proposed, which was applied to a photovoltaic grid system. In [25], an FOPID controller was proposed in a magnetic ball suspension system, and a modified opposition-based learning technique and the hunger games search algorithm were used to tune the parameters of the FOPID controller. In [26], an FO $PI^{\lambda}D^{\mu}$ controller was applied to an automatic voltage regulator (AVR), and a self-regulated off-line optimal tuning method based on the gradient-based optimization algorithm was proposed to tune the five parameters of the FO $PI^{\lambda}D^{\mu}$ controller. Davut Izci proposed an FOPID controller for a buck converter system and hybrid Lévy flight distribution and simulated annealing algorithms (LFDSAs), which involved a balanced structure in terms of explorative and exploitative phases being used to tune the parameters of the FOPID controller [27]. In addition, some new algorithms were also proposed to optimize the PID [28–34].

In this paper, an FO $PI^{\lambda}D^{\mu}$ controller is applied to a ship's course control. A modified Oustaloup approximation method is adopted. Given the values of the gain margin A_m and phase margin ϕ_m, the parameters of the FO $PI^{\lambda}D^{\mu}$ controller are calculated. The effectiveness of the algorithm is demonstrated by simulations comparing the FO $PI^{\lambda}D^{\mu}$ controller with a PID controller. However, as the FO $PI^{\lambda}D^{\mu}$ controller is designed for ship course control at a rated speed, if the ship speed changes dynamically, the non-linear model of the ship will also change accordingly, which will result in the original controller parameters no longer meeting the needs of the ship's course control and needing to be recalculated. Therefore, this paper proposes an improved particle swarm algorithm (IPSO) for dynamic adjustment of the FO $PI^{\lambda}D^{\mu}$ controller parameters. By dynamically changing the learning factor, the particles carefully search in their own neighborhoods at the early stage of the algorithm to prevent them from skipping the global optimum and converging

on a local optimum, while at the later stage of evolution, the particles converge on the global optimal solution quickly and accurately to improve the convergence speed. Comparative experiments are carried out without wind and wave disturbances and under wind and wave disturbances. The results show that the FO PI$^{\lambda}$D$^{\mu}$ controller based on the IPSO algorithm has the advantages of a small overshoot, short adjustment time, precise control, and strong anti-disturbance control.

2. USV Model

2.1. Ship Motion Model

The movement of a ship has six degrees of freedom (DOF), which represent the position and direction of the ship's movement, and therefore six separate coordinates are required to describe these six degrees of freedom. (x, y, z) represents the position of a ship in the inertial frame, and the first-order differentiation of (x, y, z) represents the translational motion along x, y, and z. The first three DOF are surge, sway, and heave. (φ, θ, ψ) represents the direction of a ship, and the first-order differentiation of (φ, θ, ψ) represents the rotation of the ship. The last three DOF are roll, pitch, and yaw [35]. The ship motion description is shown in Figure 1.

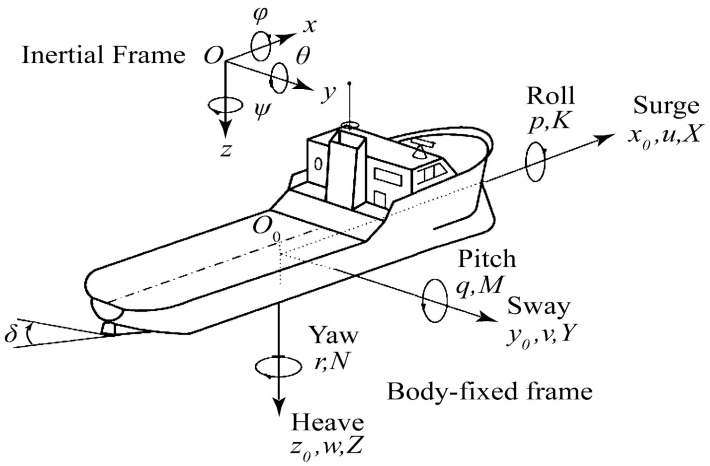

Figure 1. Ship motion model.

Ship maneuvering usually considers only the movement of the ship in the horizontal plane. For most ship movements and their control problems, it is possible to ignore heave, pitch, and roll motions, and thus only the sway, surge, and yaw are discussed. Therefore, the ship motion can be regarded as a plane motion, and the plane motion has only three degrees of freedom, which further simplifies the research object. The maneuvering mathematical model of a USV [36] can be written 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 and y denote the surge and sway, ψ denotes the yaw angle, u denotes the surge velocity, v denotes the sway velocity, r denotes the yaw velocity, m_{11}, m_{22}, and m_{33} are uncertainty parameters representing the intrinsic and additional masses, d_{11}, d_{22}, and d_{33} are also uncertainty parameters, d_{11} is the surge hydrodynamic damping factor, d_{22} is the sway hydrodynamic damping factor, and d_{33} is the yaw angle hydrodynamic damping factor.

For simplicity, the high-order nonlinear damping term is ignored. τ_u is the longitudinal force of the propeller, and τ_r is the torque of the propeller.

2.2. Nonlinear Ship Model

The nonlinear ship model is a dynamic system. The rudder angle δ is the input, and the yaw angle ψ is the output. By capturing the main threads of the ship's dynamics from $\delta \rightarrow \dot{\psi} \rightarrow \psi$, the resulting differential equation retains the nonlinear influences and even allows the effects of wind and wave disturbances to be converted into a disturbed rudder angle δ_D as an input signal that enters the ship's model together with the actual rudder angle δ. The responding nonlinear ship model is shown in Figure 2.

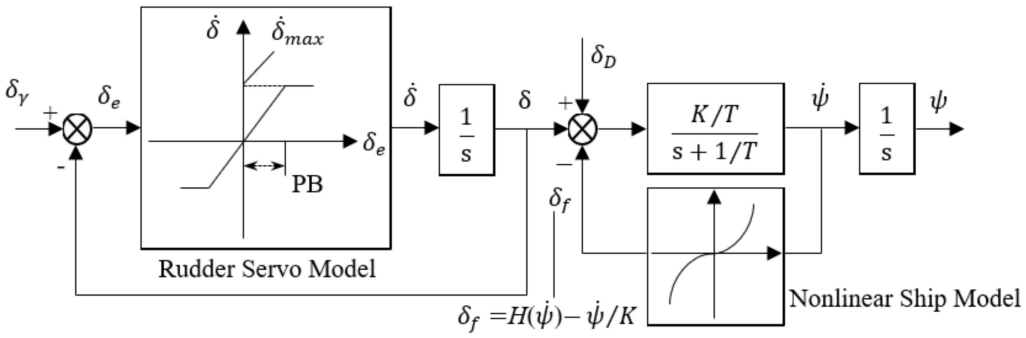

Figure 2. Responding nonlinear ship model.

The Norrbin nonlinear model for course control [37] can be derived from the ship motion maneuvering model in cases where steering is not very frequent:

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

where ψ and δ are the yaw angle and rudder angle. $a_1 = \alpha b$, $a_2 = \beta b$, and $b = K/T$, where K, T, α, and β are all related to the navigation speed and their values have a large impact on the accuracy of the simulation model.

3. FO PI$^{\lambda}$D$^{\mu}$ Controller

Fractional order (FO) calculus is essentially an arbitrary-order calculus whose order can be real or even complex, which is a further extension of integer order calculus. To improve the accuracy of the PID controller, the differential order μ and the integral order λ are extended to the real domain to obtain the FO PI$^{\lambda}$D$^{\mu}$ controller, whose differential equation [16] is expressed as

$$u(t) = K_p e(t) + K_i D_t^{-\lambda} e(t) + K_d D_t^{\mu} e(t) \tag{4}$$

where $D_t^{\mu} \equiv {}_c^a D_t^{\alpha}$ is defined by Caputo and $\lambda > 0$ and $\mu > 0$ are any real number and the order of the FO controller. We can then find the Laplace transform of the FO calculus defined by Caputo:

$$L\{{}_a^c D_t^{\alpha} f(t)\} = s^{\alpha} F(s) - \sum_{k=0}^{n-1} s^{a-k-1} f^{(k)}(0) \tag{5}$$

From Equations (4) and (5), the FO PI$^{\lambda}$D$^{\mu}$ controller transfer function can be obtained:

$$G_c(s) = K_p + \frac{K_i}{s^{\lambda}} + K_d s^{\mu} \tag{6}$$

Since the differential μ and the integral order λ are arbitrary positive real numbers, the PI controller is $\lambda = 1$ and $\mu = 0$, the PD controller is $\lambda = 0$ and $\mu = 1$, and the PID controller is $\lambda = 1$ and $\mu = 1$.

Since λ and μ are allowed to take non-integer values, the adjustment range of the controller parameters is changed from the "point" adjustment of the PID controller in Figure 3a to the "plane" adjustment in Figure 3b, thus enabling the FO PI$^{\lambda}$D$^{\mu}$ controller to control the controlled object more flexibly and to obtain better control results.

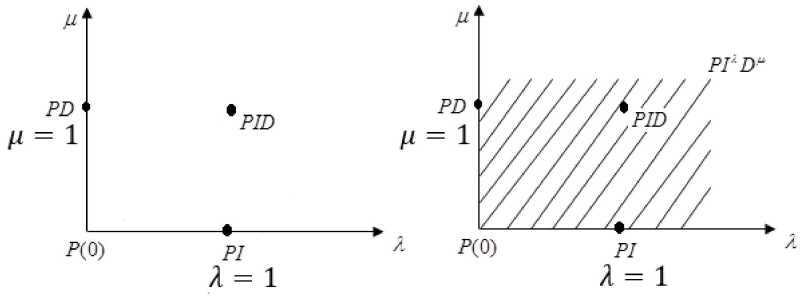

Figure 3. PID controller and FO PI$^{\lambda}$D$^{\mu}$ controller: (**a**) PID controller and (**b**) FO PI$^{\lambda}$D$^{\mu}$ controller.

In order to better study the FO PI$^{\lambda}$D$^{\mu}$ controllers, researchers wish to extend the design methods of integer-order controllers to FO controllers, which requires approximating the FO calculus to integer-order calculus. This paper uses an improved Oustaloup approximation for the FO calculus operator s^{α} to convert it to integer-order calculus [38]. We set the order to N, where (ω_b,ω_h) is the fitting frequency range and s^{α} is transformed into an integer-order transfer function as follows:

$$s^{\alpha} = \left(\frac{d\omega_b}{b}\right)^{\alpha}\left(\frac{ds^2 + bs\omega_h}{d(1-\alpha)s^2 + bs\omega_h + d\alpha}\right)\left[\frac{1+\frac{s}{d\omega_b/b}}{1+\frac{s}{d\omega_h/b}}\right]^{\alpha} \tag{7}$$

where $0 < \alpha < 1$, $b > 0$, $d > 0$, and $d = j\omega$. We express the FO part of Equation (7) (i.e., $K(s)$) as a rational transfer function in the form of zeros and poles:

$$K(s) = \lim_{N\to\infty} K_N(s) = \lim_{N\to\infty_k} \prod_{k=-N}^{N} \frac{1+s/\omega_k'}{1+s/\omega_k} \tag{8}$$

The k zero pole is

$$\omega_k' = \left(\frac{b}{d}\right)^{\frac{2k-\alpha}{2N+1}} \omega_h^{\frac{N+k+\frac{1}{2}(1-\alpha)}{2N+1}} \omega_b^{\frac{N-k+\frac{1}{2}(1+\alpha)}{2N+1}} \tag{9}$$

$$\omega_k = \left(\frac{b}{d}\right)^{\frac{2k+\alpha}{2N+1}} \omega_h^{\frac{N+k+\frac{1}{2}(1+\alpha)}{2N+1}} \omega_b^{\frac{N-k+\frac{1}{2}(1-\alpha)}{2N+1}} \tag{10}$$

An FO calculus operator is obtained by constructing a continuous rational transfer function model:

$$G(s) = K\left(\frac{ds^2 + b\omega_h s}{d(1-\alpha)s^2 + b\omega_h s + d\alpha}\right)\prod_{k=-N}^{N} \frac{1+s/\omega_k'}{1+s/\omega_k} \tag{11}$$

where $K = (\omega_b\omega_h)^{\alpha}$.

4. The FO PI$^{\lambda}$D$^{\mu}$ Controller of a Ship's Course

4.1. FO PI$^{\lambda}$D$^{\mu}$ Controller Design

The working principle of the FO PI$^{\lambda}$D$^{\mu}$ controller system of a ship's course is shown in Figure 4.

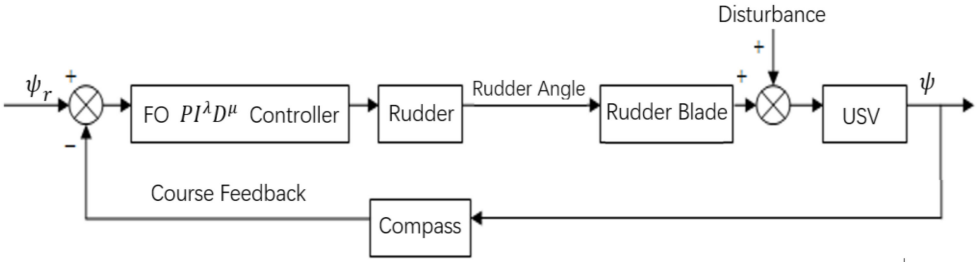

Figure 4. FO $PI^{\lambda}D^{\mu}$ control system of a ship's course.

In Figure 4, the actual course is determined with the compass and compared with the given course to generate the course deviation, which enters the FO $PI^{\lambda}D^{\mu}$ controller. It then calculates the rudder angle signal to be generated, and the rudder is turned to the required angle when the rudder receives the command signal to make the ship start to change its course. The rudder then returns to its zero position, completing the course revision.

In this paper, given the values of the gain margin A_m and phase margin ϕ_m, the parameters of the FO $PI^{\lambda}D^{\mu}$ controller parameters can be calculated [39]. Starting from the basic definitions of A_m and ϕ_m, the dynamically controlled object $G_p(s)$ and the controller $G_c(s)$ should satisfy the following relationships:

$$\phi_m = arg\left[G_c(j\omega_g)G_p(j\omega_g)\right] + \pi \tag{12}$$

$$A_m = \frac{1}{\left|G_c(j\omega_p)G_p(j\omega_p)\right|} \tag{13}$$

where ω_g meets

$$\left|G_c(j\omega_g)G_p(j\omega_g)\right| = 1 \tag{14}$$

and where ω_p meets

$$arg\left[G_c(j\omega_p)G_p(j\omega_p)\right] = -\pi \tag{15}$$

By substituting $G_c(s)$ with Equation (6), the following relationship is obtained:

$$K_p + K_i\frac{cos\frac{\pi\lambda}{2}}{\omega_p^{\lambda}} + K_d cos\frac{\pi\mu}{2}\omega_p^{\mu} = R_{mp} \tag{16}$$

$$K_p + K_i\frac{cos\frac{\pi\lambda}{2}}{\omega_g^{\lambda}} + K_d cos\frac{\pi\mu}{2}\omega_g^{\mu} = R_{mg} \tag{17}$$

$$-K_i\frac{sin\frac{\pi\lambda}{2}}{\omega_p^{\lambda}} + K_d sin\frac{\pi\mu}{2}\omega_p^{\mu} = I_{mp} \tag{18}$$

$$K_i\frac{sin\frac{\pi\lambda}{2}}{\omega_g^{\lambda}} + K_d sin\frac{\pi\mu}{2}\omega_g^{\mu} = I_{mg} \tag{19}$$

where

$$-\frac{1}{A_m G_p(j\omega_p)} = R_{mp} + jI_{mp} \tag{20}$$

$$\frac{-cos\varphi_m - jsin\varphi_m}{G_p(j\omega_g)} = R_{mg} + jI_{mg} \tag{21}$$

When the controller is designed, as the controlled object $G_p(s)$ and A_m and ϕ_m are known from Equations (16)–(19), the error squared minimization equation is introduced into the parameter calculation:

$$J = \int_0^{\infty} e^2(t)dt \tag{22}$$

Here, ω_p, ω_g, λ, and μ need to meet the following constraints:

$$\begin{array}{c}\left(\omega_g^{\lambda+\mu}-\omega_p^{\lambda+\mu}\right)\left(R_{mp}-R_{mg}\right)+\left(\omega_p^{\lambda+\mu}I_{mp}+\omega_g^{\lambda+\mu}I_{mg}\right)cot\left(\frac{\pi\mu}{2}\right)+\\ \left(\omega_p^{\lambda+\mu}I_{mg}+\omega_g^{\lambda+\mu}I_{mp}\right)cot\left(\frac{\pi\lambda}{2}\right)-\left(\omega_p^{\lambda}\omega_g^{\mu}I_{mp}+\omega_g^{\lambda}\omega_p^{\mu}I_{mg}\right)\left(cot\frac{\pi\lambda}{2}+cot\frac{\pi\mu}{2}\right)=0\end{array} \tag{23}$$

If ω_p, ω_g, λ, and μ are known, then the parameters K_p, K_i, and K_d are expressed as follows:

$$K_p=\left[\omega_p^{\lambda}R_{mp}-\omega_g^{\lambda}R_{mg}-\left(\omega_p^{\lambda}I_{mp}-\omega_g^{\lambda}I_{mg}\right)cot\left(\frac{\pi\mu}{2}\right)\right]/\left(\omega_p^{\lambda}-\omega_g^{\lambda}\right) \tag{24}$$

$$K_i=\frac{\omega_p^{\lambda}\omega_g^{\lambda}\left(\omega_g^{\mu}I_{mp}-\omega_p^{\mu}I_{mg}\right)}{\left(\omega_p^{\lambda+\mu}-\omega_g^{\lambda+\mu}\right)sin\left(\frac{\pi\lambda}{2}\right)} \tag{25}$$

$$K_d=\frac{\omega_p^{\lambda}I_{mp}-\omega_g^{\lambda}I_{mg}}{\left(\omega_p^{\lambda+\mu}-\omega_g^{\lambda+\mu}\right)sin\left(\frac{\pi\mu}{2}\right)} \tag{26}$$

4.2. Autotuning of the FO PI$^{\lambda}$D$^{\mu}$ Controller Based on the IPSO Algorithm

However, as the FO PI$^{\lambda}$D$^{\mu}$ controller is designed for ship course control at a rated speed, if the ship speed changes dynamically, the non-linear model of the ship will also change accordingly, which will result in the original controller parameters no longer meeting the needs of the ship's course control and needing to be recalculated. Therefore, many novel optimization algorithms [40–54] are used to tune the parameters of the controller. This paper proposes an IPSO algorithm for dynamic adjustment of the FO PI$^{\lambda}$D$^{\mu}$ controller parameters.

The PSO algorithm was originally designed to optimize the unpredictable movement of simulated flocks of birds. It is an iterative random search algorithm that is easy to implement, is robust, has good parallel processing capabilities, and is likely to obtain a global optimal solution [55,56]. However, in the later stages of optimization, the convergence time is long, the accuracy of the optimization is not high, and it easily falls into local extremes [57–59]. The mathematical description of the PSO algorithm [55,56] is as follows. We set the population size to N. At the iteration time, the position vector of each particle in the D-dimensional space is expressed as $\overline{x}_i(t)=\left(x_i^1,x_i^2,\ldots,x_i^d,\ldots,x_i^D\right)$. The velocity of each particle in the D-dimensional space is $\overline{v}_i(t)=\left(v_i^1,v_i^2,\ldots,v_i^d,\ldots,v_i^D\right)$. The optimal position for the particle i to search thus far is $\overline{p_i}(t)=\left(p_i^1,p_i^2,\ldots,p_i^d,\ldots,p_i^D\right)$. The optimal position for all of the particles searched thus far is $\overline{p_g}(t)=\left(p_g^1,p_g^2,\ldots,p_g^d,\ldots,p_g^D\right)$. The velocity $\overline{v}_i(t)$ and the position $\overline{x}_i(t)$ of particle i are adjusted at time $t+1$ as follows:

$$\overline{v}_i(t+1)\ =\ \omega\overline{v}_i(t)\ +\ c_1r_1(\overline{p}_i(t)\ -\ \overline{x}_i(t))\ +c_2r_2\left(\overline{p}_g(t)\ -\ \overline{x}_i(t)\right) \tag{27}$$

$$\overline{x}_i(t+1)\ =\ \overline{x}_i(t)\ +\ \overline{v}_i(t+1) \tag{28}$$

$$\begin{cases} v_i^d=v_{max},if\ v_i^d>v_{max} \\ v_i^d=-v_{max},if\ v_i^d<-v_{max} \end{cases} \tag{29}$$

where $i=[1,N]$, ω is the inertia weight and non-negative, learning factors c_1 and c_2 are non-negative, and r_1 and r_2 are pseudo-random numbers that obey a uniform distribution on $[0,\ 1]$. $v_i^d\in[-v_{max},v_{max}]$, where v_{max} is a constant. If the search space is in $[-x_{max},x_{max}]$, one can set $v_{max}=kv_{max}$, where $0.1\le k\le 1.0$.

From Equation (27), it can be seen that in the PSO algorithm, the particle seeking performance is related to the learning factors c_1 and c_2, which are used to regulate the maximum step size in the direction of motion of the particles $\overline{p_i}(t)$ and $\overline{p_g}(t)$ respectively.

A small learning factor can make the particles oscillate in the region away from the target, while a large learning factor can make the particles move rapidly toward the target region or even move away from the target. The learning factors c_1 and c_2 reflect the exchange of information between particle swarms and determine the influence of a single particle and swarm experience on the particle trajectories. The effect of the values of c_1 and c_2 on the search for an optimum is as follows:

- If c_1 and c_2 are both set to 0, the particle flight velocity will remain constant, making it impossible to search the entire space;
- If $c_1 = 0$, the particle loses its cognitive ability and easily falls into a local optimum solution;
- If $c_2 = 0$, which leads to a lack of communication and cooperation between particles, the algorithm will fall into a local minimum, and the probability of obtaining an optimal solution will become smaller;
- Smaller values for both c_1 and c_2 result in particles flying away from the target area and oscillating in that area;
- With larger values for both c_1 and c_2, particles fly faster toward the target area, and this may also cause particles to fly away from the target area.

In general, when using PSO algorithms to solve engineering optimization problems, the goal is to allow the particle in the swarm to perform optimization in the entire search space at the early stage to avoid the problem of lost local optima. In the later stages of the search, the goal is to increase the convergence accuracy and algorithm speed, which can lead to finding the global optimal solution quickly and efficiently. Therefore, the disadvantages of traditional methods can be overcome by introducing the dynamic accelerated learning factors c_1 and c_2 in the PSO algorithms. The maximum initial value of the learning factor is set to 2, while c_1 and c_2 are constructed as functions (i.e., monotonically decreasing and monotonically increasing functions, respectively) with the expressions shown in Equations (30) and (31):

$$c_1(t) = 2\left(1 - e^{(-t/t_{max})}\right) \tag{30}$$

$$c_2(t) = 2e^{(t-t_{max})/t_{max}} \tag{31}$$

Therefore, the particle swarm velocity update equation is

$$\overline{v}_i(t+1) = \omega\overline{v}_i(t) + c_1(t)r_1(\overline{p}_i(t) - \overline{x}_i(t)) + c_2(t)r_2\left(\overline{p}_g(t) - \overline{x}_i(t)\right) \tag{32}$$

In the early stages of evolution, if c_1 takes a larger value and c_2 takes a smaller value, this will allow the particles to move throughout the search space. In the later stages, if c_1 takes a smaller value and c_2 takes a larger value, this will increase the convergence rate toward the optimal solution.

In this paper, two classical benchmark functions (Griewank and Schaffer) are selected to conduct comparative simulations on the PSO algorithm [31] and the IPSO algorithm with the dynamic acceleration learning factor, which are used to verify the correctness and superiority of the IPSO algorithm.

The Griewank function is as follows:

$$minf(x_i) = \sum_{i=1}^{N} \frac{x_i^2}{4000} - \prod_{i=1}^{N} cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \tag{33}$$

where $x_i \in [-600, 600]$. The function has a local minimal value, the number of which is related to the dimensions of the problem, with a global minimum of 0, and $(x_1, x_{2,} \ldots, x_n) = (0, 0, \ldots, 0)$ can be obtained. The function is typically a nonlinear multi-modal function with a wide search space and is often considered a complex multi-modal problem that is difficult for optimization algorithms to handle.

The Schaffer function is as follows:

$$minf(x_1, x_2) = 0.5 + \frac{\left(sin\sqrt{x_{1+}^2 x_2^2}\right)^2 - 0.5}{\left[1 + 0.01\left(x_{1+}^2 x_2^2\right)\right]^2} \tag{34}$$

where $x_1, x_2 \in [-10, 10]$. The function is a two-dimensional complex function with an infinite number of minimal value points, achieving a minimum value of 0 at (0,0). Since the function has a strongly oscillatory nature, it is difficult to find a globally optimal solution. The curves of the Griewank and Schaffer functions when their independent variables are two-dimensional are shown in Figure 5.

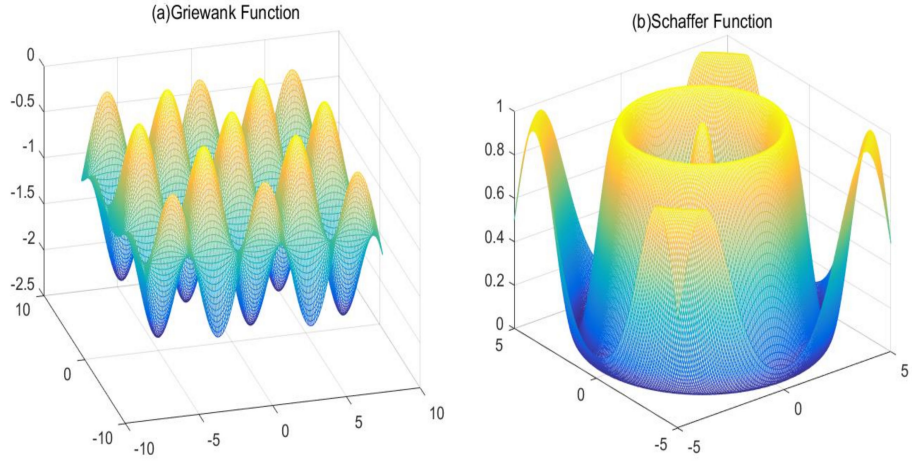

Figure 5. Distribution chart of test functions.

In this paper, the Griewank and Schaffer functions are used to test the performance of the IPSO algorithm. The parameter settings are shown in Table 1, and the test results are shown in Figure 6.

Table 1. Parameter settings for different PSO algorithms.

Algorithm	Particles m	Weight ω	Learning Factors c_1, c_2	Maximum Speed V_{max}	Maximum Iterations T_{max}
PSO	20	1	2, 2	1	100
IPSO	20	1	Equations (30) and (31)	1	100

From the simulations, it can be seen that the test results using the Griewank function show that the PSO and IPSO algorithms could converge to the optimal solution, but the number of iterations used to find the optimal IPSO algorithm was significantly smaller than that of the PSO algorithm. The results of the Schaffer function test show that the PSO and IPSO algorithms could converge to the optimal solution, but the number of iterations used to find the optimal solution was significantly smaller for the IPSO algorithm than for the PSO algorithm. Therefore, it is reasonable to believe that the addition of dynamic learning factors to the PSO algorithm in the process of finding the best search faster and with higher convergence accuracy and higher stability to solve the PSO algorithm easily fell into the local optimum problem.

In this part, the IPSO algorithm is used to autotune the five parameters (K_p, K_i, K_d, λ, and μ) of the FO PI$^{\lambda}$D$^{\mu}$ controller in Equation (6), which is used for course-keeping of the USV. The working principal diagram and model of the FO PI$^{\lambda}$D$^{\mu}$ controller with the IPSO algorithm are shown in Figures 7 and 8.

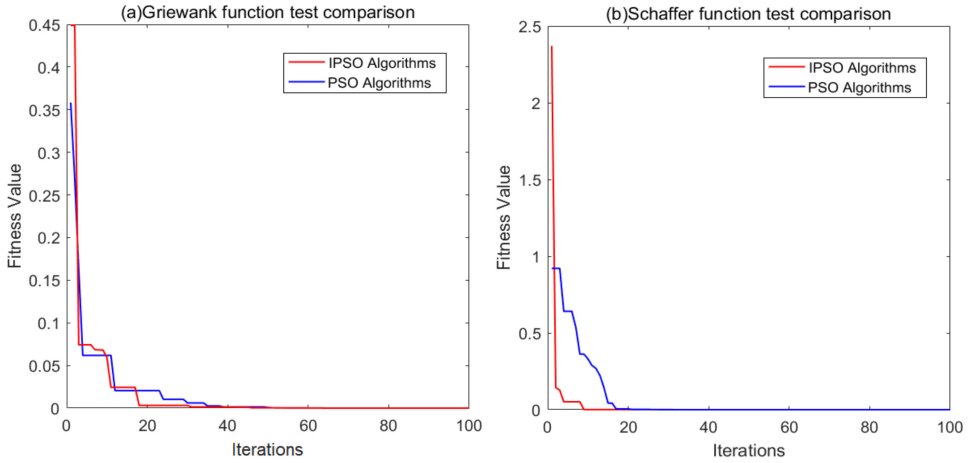

Figure 6. Performance comparison of PSO and IPSO algorithms.

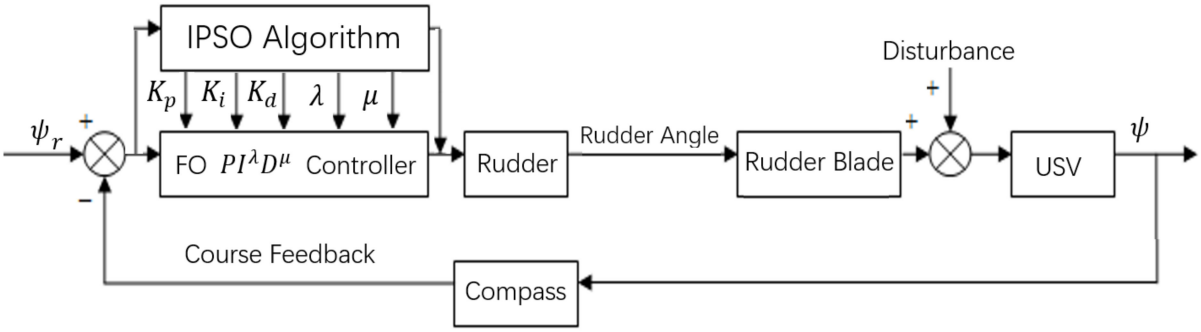

Figure 7. FO $PI^{\lambda}D^{\mu}$ control system of ship's course based on IPSO.

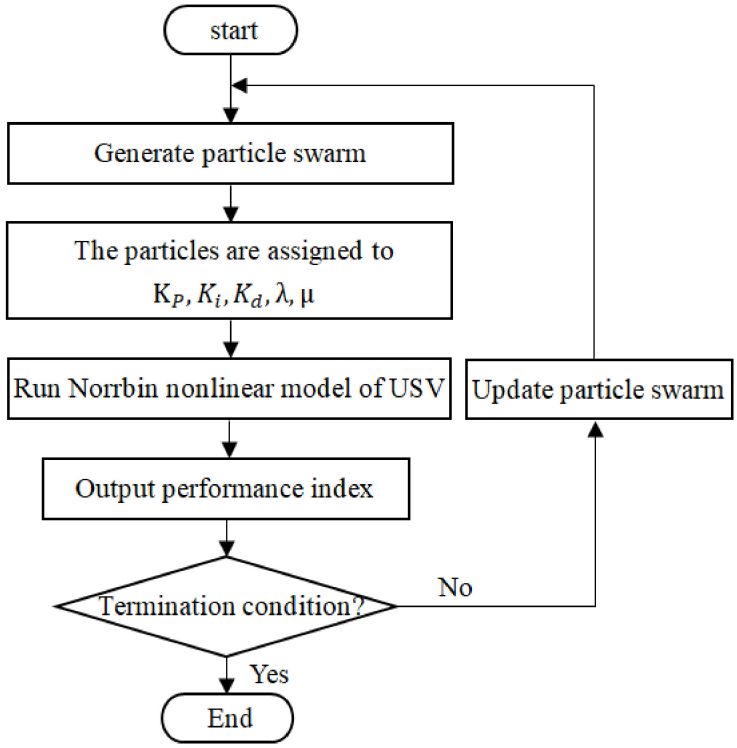

Figure 8. Process of IPSO optimizing the FO $PI^{\lambda}D^{\mu}$ controller.

The IPSO algorithm is used to autotune the five parameters (K_p, K_i, K_d, λ, and μ) of the FO $PI^{\lambda}D^{\mu}$ control, which is a five-dimensional solution space. Considering the system

overshoot and the square integral of the control loop error, the equation of the optimization objective function is as follows:

$$J = \varpi_1 \int_0^{+\infty} e^2(t)dt + \varpi_2 \sigma \tag{35}$$

where $e(t)$ is the systematic error, σ is the system overshoot, and ϖ_1 and ϖ_2 are weights.

5. Simulations

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

The research object used the actual ship data of COSCO's large container vessel of 5446 TEU, whose parameters are shown in Table 2. Let the ship's rated speed be V = 19.8 knots = 10.186 m/s (1 knot = 1.852 km/s) and the ship's nonlinear course model adopt the Norrbin nonlinear model. Since the ship could be regarded as a large mass object in movement, the ship's course control could be regarded as a kind of slow turning movement when the steering was not very frequent, so Equation (3) could be simplified. The simplified equation [37] is shown here:

$$T\dot{\psi} + \psi = K\delta \tag{36}$$

Table 2. Structure data of 5446 TEU container vessel.

Structure	Value	Structure	Value
length of ship	280 m	length of two columns	267 m
width of ship	39.8 m	distance of gravity center	2.64 m
area of Rudder	61.0 m^2	square coefficient	0.67
no-load weight	3.5453 M tons	full load weight	6.5531 M tons
draft	12.532 m	draft with fully load	14.023 m

According to the calculation, the model coefficients K = 0.1955 and T = 255.8837 were obtained. A_m = 1.3, $\varnothing_m = \pi/4$, λ = 0.1, and μ = 0.85 for the FO $PI^{\lambda}D^{\mu}$ controller were determined by the system characteristics and practical experience. Then, K_p = 1.4516, K_i = 1.1355, and K_d = 95.0636 were calculated from Equations (22)–(26), respectively. The fitting frequency of the modified Oustaloup filtering method (ω_b, ω_h) chose [0.001, 1000] and the order N = 5. According to the design method of the *PID* ship autopilot [37], the K_p, K_i, and K_d are shown here:

$$K_p = \frac{T\omega_n^2}{K}, K_i = \frac{T\omega_n^3}{10K}, K_d = \frac{2T\varepsilon\omega_n - 1}{K} \tag{37}$$

where ω_n is the system's natural frequency and ε is the system's relative attenuation coefficient. Usually, one would have $0.8 \leq \varepsilon \leq 1.0$, ω_n = 0.06, and ε = 0.9. From Equation (37), we could obtain K_p = 4.712, K_i = 0.028, and K_d = 136.243. The FO $PI^{\lambda}D^{\mu}$ controller was applied to the ship's course, where the maximum rudder angle was restricted from −35° to +35°, and the desired course angle was 30° when time was 0–1000 s. The FO $PI^{\lambda}D^{\mu}$ controller and PID controller were compared. The comparison results can be seen in Figures 9–11 and Table 3.

In the next step, we set the ship's course angle to change in the following order: (1) $0 \leq t \leq 1000$ while setting the desired course angle to $\psi_r(t) = 20°$ and (2) $1000 \leq t \leq 2000$, while setting the desired course angle to $\psi_r(t) = 30°$. The course-tracking curve and rudder angle curve with the FO $PI^{\lambda}D^{\mu}$ controller and the PID controller are shown in Figures 12 and 13.

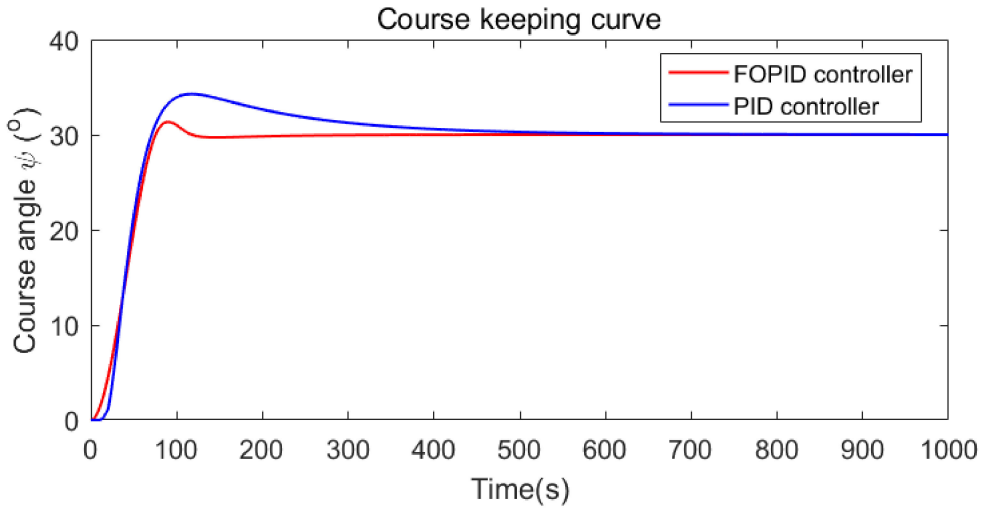

Figure 9. Course-keeping curve with FO PI$^{\lambda}$D$^{\mu}$ controller and the PID controller.

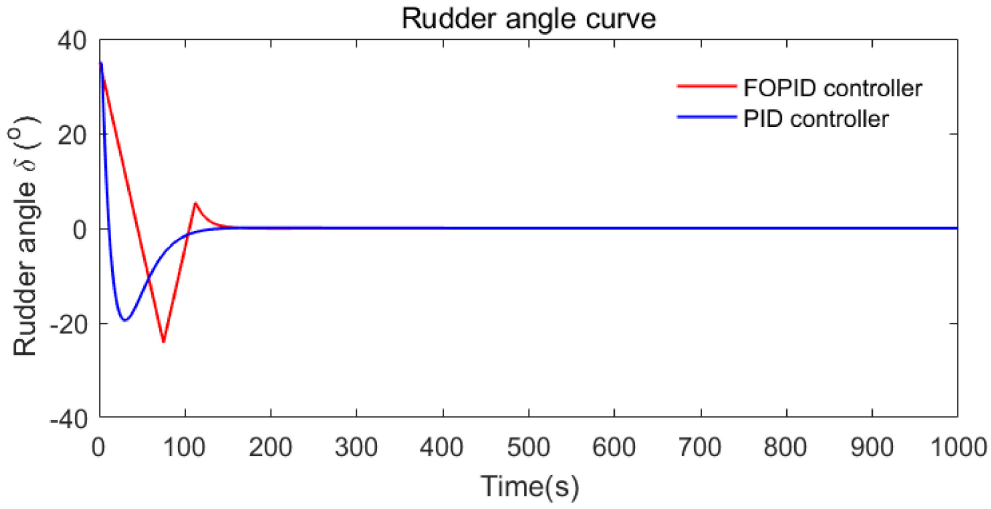

Figure 10. Rudder angle curve with FO PI$^{\lambda}$D$^{\mu}$ controller and the PID controller.

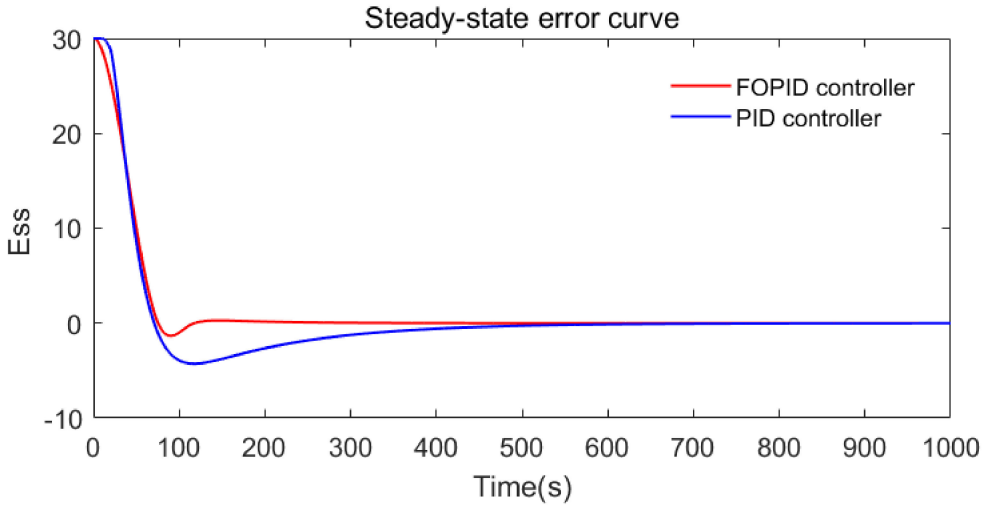

Figure 11. Steady state error curve with FO PI$^{\lambda}$D$^{\mu}$ controller and the PID controller.

Table 3. Performance of PID and FO PI$^{\lambda}$D$^{\mu}$ controller.

Controller Type	Adjust Time t_s	Rise Time t_r	Overshoot M	E_{ss}
FO PI$^{\lambda}$D$^{\mu}$	114	75	4.42%	0.002
PID controller	519	71	14.25%	0.007

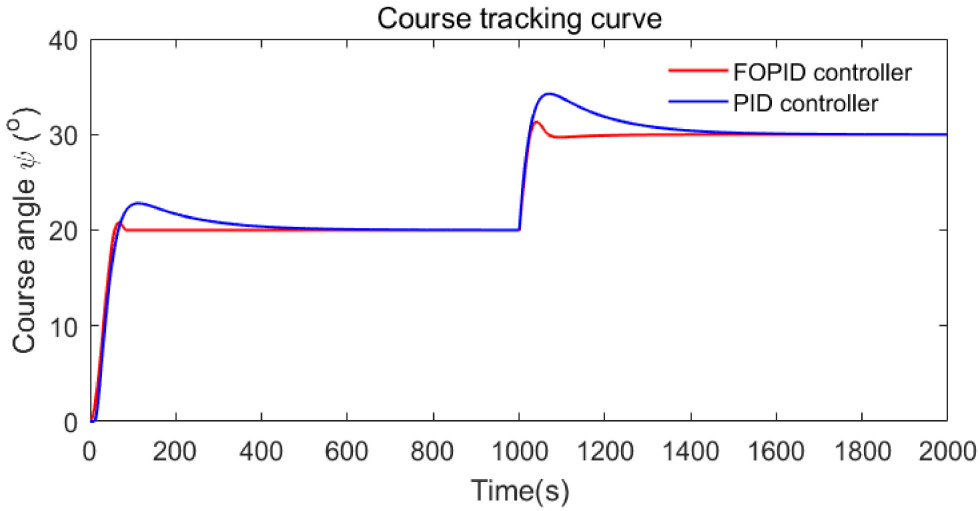

Figure 12. Course-tracking curves with FO PI$^{\lambda}$D$^{\mu}$ controller and PID controller.

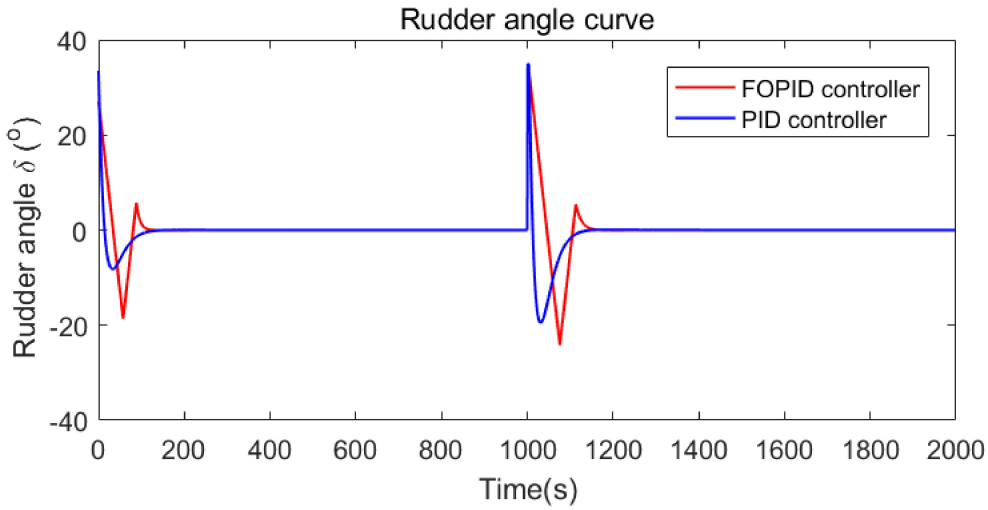

Figure 13. Rudder angle curves with FO PI$^{\lambda}$D$^{\mu}$ controller and PID controller.

By comparing the simulation experiments, it can be seen that the FO PI$^{\lambda}$D$^{\mu}$ controller had a significantly better control effect than the PID automatic rudder controller proposed in [37]. The FO PI$^{\lambda}$D$^{\mu}$ controller had the advantages of a small overshoot, short adjustment time, and accurate control.

5.2. FO PI$^{\lambda}$D$^{\mu}$ Controller Based on the IPSO Algorithm

Since the ship's nonlinear model parameters changed with the speed, the calculated FO PI$^{\lambda}$D$^{\mu}$ controller parameters could not meet the dynamic changes of the model's parameters. The corresponding nonlinear model's parameter values for ship course control at different speeds are shown in Table 4.

Table 4. Corresponding model parameters under different speeds.

V (knots)	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

In the next stage, the objective was to check the self-adjustment of the controller based on IPSO. The simulation used the Norrbin nonlinear model in Equation (3). The five parameters of the FO $PI^{\lambda}D^{\mu}$ controller were adjusted using the IPSO algorithm for the ship's course, and the effectiveness of the algorithm was verified by simulation experiments. Following the design step of the FO $PI^{\lambda}D^{\mu}$ controller, the fitting frequency of the Oustaloup filtering method (ω_b, ω_k) was [0.001,1000] and $N = 5$. The population size was set to 50. The maximum number of the birds' steps was 50. The particle dimensions were 5, meansing [K_p, K_i, K_d, λ, μ], and the initial values of c_1 and c_2 were set at 2. The inertia weight was set to 0.9. Four different control strategies were used for the ship's course, which were the PID controller [37], the PID controller based on the PSO algorithm [9], the FO $PI^{\lambda}D^{\mu}$ controller based on the GA algorithm [19], and the FO $PI^{\lambda}D^{\mu}$ controller based on the IPSO algorithm. The ship's speed increased from V = 19.8 knots to V = 24.5 knots at a constant speed, the maximum rudder angle was restricted from $-35°$ to $+35°$, and the desired course angle was 30° when the time was 0–1000 s. The comparison results can be seen in Figures 14 and 15 and Table 5.

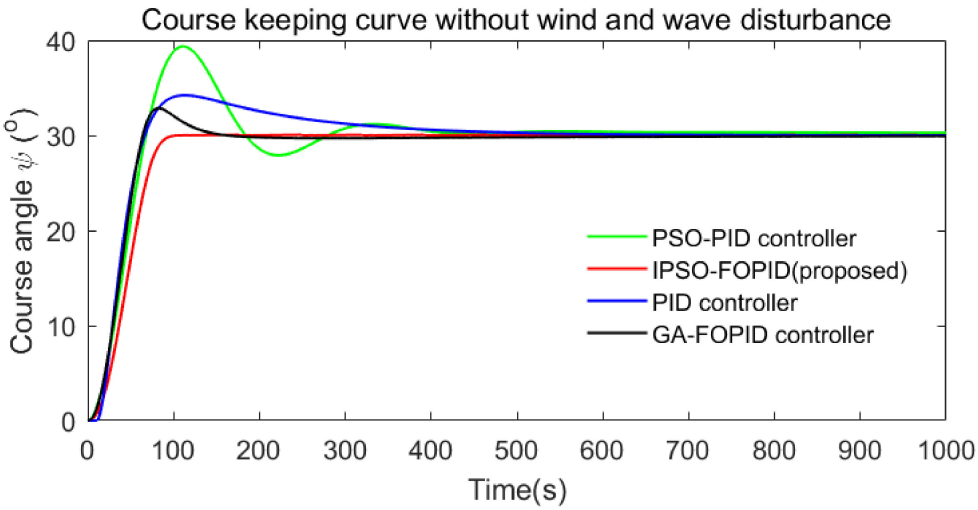

Figure 14. Course-keeping curves without disturbance.

Table 5. Performance of four different controllers.

Controller Type	Adjust Time t_s	Rise Time t_r	Overshoot M	E_{ss}
IPSO-FOPID (proposed)	103	103	0.03%	0.005
PSO-PID	416	67	31.16%	0.024
PID	519	71	14.25%	0.007
GA-FOPID	163	65	9.53%	0.086

Meanwhile, wind and wave disturbances have been the main causes of yawing of ships when they are sailing at sea. In this design, wind and wave disturbances were introduced into the system, which verified the robustness. The disturbances used white noise to drive a typical second-order oscillation. The wave model [60] obtained under the

effect of a force-six wind is that of Equation (38), and the simulation curve is shown in Figure 16.

$$h(s) = \frac{0.4198s}{s^2 + 0.3638s + 0.3675} \tag{38}$$

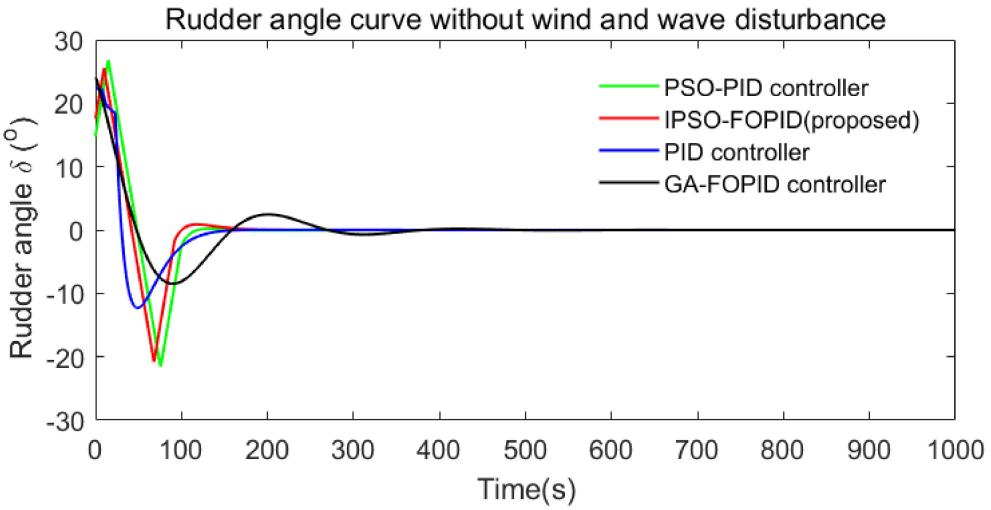

Figure 15. Rudder angle curves without disturbance.

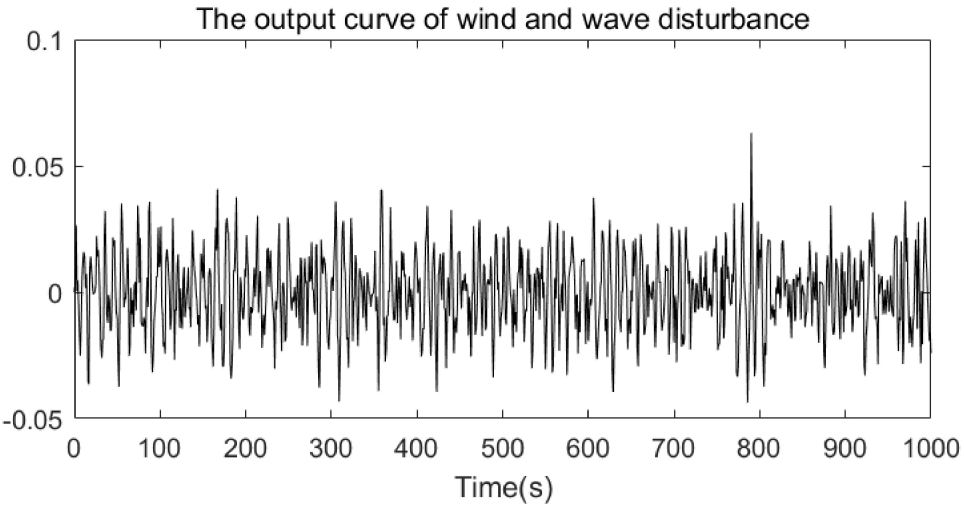

Figure 16. The output curve of wind and wave disturbances.

Four different control strategies were used for a ship's course under wind and waves, and the results can be seen in Figures 17 and 18.

It can be seen from Figures 17 and 18 that the FO PI$^{\lambda}$D$^{\mu}$ controller based on the IPSO algorithm had a significantly better control effect than the automatic rudder PID controller, the PID controller based on the PSO algorithm, and the FO PI$^{\lambda}$D$^{\mu}$ controller based on a GA algorithm. The FO PI$^{\lambda}$D$^{\mu}$ controller based on the IPSO algorithm had the advantages of a small overshoot, short adjustment time, accurate control, and strong wind and wave resistance.

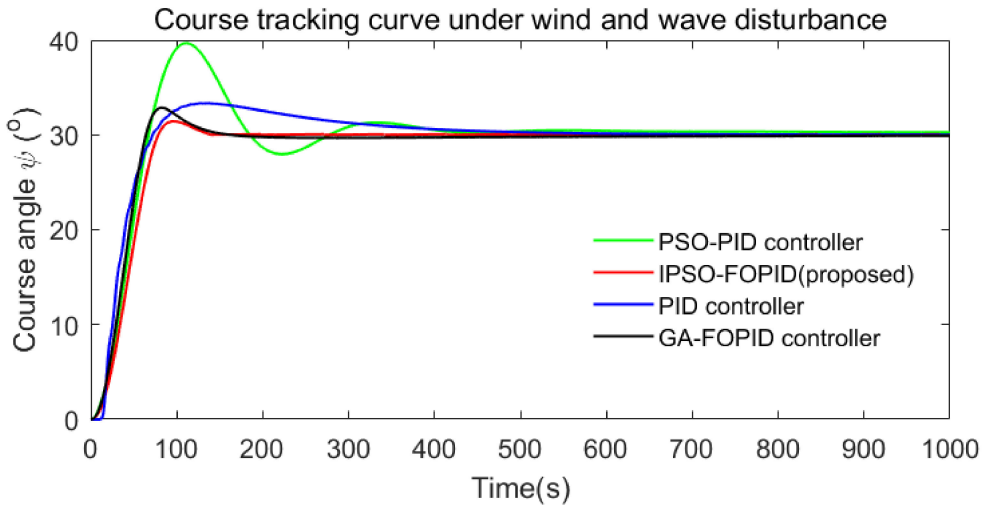

Figure 17. Course-keeping curves under wind and wave disturbances.

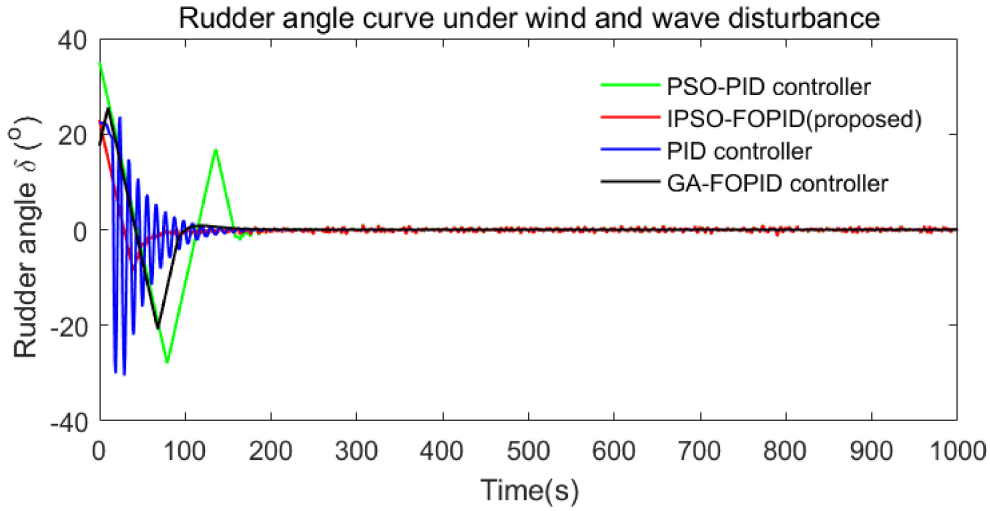

Figure 18. Rudder angle curves under wind and wave disturbances.

6. Conclusions

This paper proposed two methods for an FO PI$^{\lambda}$D$^{\mu}$ controller for USV course control. First, this paper proposed an FO PI$^{\lambda}$D$^{\mu}$ controller based on frequency domain specification, which used a desired gain margin and phase margin to meet the performance requirements of the ship's course control. The simulation results showed that the FO PI$^{\lambda}$D$^{\mu}$ controller had the advantages of a small overshoot, short adjustment time, and precise control over the automatic rudder PID controller. However, if the ship's speed changed, the ship's nonlinear course model would change. The parameters of the FO PI$^{\lambda}$D$^{\mu}$ controller no longer met the needs of course control. Secondly, the FO PI$^{\lambda}$D$^{\mu}$ controller based on the IPSO algorithm was proposed for USV course control. To address the influence of the learning factor on the search speed of the particles in the solution space, this paper constructed a particle swarm optimization algorithm with a time-varying learning factor. In the initial stage of optimization, the particles had a stronger self-learning ability and weaker social learning ability to strengthen the global search ability. In the later stage, the particles had a stronger social learning ability and weaker self-learning ability to facilitate convergence at the global optimum. Comparative experiments were carried out without wind and wave disturbances and with wind and wave disturbances. The results show that the FO PI$^{\lambda}$D$^{\mu}$ controller based on the IPSO algorithm had the advantages of a small overshoot, short adjustment time, precise control, and strong anti-disturbance control. The controller output was subjected to wind and wave interference, and the control object itself changed due

to certain factors. It could also overcome these effects and had a strong robustness. The control performance was better than that of the three other controllers, and it was easy to modify the existing PID-type automatic rudder while having a high promotion value.

In light of the effectiveness of the FO $PI^{\lambda}D^{\mu}$ controller based on the IPSO algorithm, more improvements need to be studied in future. An interesting aspect for future study is how to combine an FO $PI^{\lambda}D^{\mu}$ controller with other intelligent algorithms (e.g., neural networks, Lévy flight distribution, the Nelder–Mead algorithm, and fuzzy control) and apply them to a ship's course control to further verify the superiority of the FO $PI^{\lambda}D^{\mu}$ controller in ship course control. Another aspect which needs to be studied involves the use of FO $PI^{\lambda}D^{\mu}$ controllers for a ship's trajectory control.

Author Contributions: Conceptualization, G.L. and Y.L.; methodology, G.L. and Y.L.; software, H.C.; validation, G.L. and H.C.; data curation, G.L. and Y.L.; writing—original draft preparation, G.L. and Y.L.; writing—review and editing, W.D. and H.C.; project administration, H.C.; funding acquisition, G.L. and Y.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.

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. Park, B.S.; Kwon, J.W.; Kim, H. Neural network-based output feedback control for reference tracking of underactuated surface vessels. *Automatica* **2017**, *77*, 353–359. [CrossRef]
2. Lu, Y.; Zhang, G.; Sun, Z.; Zhang, W. Robust adaptive formation control of underactuated autonomous surface vessels based on MLP and DOB. *Nonlinear Dyn.* **2018**, *94*, 503–519. [CrossRef]
3. Liu, C.; Negenborn, R.R.; Chu, X.; Zheng, H. Predictive path following based on adaptive line-of-sight for underactuated autonomous surface vessels. *J. Mar. Sci. Technol.* **2018**, *23*, 483–494. [CrossRef]
4. Sahu, B.K.; Subudhi, B.; Gupta, M.M. Stability Analysis of an Underactuated Autonomous Underwater Vehicle Using Extended-Routh's Stability Method. *Int. J. Autom. Comput.* **2018**, *15*, 299–309. [CrossRef]
5. Xie, W.; Reis, J.; Cabecinhas, D.; Silvestre, C. Design and experimental validation of a nonlinear controller for underactuated surface vessels. *Nonlinear Dyn.* **2020**, *102*, 2563–2581. [CrossRef]
6. Kolmanovsky, I.; McClamroch, N.H. Developments in nonholonomic control problems. *IEEE Control Syst. Mag.* **1995**, *15*, 20–36.
7. Le, M.D.; Nguyen, S.H.; Nguyen, L.A. Study on a new and effective fuzzy PID ship autopilot. *Artif. Life Robot.* **2004**, *8*, 197–201. [CrossRef]
8. Gao, C.; Guo, J. On the existence of an optimal control of ship automatic steering instruments. *J. Ocean Univ. China* **2005**, *4*, 185–188. [CrossRef]
9. Dlabač, T.; Ćalasan, M.; Krčum, M.; Marvučić, N. PSO-BASED PID controller design for ship course-keeping autopilot. *Brodogradnja* **2019**, *70*, 1–15. [CrossRef]
10. Liu, S.; Fang, L. Application of H∞ control in rudder/flap vector robust control of a ship's course. *J. Mar. Sci. Appl.* **2007**, *6*, 62–67. [CrossRef]
11. Zhang, Q.; Zhang, M.; Hu, Y.; Zhu, G. Error-driven-based adaptive nonlinear feedback control of course-keeping for ships. *J. Mar. Sci. Technol.* **2021**, *26*, 357–367. [CrossRef]
12. Wu, J.; Zeng, C.; Hu, Y. Indirect Adaptive Robust Control Design for Course Tracking of Ships Subject to Unknown Control Coefficient and Disturbances. *Int. J. Control Autom. Syst.* **2021**, *19*, 2059–2067. [CrossRef]
13. Oldham, K.B.; Spanier, J. *The Fractional Calculus*; Academic Press: New York, NY, USA, 1974.
14. Miller, K.S.; Ross, B. *An Introduction to the Fractional Calculus and Fractional Differential Equations*; Wiley: New York, NY, USA, 1993.
15. Podlubny, I. *Fractional Differential Equations*; Academic Press: San Diego, CA, USA, 1999.
16. Podlubny, I. Fractional-order System and Fractional-order Control. *IEEE Trans. Autom. Control* **1999**, *44*, 208–214. [CrossRef]
17. Podlubny, I. Realization of Fractional-Order Control. *Acta Montan. Slovaca* **2003**, *8*, 233–235.
18. Podlubny, I.; Petráš, I.; Vinagre, B.M. Analogue Realizations of Fractional-Order Controllers. *Nonlinear Dyn.* **2002**, *29*, 281–296. [CrossRef]
19. 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. Electr. Eng. Comput. Sci.* **2012**, *20*, 1123–1136.

20. Zhang, C.; Tan, N.; Zhou, T.; Liu, M.; Shan, H. Research on Optimal Control of Subway Train Based on Fractional Order PID Controller. *J. China Railw. Soc.* **2018**, *40*, 8–14.
21. Qi, Z.D.; Bian, H.J.; Leng, B.Y.; Shan, L. Design of a new fractional order PI^{λ}–PD^{μ} controller based on SQP. *Control Decis.* **2016**, *31*, 2275–2279.
22. Safikhani Mohammadzadeh, H.; Tabatabaei, M. Design of Non-overshooting Fractional-Order PD and PID Controllers for Special Case of Fractional-Order Plants. *J. Control Autom. Electr. Syst.* **2019**, *30*, 611–621. [CrossRef]
23. Kumar, J.; Kumar, V.; Rana, K.P.S. Fractional-order self-tuned fuzzy PID controller for three-link robotic manipulator system. *Neural Comput. Appl.* **2020**, *32*, 7235–7257. [CrossRef]
24. Nasir, A.; Rasool, I.; Sibtain, D.; Kamran, R. Adaptive Fractional Order PID Controller Based MPPT for PV Connected Grid System under Changing Weather Conditions. *J. Electr. Eng. Technol.* **2021**, *16*, 2599–2610. [CrossRef]
25. Izci, D.; Ekinci, S.; Eker, E.; Kayri, M. A novel modified opposition-based hunger games search algorithm to design fractional order proportional-integral-derivative controller for magnetic ball suspension system. *Adv. Control Appl. Eng. Ind. Syst.* **2022**, *4*, e96. [CrossRef]
26. Altbawi, S.M.A.; Mokhtar, A.S.B.; Jumani, T.A. Optimal design of Fractional order PID controller based Automatic voltage regulator system using gradient-based optimization algorithm. *J. King Saud Univ. Eng. Sci.* **2021**, in press. [CrossRef]
27. 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**, in press. [CrossRef]
28. Cui, H.; Guan, Y.; Chen, H. Rolling element fault diagnosis based on VMD and sensitivity MCKD. *IEEE Access* **2021**, *9*, 120297–120308. [CrossRef]
29. Deng, W.; Xu, J.; Gao, X.; Zhao, H. An enhanced MSIQDE algorithm with novel multiple strategies for global optimization problems. *IEEE Trans. Syst. Man. Cybern. Syst.* **2022**, *52*, 578–1587. [CrossRef]
30. Ran, X.; Zhou, X.; Lei, M.; Tepsan, W.; Deng, W. A novel k-means clustering algorithm with a noise algorithm for capturing urban hotspots. *Appl. Sci.* **2021**, *11*, 11202. [CrossRef]
31. Deng, W.; Zhang, X.X.; Zhou, Y.Q.; Liu, Y.; Zhou, X.B.; Chen, H.L.; Zhao, H.M. An enhanced fast non-dominated solution sorting genetic algorithm for multi-objective problems. *Inform. Sci.* **2022**, *585*, 441–453. [CrossRef]
32. Li, T.Y.; Qian, Z.J.; Deng, W.; Zhang, D.Z.; Lu, H.H.; Wang, S.H. Forecasting crude oil prices based on variational mode decomposition and random sparse Bayesian learning. *Appl. Soft Comput.* **2021**, *113*, 108032. [CrossRef]
33. Shao, H.D.; Lin, J.; Zhang, L.W.; Galar, D.; Kumar, U. A novel approach of multisensory fusion to collaborative fault diagnosis in maintenance. *Inf. Fusion* **2021**, *74*, 65–76. [CrossRef]
34. Deng, W.; Xu, J.; Zhao, H.; Song, Y. A novel gate resource allocation method using improved PSO-based QEA. *IEEE Tran. Intell. Transp. Syst.* **2022**, *23*, 1737–1745. [CrossRef]
35. Wu, X.H. *Ship Maneuverability and Seakeeping*; China Communications Press: Beijing, China, 1999.
36. Wei, M.; Guo, C.; Liu, Y. Robust Adaptive path following for underactuated surface vessels with uncertain dynamics. *J. Mar. Sci.* **2012**, *11*, 244–250.
37. Fossen, T.I. *Guidance and Control of Ocean Vehicle*; John Wiley & Sons Ltd.: Chichester, UK, 1994.
38. 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]
39. Xue, D.Y.; Zhao, C.N. Fractional order PID controller design for fractional order system. *Control Theory Appl.* **2007**, *24*, 771–776.
40. Li, T.Y.; Shi, J.Y.; Zhang, D.Z. Color image encryption based on joint permutation and diffusion. *J. Electron. Imaging* **2021**, *30*, 013008. [CrossRef]
41. 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]
42. Chen, H.; Zhang, Q.; Luo, J. An enhanced Bacterial Foraging Optimization and its application for training kernel extreme learning machine. *Appl. Soft Comput.* **2020**, *86*, 105884. [CrossRef]
43. Deng, W.; Shang, S.; Cai, X.; Zhao, H.; Zhou, Y.; Chen, H.; Deng, W. Quantum differential evolution with cooperative coevolution framework and hybrid mutation strategy for large scale optimization. *Knowl.-Based Syst.* **2021**, *224*, 107080. [CrossRef]
44. Wang, X.; Wang, H.Y.; Du, C.Z.; 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.
45. Wei, Y.Y.; Zhou, Y.Q.; Luo, Q.F.; Deng, W. Optimal reactive power dispatch using an improved slime mould algorithm. *Energy Rep.* **2021**, *7*, 8742–8759. [CrossRef]
46. Wang, L.Z.; Zhang, J.B.; Liu, P.; Choo, K.K.; Huang, F. Spectral-spatial multi-feature-based deep learning for hyperspectral remote sensing image classification. *Appl. Soft Comput.* **2017**, *21*, 213–221. [CrossRef]
47. Luo, X.; Shen, Z.; Xue, R. Unsupervised band selection method based on importance-assisted column subset selection. *IEEE Access* **2018**, *7*, 517–527. [CrossRef]
48. Chang, C.I.; Kuo, Y.M.; Chen, S. Self-mutual information-based band selection for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 5979–5997. [CrossRef]
49. Lin, Z.; Yan, L. A support vector machine classifier based on a new kernel function model for hyperspectral data. *Mapp. Sci. Remote Sens.* **2015**, *53*, 85–101. [CrossRef]

50. He, Z.Y.; Shao, H.D.; Zhong, X.; Zhao, X.Z. Ensemble transfer CNNs driven by multi-channel signals for fault diagnosis of rotating machinery cross working conditions. *Knowl.-Based Syst.* **2020**, *207*, 106396. [CrossRef]
51. Li, T.Y.; Shi, J.Y.; Deng, W.; Hu, Z.D. Pyramid particle swarm optimization with novel strategies of competition and cooperation. *Appl. Soft Comput.* **2021**, *121*, 108731.
52. 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**, in press. [CrossRef]
53. Izci, D. A novel improved atom search optimization algorithm for designing power system stabilizer. *Evol. Intell.* **2021**, in press. [CrossRef]
54. Izci, D. Design and application of an optimally tuned PID controller for DC motor speed regulation via a novel hybrid Lévy flight distribution and Nelder–Mead algorithm. *Trans. Inst. Meas. Control* **2021**, *43*, 3195–3211. [CrossRef]
55. Eberhart, R.C.; Kennedy, J. A New Optimizer Using Particle Swarm Theory. In Proceedings of the 6th International on Symposium of Micromachine Human Science, Nagoya, Japan, 4–6 October 1995; pp. 39–43.
56. Kennedy, J.; Eberhart, R.C. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; pp. 1942–1948.
57. Angeline, P.J. Using selection to improve particle swarm optimization. In Proceedings of the IEEE International Conference on Evolutionary Computation and IEEE World Congress on Computational Intelligence, Anchorage, AK, USA, 4–9 May 1998; pp. 84–89.
58. Naka, S.; Genji, T.; Yura, T.; Fukuyama, Y. A hybrid particle swarm optimization for distribution state estimation. *IEEE Trans. Power Syst.* **2003**, *18*, 60–68. [CrossRef]
59. Wonohadidjojo, D.M.; Kothapalli, G.; Hassan, M.Y. Position control of electro-hydraulic actuator system using fuzzy logic controller optimized by particle swarm optimization. *Int. J. Autom. Comput.* **2013**, *10*, 181–193. [CrossRef]
60. Jia, X.L.; Zhang, X.K. *Intelligent Control of Ship Motion with H∞ Robust Control*; Dalian Maritime University Press: Dalian, China, 2002.