

Article

A Temperature Control Method of Lysozyme Fermentation Based on LRWOA-LSTM-PID

Chenhua Ding ^{1,2}, Xungen Li ^{1,2}, Hanlin Zhou ¹, Jianming Yu ¹, Juling Du ^{3,*} and Shixiang Zhao ^{1,2} 

¹ College of Electronics and Information, Hangzhou Dianzi University, Hangzhou 310018, China; 211040073@hdu.edu.cn (C.D.); lixg@hdu.edu.cn (X.L.); zhl2430@163.com (H.Z.); mktacclhzbyeye@gmail.com (J.Y.); zsx15216117795@163.com (S.Z.)

² Pujiang Microelectronics and Intelligent Manufacturing Research Institute, Hangzhou Dianzi University, Jinhua 322200, China

³ College of Economics and Management, China Jiliang University, Hangzhou 310018, China

* Correspondence: dujuling@cjlu.edu.cn

Abstract: In order to overcome the difficulty of parameter tuning caused by the large lag and time-varying nonlinearity of the tank for lysozyme fermentation, a temperature control method based on LRWOA-LSTM-PID is proposed in this paper. Firstly, according to the intrinsic mechanism of the fermenter, a temperature mechanism model based on a dynamic equation is designed, which can better reflect the temperature changes in the fermenter. Secondly, a Proportional Integral Derivative (PID) parameter tuning method based on a Long-Short Term Memory Network (LSTM) is proposed, which takes advantage of the ability of LSTM to learn time sequence information and obtains the variation trend between error sequences under continuous time sampling, thereby adjusting network weights more reasonably and accelerating PID parameter tuning. Finally, a Whale Optimization Algorithm (WOA) based on the Lévy flight and random walk strategy (LRWOA) is proposed for the initialization of LSTM parameters; this algorithm has excellent optimization capabilities and overcomes the problem of LSTM falling into local optimal solutions prematurely during parameter randomization. The results show that the method proposed in this paper can achieve rapid tuning of PID parameters, thereby improving the convergence speed of the system and reducing system overshoot.



Citation: Ding, C.; Li, X.; Zhou, H.; Yu, J.; Du, J.; Zhao, S. A Temperature Control Method of Lysozyme Fermentation Based on LRWOA-LSTM-PID. *Processes* **2024**, *12*, 866. <https://doi.org/10.3390/pr12050866>

Academic Editor: Hah Young Yoo

Received: 30 March 2024

Revised: 17 April 2024

Accepted: 23 April 2024

Published: 25 April 2024



Copyright: © 2024 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/>).

Keywords: PID controller; fermentation process; mechanism model; whale optimization algorithm; Lévy flight strategy; random walk strategy

1. Introduction

Lysozyme is a hydrolytic enzyme that possesses the function of decomposing microbial cell walls. It functions in bactericidal, anti-inflammatory, antiviral, hemostatic, and tissue regeneration acceleration [1]. As a non-toxic natural antimicrobial agent, lysozyme plays a vital role in biotechnology, food industry and pharmaceutical formulations. Currently, the main method for large-scale industrial production of lysozyme is through the fermentation of genetically modified microorganisms [2].

The process of lysozyme production has strict requirements for temperature control [3–5]. During the process, due to the lag issue in the refrigerant cooling method of the fermenter system, using traditional control algorithms to control the temperature will cause a large overshoot, which cannot be recovered in a short time. The temperature error will affect the quality of lysozyme. Therefore, researchers have tried many different control algorithms to improve the precision and stability of temperature control.

Among them, the PID control algorithm has been widely used in temperature control. It has been applied in various fields such as controlling the temperature of aircraft cabins, the temperature control areas of 3D bioprinting devices and the environmental temperature

in military equipment development and production [6–9]. Satisfactory results have been obtained in PID temperature control.

Tuning the PID gain quickly and precisely is necessary to make it work in the control system. The traditional parameter tuning method is the Ziegler–Nichols tuning method [10–12], which adjusts the PID parameters based on the open-loop dynamic characteristics of the control system. This method is simple and operable, but it often produces a large overshoot and oscillates back and forth with the response of the controlled system. With the development of technology, some new parameter tuning methods continue to emerge. The Fuzzy PID control algorithm uses fuzzy logic to optimize and tune PID parameters in real time according to certain fuzzy rules [13–16]. However, this method relies on operators' experience to establish fuzzy rules, making it unsuitable for systems that require frequent changes in conditions. The PID parameter tuning method based on the Particle Swarm Optimization (PSO) algorithm [17–20] combines bird foraging behavior, group organization, and optimization mechanisms to better optimize system performance, but it can fall into local optima and premature convergence. The PID parameter tuning method based on the Genetic Algorithm (GA), through a compilation mechanism [21–25], avoids the PID parameter optimization from falling into local optimal solutions. Although the aforementioned meta-heuristic algorithms have strong adaptability and high efficiency [26–29], their nature of work is still based on iteration, and they are usually based on one-time parameter optimization. When the system model changes, it is necessary to re-optimize the parameters. The ability to control temperature in real-time is poor. How to improve the dynamic characteristics and adaptability of the control system has gradually become an urgent engineering problem to be solved.

Artificial Neural Networks (ANN) [30–33] have advantages such as good fault tolerance, high non-linearity, unique associative memory function, and excellent adaptability. They can adaptively handle non-linear dynamic information and perform complex logical operations, and have been widely used in PID parameter tuning. The BP neural network can be used to optimize PID controller parameters [34]. It has been utilized to control camera stabilizers, achieving the precise control of the stable position of the camera. The Dynamic Neural Network (DNN) can be used to tune the gain of the PID controller online [35], which is applied to control pneumatic artificial muscles. This improves the speed and performance of the control process, and features a dynamically self-organizing structure. In [36], a Radial Basis Function (RBF) neural network is used to identify the information of a floating system, and adjust controller parameters in real time. This method exhibits good control effects when dealing with external disturbances and internal parameter changes in maglev systems. The Recurrent Neural Network (RNN) can be used to adapt the PID controller [37], which controls a single degree of freedom pendulum to track the desired reference position with a rapid response and minimal tracking error. This provides excellent position tracking results and robustness to changes in system parameters. In [38], the Diagonal Recurrent Neural Network (DRNN) is used to optimize PID controller parameters. In a multi-point mooring system, the network learns the dynamic changes in the mooring system, minimizing square positioning error, which enhances recognition ability and interference resistance. In order to further improve the optimization ability of neural networks, the PSO algorithm is used to optimize the initial values of the BP neural network, and then optimize the control parameters of PID to achieve the precise control of liquid fertilizer flow rate [39]. The Grey Wolf Optimization algorithm is used to optimize the initial parameters of fuzzy neural networks; then, the fuzzy neural network with the optimal parameters is used to adjust the PID parameters to achieve the smoothness and stability of vehicle control [40]. Although the aforementioned neural networks can quickly tune PID parameters and improve the system's dynamic characteristics and adaptability, they have not effectively utilized target information, historical parameter information, and historical control information. Moreover, the backpropagation algorithm in neural networks has poor global optimization capability and is prone to falling into local optima, limiting the application of neural networks in PID controllers.

To improve the shortcomings of traditional neural network methods in optimizing PID controller parameters, a temperature control method of lysozyme fermentation based on the LRWOA-LSTM-PID method is proposed in this paper. The method first establishes a fermenter temperature mechanism model described using dynamic equations. Then, it utilizes the ability of LSTM to learn time sequence information to capture the trend of error sequences under continuous time sampling. Finally, it initializes LSTM parameters based on the LRWOA algorithm to enhance the stability of the initial parameters of the LSTM, thus endowing the controller with an excellent temperature control capability.

The contributions of the method proposed in this paper are as follows:

- (1) A temperature mechanism model based on a dynamic equation is designed, which can better reflect the temperature changes in the fermenter.
- (2) A Whale Optimization Algorithm based on the Lévy flight and random walk strategy is proposed. By introducing Levy flight and random walk strategies into the Whale Optimization Algorithm, the global and local optimization abilities of the algorithm are improved. It overcomes the problem of the original algorithm falling into local extrema prematurely when dealing with complex issues.
- (3) A PID parameter tuning method based on LSTM is proposed. This method has the ability to learn time sequences information and capture the trend of error sequences under continuous time sampling, so the network weight can be adjusted more reasonably. As a result, it achieves the rapid tuning of the PID parameters K_p , K_i , and K_d .
- (4) An LSTM parameter initialization method based on the LRWOA algorithm is proposed. The method takes advantage of the excellent optimization capability of the LRWOA algorithm, overcoming the problem of traditional random parameter initialization methods falling into local optimum solutions prematurely, thereby enhancing the stability of the initial parameters of the LSTM neural network.

The remainder of this paper is organized as follows: Section 2 introduces the modeling of the lysozyme fermenter temperature system. Section 3 introduces the improved LRWOA algorithm. Section 4 describes the overall network structure and the LRWOA-LSTM-PID algorithm. Section 5 analyzes the experimental results and draws conclusions. Finally, the conclusion is given in Section 6.

2. Modeling of Fermenter Temperature

An appropriate mathematical model can accurately describe the dynamic relationship between input and output variables over time, playing a crucial role in lysozyme fermentation temperature control.

Fermenter Temperature Mathematical Model

The lysozyme fermenter is shown in Figure 1a. The lysozyme fermenter is a non-adiabatic continuously stirred liquid-phase reaction vessel, as shown in Figure 1b.

According to the internal mechanism of the fermenter, the mathematical model of the lysozyme reaction vessel was established using mechanism modeling method in this paper.

- (1) Chemical reaction rate equation

The chemical reaction inside the fermenter is an irreversible exothermic reaction $A \xrightarrow{k_0} B$, and the concentration of yeast C_A and temperature T are essentially uniformly distributed inside the fermenter. From this, r_A for the chemical reaction rate equation of A can be obtained:

$$r_A = \frac{dC_A}{dt} = -k_0 C_A e^{-\frac{E}{RT}} \quad (1)$$

In the equation, A represents the seed of the microorganism; B is lysozyme; E is the activation energy of the reaction (J/mol); k_0 is the reaction frequency factor; R is the gas constant; C_A is the concentration of reactant A (mol/m³).

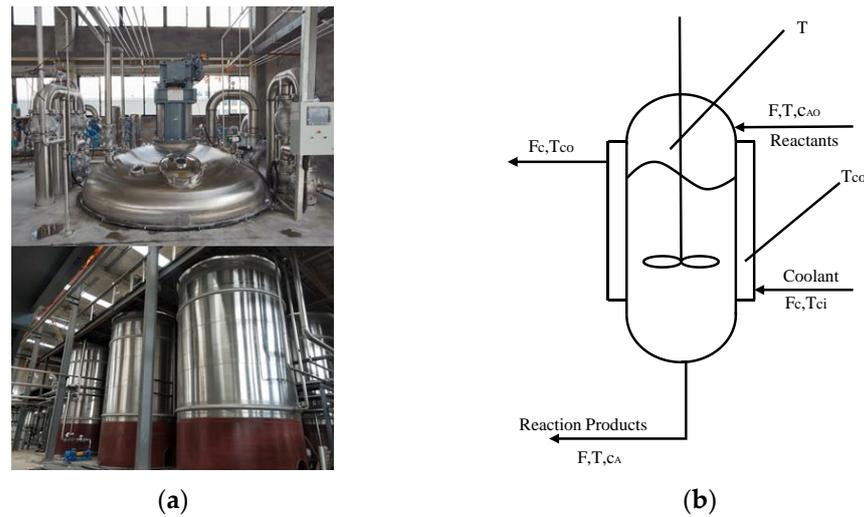


Figure 1. (a) Real lysozyme fermenter. (b) Non-adiabatic continuous reaction vessel.

(2) Material balance equation

The changing rate of cumulant of materials in the fermenter should equal the amount of material entering the fermenter minus the amount leaving, plus or minus the amount consumed or produced using the chemical reaction of the yeast. Given that the stirrer in the fermenter can mix the yeast thoroughly, we can assume complete mixing within the fermenter, making the density, temperature, and yeast molecular concentration equal everywhere.

For the fermenter, the overall material balance equation is:

$$\frac{d(\rho V)}{dt} = \rho_0 F_0 - \rho F \quad (2)$$

In (2), F_0 is the volume flow of yeast (m^3/s); ρ_0 is the yeast density (kg/m^3); F is the volume flow of fermentation products (m^3/s); ρ is the fermentation product density (kg/m^3); V represents the effective volume of fermenter (m^3).

The material balance equation for reactant A is as follows:

$$\frac{dV_{C_A}}{dt} = F_0 C_{A0} - F_{C_A} + V_{r_A} \quad (3)$$

In (3), C_{A0} is the concentration of reactant in yeast (mol/m^3); F_{C_A} stands for the volumetric flow rate of reactant A (m^3/s); V_{r_A} is the volume of reactant A (m^3).

(3) The heat balance equation

From the heat balance relationship, we know:

$$\frac{dH_R}{dt} = Q_1 - Q_2 \pm Q_3 \pm Q_4 \quad (4)$$

In (4), H_R represents the accumulated heat within the fermenter (J); Q_1 is the heat brought into the fermenter by the material per unit time (J/s); Q_2 is the heat taken away by the material leaving the fermenter per unit time (J/s); Q_3 refers to the heat effect brought about by the chemical reaction per unit time (J/s)—if it is an endothermic reaction, the minus sign is taken before Q_3 , if it is exothermic, the plus sign is taken; Q_4 is the heat exchanged between the fermenter and the environment per unit time (J/s). If heat is dissipated, the minus sign is taken before Q_4 , if heat is absorbed from the environment, the plus sign is taken. If the fermenter is adiabatic, Q_4 is zero.

Assuming that the temperature of the cooling liquid in the interlayer is consistent with its outlet temperature T_{co} , ignoring the thermal melt of the inner wall, and assuming that

the specific heat capacity of the feed and the effluent always remain consistent, the heat balance equation can be derived as follows:

$$\frac{dV\rho C_P T}{dt} = F_0\rho_0 C_P T_f - C_P F_P T + K_e A_e (T_{co} - T) - (\Delta H) V_{r_A} \quad (5)$$

In the (5), T_f is the feed temperature during yeast fermentation (K); C_P is the average specific heat capacity of the feed and effluent in the fermentation process (J/(kg·K)); F is the average volumetric flow rate of materials entering and exiting the fermenter (m³/s); ΔH is the reaction heat per unit time (J/s); K_e is the total heat transfer coefficient between the fermenter wall and the material inside the tank (W/(m²·K)); A_e is the heat transfer area (m²); T_{co} is the outlet temperature of the condensate (K).

The heat balance equation for the condensate in the fermenter jacket can also be obtained as follows:

$$\frac{dV_c \rho_c C_{pc}}{dt} = F_c \rho_c C_{pc} (T_{ci} - T_{co}) + K_e A_e (T - T_{co}) \quad (6)$$

In the (6), C_{pc} represents the average specific heat capacity of the condensate (J/(kg·K)); V_c is the volume of the condensate in the jacket (m³); ρ_c represents the density of the condensate (kg/m³); F_c is the volume flow rate of the condensate (m³/s); T_{ci} is the inlet temperature of the condensate (K).

As the volumes in the fermenter and jacket remain constant and the density of the inflowing and outflowing materials is identical, (1) indicates that $F = F_0$. Substituting Equations (1) and (2) into Equations (3), (5) and (6) yields:

$$\frac{dc_A}{dt} = \frac{F}{V} (c_{A0} - c_A) - k_0 c_A e^{-\frac{E}{RT}} \quad (7)$$

$$\frac{dT}{dt} = \frac{F}{V} (T_f - T) + \frac{\Delta H}{\rho C_P} c_A k_0 e^{-\frac{E}{RT}} + \frac{K_e A_e}{V \rho C_P} (T_{co} - T) \quad (8)$$

$$\frac{dT_{co}}{dt} = \frac{F_c}{V_c} (T_{ci} - T_{co}) + \frac{K_e A_e}{V_c \rho_c C_{pc}} (T - T_{co}) \quad (9)$$

The mathematical Equations (7)–(9) established above are nonlinear dynamic equations. To use existing control theory to analyze the dynamic characteristics of the fermenter under small disturbances, the mathematical equations need to be linearized. The state variable of the fermenter temperature system is denoted by $\{c_A \ T \ T_{co}\}$, and the input variable, denoted by $\{F \ c_{A0} \ T_f \ F_c \ T_{ci}\}$, consists of control variables and external disturbances from the actual fermentation process.

The steady-state values of the state and input variables of the fermenter temperature system are $\{\bar{c}_A \ \bar{T} \ \bar{T}_{co} \ \bar{F} \ \bar{c}_{A0} \ \bar{T}_f \ \bar{F}_c \ \bar{T}_{ci}\}$ and the corresponding changes in the input variables are $\{\Delta c_A \ \Delta T \ \Delta T_{co} \ \Delta F \ \Delta c_{A0} \ \Delta T_f \ \Delta F_c \ \Delta T_{ci}\}$. By using differential operations, a linearized model can be obtained. According to (7), the equation is:

$$\frac{d\Delta c_A}{dt} = \frac{\bar{F}}{V} (\Delta c_{A0} - \Delta c_A) + \left(\frac{\bar{c}_{A0}}{V} - \frac{\bar{c}_A}{V} \right) \Delta F - K_T \Delta c_A - K_T \bar{c}_A \frac{E}{RT^2} \Delta T \quad (10)$$

In (10), $K_T = K_0 e^{-\frac{E}{RT}}$; K_0 is the total heat transfer coefficient, (W/(m²·K)); E is the activation energy (J/mol); R is the molar gas constant, ((Pa·m³)/(mol·K)); T is the thermodynamic temperature (K); Δ represents changes; the superscript indicates the vector of each variable.

Similarly, based on Equations (8) and (9), we have:

$$\frac{d\Delta T}{dt} = \frac{\bar{F}}{V} (\Delta T_f - \Delta T) + \left(\frac{\bar{T}_f}{V} - \frac{\bar{T}}{V} \right) \Delta F + \frac{\Delta H}{\rho C_P} K_T \Delta c_A + \frac{\Delta H}{\rho C_P} K_T \bar{c}_A \frac{E}{R \bar{T}^2} \Delta T + \frac{K_e A_e}{V \rho C_P} (\Delta T_{co} - \Delta T) \quad (11)$$

$$\frac{d\Delta T_{co}}{dt} = \frac{\bar{F}_c}{V_c} (\Delta T_{ci} - \Delta T_{co}) + \left(\frac{\bar{T}_{ci}}{V} - \frac{\bar{T}_{co}}{V} \right) \Delta F_c + \frac{K_e A_e}{V_c \rho_c C_{Pc}} (\Delta T - \Delta T_{co}) \quad (12)$$

If we set $Z = \{\Delta c_A \Delta T \Delta T_{co}\}^T$ and $Z = \{\Delta F \Delta c_{A0} \Delta T_f \Delta F_c \Delta T_{ci}\}^T$, we can obtain the following standard linear state equation:

$$\frac{dZ}{dt} = AZ + BU \quad (13a)$$

$$A = \begin{bmatrix} -\left(\frac{\bar{F}}{V} + K_T\right) & -\frac{E}{R \bar{T}^2} K_T \bar{c}_A & 0 \\ K_T \frac{\Delta H}{\rho c_p} & K_T \bar{c}_A \frac{\Delta H}{\rho c_p} \frac{E}{R \bar{T}^2} - \frac{\bar{F}}{V} - \frac{K A_e}{V \rho c_p} & \frac{K A_e}{V \rho c_p} \\ 0 & \frac{K A_e}{V_c \rho_c c_{pc}} & -\left(\frac{\bar{F}_c}{V_c} + \frac{K A_e}{V_c \rho_c c_{pc}}\right) \end{bmatrix} \quad (13b)$$

$$B = \begin{bmatrix} \frac{c_{A0}}{V} - \frac{\bar{c}_A}{V} & \frac{\bar{F}}{V} & 0 & 0 & 0 \\ \frac{\bar{T}_f}{V} - \frac{\bar{T}}{V} & 0 & \frac{\bar{F}}{V} & 0 & 0 \\ 0 & 0 & 0 & \frac{\bar{T}_{ci}}{V_c} - \frac{\bar{T}_{co}}{V_c} & \frac{\bar{F}_c}{V_c} \end{bmatrix} \quad (13c)$$

The temperature state equation established using the above mathematical equation can reflect the dynamic impact of control variables and external disturbances on the output of the fermenter temperature system. This mathematical model can be used to analyze the stability of the fermenter temperature.

First, parameters such as the temperature, density, flow rate, volume of the lysozyme in the fermenter, the volume of the fermenter jacket, and the volume of the condensate must be collected; then, the heat transfer area, heat transfer coefficient, specific heat capacity, condensate density, reaction activation energy, gas constant, and reaction frequency factor based on the parameters of the fermenter and the lysozyme fermentation process is determined. The collected parameters are then analyzed and set, and brought into the established mathematical model to obtain the transfer function designed using the mechanistic modeling. The equation is as follows:

$$G(S) = \frac{4.6e^{-8.6s}}{8s^2 + 8.9s + 2} \quad (14)$$

3. Algorithm

3.1. Whale Optimization Algorithm

WOA [41] is a new swarm intelligence optimization algorithm, which contains three parts: Encircling Prey, Bubble-net attacking and Search Predation.

(1) Encircling Prey

Assume that in a D-dimensional space, the location of the whale individual with the best solution in the current whale population is $\vec{X}^*(t)$, and the location of any other whale individual is $\vec{X}(t)$. Then, the whale individual $\vec{X}(t)$ will change its location under the influence of the best whale individual. The calculation equations for the next location $\vec{X}(t + 1)$ is as follows:

$$\vec{D} = \left| \vec{C} \cdot \vec{X}^*(t) - \vec{X}(t) \right| \quad (15)$$

$$\vec{X}(t + 1) = \vec{X}^*(t) - \vec{A} \cdot \vec{D} \quad (16)$$

$$\vec{C} = 2 \cdot \vec{r} \quad (17)$$

$$\vec{A} = 2\vec{a} \cdot \vec{r} - \vec{a} \quad (18)$$

In the calculation of \vec{D} , the meaning of $|\cdot|$ is to find the absolute value; a gradually decays from 2 to 0 as the number of iterations of the algorithm increases; r is a random vector between $[0, 1]$.

(2) Bubble-net attacking

Bubble-net attacking is a unique hunting method of whales. To mathematically model the behavior of bubble-net behavior, the following two methods are designed:

- a. Contraction enveloping mechanism: This method is roughly the same as enveloping prey, the difference being that the original range of a is adjusted from 2 to 0 to a fixed value of 1. Since a determines the range of A , the range of A changes from the original $[-a, a]$ to now $[-1, 1]$. At this time, the new position of the whale individual can appear at any position between the original position and the optimal solution position.
- b. Spiral updating position: This method first calculates the distance between the whale individual located at (X, Y) and the prey located at (X^*, Y^*) . Then, a spiral equation is established between the two positions, thereby simulating the spiral movement pattern of the whale population. The equation is as follows:

$$\vec{X}(t + 1) = \vec{X}^*(t) + \vec{D}' \cdot e^{bl} \cdot \cos(2\pi l) \quad (19)$$

$$\vec{D}' = \left| \vec{X}^*(t) - \vec{X}(t) \right| \quad (20)$$

In (19), b is the constant for the shape of the logarithmic spiral; l is a random number between -1 and 1 . At this time, the whale will move around the prey in a spiral path within the previous circular area.

To simulate the synchronized behavior of whales, we assume that the probabilities of choosing the contraction enveloping mechanism and the spiral updating position are equal. This makes it convenient to update the position of the whales during the optimization process. The mathematical model is as follows:

$$\vec{X}(t + 1) = \begin{cases} \vec{X}^* - \vec{A} \cdot \vec{D} & \text{if } p < 0.5 \\ \vec{X}^*(t) + \vec{D}' \cdot e^{bl} \cdot \cos(2\pi l) & \text{if } p \geq 0.5 \end{cases} \quad (21)$$

(3) Search Predation

In the mathematical model of the contraction enveloping predation behavior, the range of A_1 is limited to $[-1, 1]$. If the range of A_1 is not within $[-1, 1]$, any whale in the current swarm may not approach the currently optimal whale individual, but instead randomly select one individual from the current whale swarm to approach. Although

search predation can cause a random individual in the current whale swarm to deviate from the target prey, it can enhance the global search ability of the whale swarm. A and C are coefficients. In the D -dimensional space, the position of a random individual in the current whale swarm is \vec{X}_{rand} . The mathematical model of search predation behavior can be described using the following equation:

$$\vec{D} = \left| \vec{C} \cdot \vec{X}_{rand} - \vec{X} \right| \quad (22)$$

$$\vec{X}(t+1) = \vec{X}_{rand} - \vec{A} \cdot \vec{D} \quad (23)$$

3.2. LRWOA Algorithm

The WOA tends to prematurely fall into local optimums when solving complex optimization problems. To solve this issue, this paper proposes a WOA based on Lévy flight and random walk strategies. The Lévy flight strategy is introduced to make individual whales exist widely in the search range and improve the global optimization ability of the algorithm. The random walk strategy is introduced to make individual whales search for the best value in a relatively concentrated range and improves the local optimization ability of the algorithm.

3.2.1. Lévy Flight Strategy

The Lévy flight [42] refers to a random walk where the probability distribution of the step length follows a heavy-tail distribution, which can maximize the search efficiency in uncertain environments. When the spatial dimension of the Lévy flight's random movement is higher than one dimension, the step length distribution shows isotropy. The Lévy flight strategy exhibits excellent global search capability due to its high probability of performing a small area search and a low probability of long-distance flight. The position update equation in the Lévy flight strategy is as follows:

$$x_k^{new}(t) = x_k(t) + l \oplus levy(\lambda) \quad (24)$$

where $levy(\lambda)$ is the step length of the Lévy distribution, which satisfies $levy \sim u = t^{-\lambda}$, $1 < \lambda \leq 3$; \oplus denotes element-wise multiplication; l represents the control step factor, $l = 0.01(x_k(t) - x_b)$; x_b is the optimal solution obtained after t iterations; $x_k(t)$ is the k -th solution of the t -th generation.

The Mantegna method is commonly used to obtain the step length s of the Lévy flight strategy, as shown in (25):

$$s = \frac{\mu}{|v|^{1/\gamma}} \quad (25)$$

where γ is typically 1.5; μ and v follow a normal distribution, $\mu \sim N(0, \sigma_\mu^2)$, $v \sim N(0, \sigma_v^2)$, $\sigma_v = 1$, defined as follows:

$$\sigma_\mu = \left\{ \frac{\Gamma(1+\gamma)\sin(\pi\gamma/2)}{\gamma \cdot \Gamma[(\gamma+1)/2 \cdot 2^{(\gamma-1)/2}]} \right\}^{1/\gamma} \quad (26)$$

Figure 2 shows the walking trajectory of a two-dimensional random walk and a two-dimensional Lévy flight. The difference between the two modes can be seen: the random walk provides roughly the same size for each step, while the Lévy flight performs many small steps with the occasional large step.

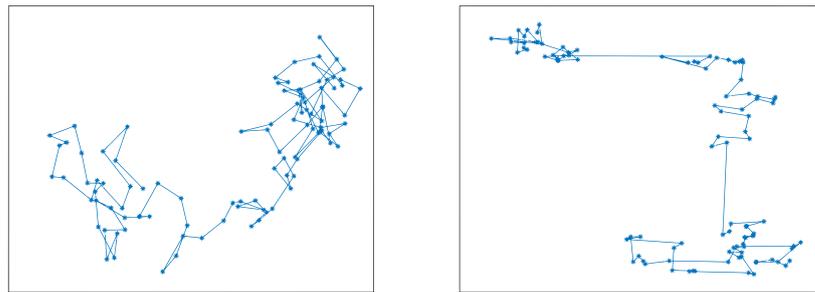


Figure 2. Random walk (left) vs. Levy flight (right).

3.2.2. Random Walk Strategy

The basic idea of random walk [43] is as follows: starting from any individual in a group, traverse the entire group. For any given individual, the probability of moving to an adjacent individual is $1 - a$, and the probability of randomly jumping to any individual in the group is a . The a mentioned above is the probability of the jump occurring. After each jump, the probability distribution can be obtained that reflects the probability of each individual in the group being found. This probability distribution is then used as an input for the next walk and the process is repeated. Under predetermined conditions, the probability distribution tends to converge, and the stable probability distribution is obtained after convergence.

Random walks utilize crossover and hybrid mutation to generate new solutions, which can enhance the diversity of the group and effectively avoid falling into local optimal solutions, thereby improving the local search ability of the target algorithm. Applying a random walk strategy in swarm intelligence algorithms can accelerate the speed of finding optimal solutions. The update equation for the random walk strategy is as follows:

$$x_k^{new}(t) = x_k(t) + \varepsilon(x_j(t) - x_i(t)) \quad (27)$$

where, ε is the scale factor, $\varepsilon \sim U(0, 1)$; $x_j(t)$ and $x_i(t)$ are two random solutions in the t -th generation.

3.2.3. LRWOA

While both the random walk strategy and Levy flight strategy can update individual positions, they cannot guarantee that the fitness of the new solution is greater than the fitness of the original solution. Therefore, we use a Greedy mechanism to ensure the selection of the optimal solution, as shown in the following equation:

$$x_k(t) = \begin{cases} x_k(t), & \text{fit}(x_i'(t)) < \text{fit}(x_k(t)) \\ x_i'(t), & \text{fit}(x_i'(t)) > \text{fit}(x_k(t)) \end{cases} \quad (28)$$

Once the random walk strategy and Levy flight strategy have been established, the Whale Optimization Algorithm (WOA) proceeds with its optimization process as follows: If a randomly selected individual's value is equal to the best, second best, or third best individual value and is less than $0.5 - 0.5t/(\varphi * \text{max})$, the whale individual will use the Levy flight strategy to update its position. On the other hand, if the value of a randomly selected individual matches the best, second best, or third best individual value and is greater than or equal to $0.5 - 0.5t/(\varphi * \text{max})$, the whale individual will choose the random walk strategy to update its position. Otherwise, the whale individual will use the original WOA strategy to update its position. t represents the current iteration within the process, max denotes the maximum number of iterations, and φ refers to the exploration iteration ratio. The algorithm flow as shown in the Algorithm LRWOA (Algorithm 1).

Algorithm 1. LRWOA

Input: Number of whales N , maximum number of iterations max , iteration ratio φ
Output: X_α
Start:
1: Initialize the whale population X_k ($k = 1, 2, \dots, N$) in the search space.
2: Calculate the fitness values of the whale population individuals and select the best individual X_α , the second best individual X_β , and the third best individual X_δ .
3: Select a random individual $rand$ and record its position.
4: while $t < max$
5: for $k = 1$ to N perform
6: Calculate the values of parameters C and A according to $\vec{C} = 2 \cdot \vec{r}$ and $\vec{A} = 2\vec{a} \cdot \vec{r} - \vec{a}$
7: if ($X = X_\alpha \mid X = X_\beta \mid X = X_\delta$) perform
8: if ($rand < 0.5 - 0.5t / (\varphi * max)$) perform
9: Adjust the positions by $x_k^{new}(t) = x_k(t) + l \oplus levy(\lambda)$.
10: else
11: Adjust the positions by $x_k^{new}(t) = x_k(t) + \varepsilon(x_j(t) - x_i(t))$.
12: end if
13: Select the better positions based on $x_k(t) = \begin{cases} x_k(t), & fit(x_i'(t)) < fit(x_k(t)) \\ x_i'(t), & fit(x_i'(t)) > fit(x_k(t)) \end{cases}$.
14: end if
15: Update the individuals' positions based on Equations (15) to (23).
16: end for
17: Calculate the fitness of individuals, update whale positions, and record the positions.
18: $t = t + 1$
19: end while

3.2.4. Results and Analysis

To evaluate the performance of the LRWOA algorithm, this study selects 15 internationally recognized benchmark functions as the basis for the experimental tests. These benchmark functions vary in size and complexity but are all designed for minimization problems. The benchmark functions can be classified into single-peak and multi-peak functions. The single-peak functions are used to assess the algorithm's exploitation capability, while the multi-peak functions are used to assess the algorithm's exploration capability. Table 1 provides the names, search ranges, function indices, and adaptive values of the 15 benchmark functions.

Table 1. Types, names, ranges, and adaptive values of international test functions.

Function Type	Index	Function Name	Search Range	Adaptive Value
single-peak functions	F1	Sphere Function	[−100, 100]	0
	F2	Schwefel's Problem 1.2	[−100, 100]	0
	F3	Schwefel's Problem 1.2 With Noise	[−32, 32]	0
	F4	Schwefel's Problem 2.21	[−5, 5]	0
	F5	Schwefel's Problem 2.22	[−10, 10]	0
	F6	High Conditioned Elliptic Function	[−3, 1]	0
	F7	Step Function	[−10, 10]	0
multi-peak functions	F8	Schwefel Function	[−500, 500]	0
	F9	Rosenbrock's Function	[−10, 10]	0
	F10	Quartic Function	[2, 5]	0
	F11	Griewank's Function	[−600, 600]	0
	F12	Ackley's Function	[−32, 32]	0
	F13	Rastrigin's Function	[−5.12, 5.12]	0
	F14	Rastrigin's Noncontinue Function	[−5.12, 5.12]	0
	F15	Weierstrass Function	[−100, 100]	0

The LRWOA is compared with the WOA, PSO [44], Grey Wolf Optimizer (GWO) [45], and GWO enhanced using the Lévy-flight strategy (LGWO) [46] through experiments.

To ensure fairness and impartiality, considering the instability of the algorithms, each algorithm experiment is run independently 10 times, and the average results are taken. In the comparative experiments of different algorithms, the population size is set to 50, and the dimension of the decision variables is set to 60. Figures 3–6 shows the convergence curves of some functions, where F1 and F8 represent unimodal functions, and F10 and F14 represent multimodal functions.

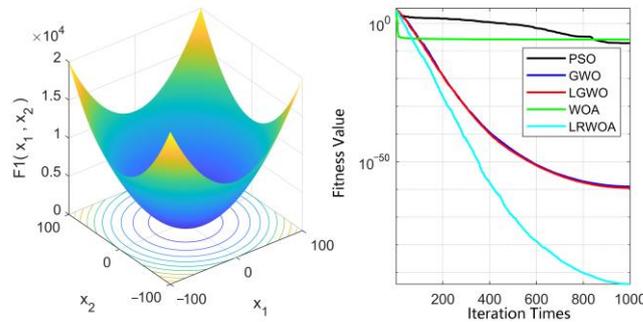


Figure 3. The convergence curves of five algorithms in optimizing the F1 benchmark function.

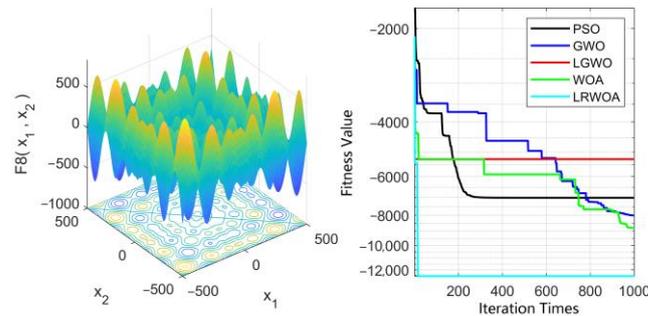


Figure 4. The convergence curves of five algorithms in optimizing the F8 benchmark function.

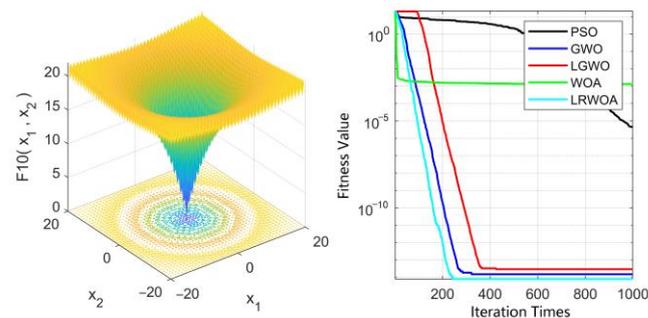


Figure 5. The convergence curves of five algorithms in optimizing the F10 benchmark function.

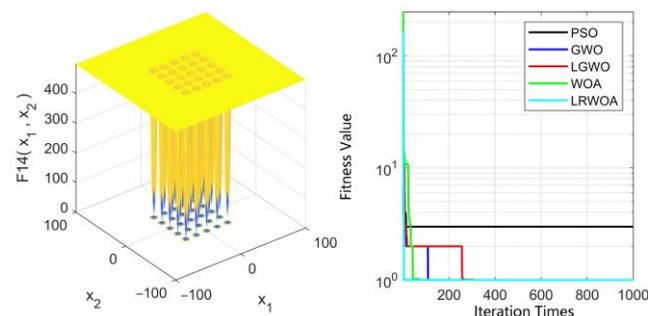


Figure 6. The convergence curves of five algorithms in optimizing the F14 benchmark function.

Upon comprehensively analyzing the convergence curves in Figures 3–6, it can be observed that the LRWOA converges rapidly on both unimodal and multimodal functions, and its convergence precision is also improved. On the unimodal functions F1 and F8, the PSO, GWO, LGWO, and WOA all fall into local optima in the early stages of the search process, indicating a weaker local search capability. Among these, the PSO performs less effectively in local search capabilities than the other four and does not exhibit a strong performance in global optimization. The LRWOA, aided by the random walk strategy, can ensure the rapid convergence and the capability of escaping local optima, manifested specifically in the steep declines on the convergence curves. On the multimodal functions F10 and F14, compared to the other four algorithms, the LRWOA demonstrates a faster drop and closer approach to the function's adaptive values on the convergence curve graph, while also avoiding stagnation points that are easily entrapped.

This indicates that the LRWOA has better global exploration capabilities with the assistance of the Lévy flight strategy. The above analysis indicates that, compared with the other four algorithms, the LRWOA excels in comprehensive performance, efficiently solving various problems.

4. LRWOA-LSTM-PID Controller

This section sequentially introduces the PID controller, the LSTM-PID controller model, and the LSTM-PID controller based on the LRWOA.

Figure 7 illustrates the overall structure diagram of the control system, which contains four modules: the LRWOA module, the LSTM module, the PID controller module, and the fermenter temperature system module. The LRWOA provides optimized initial parameters for the LSTM, thereby reducing the possibility of the network falling into local optima to some extent. The LSTM online tunes parameters for the PID controller, strengthening the connection between control information and thus accelerating the temperature control speed and stability; the PID controller still plays a leading role in the control process. It controls the actuator by outputting control quantities, thereby controlling the fermenter temperature system. The model of the fermenter temperature system in this paper is shown in (14).

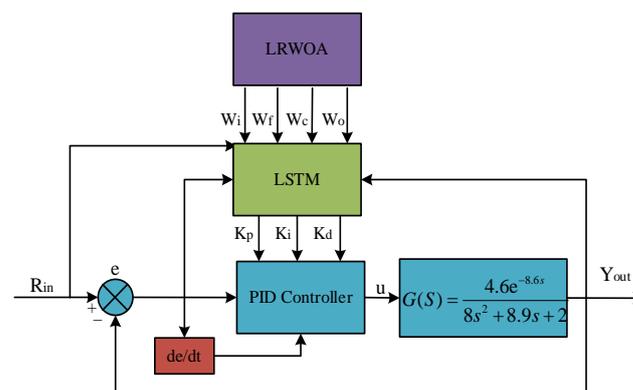


Figure 7. The structure of LRWOA-LSTM-PID controller.

4.1. PID Controller

The control law for the PID controller as shown in (29). In (29): t represents the sample number, $u(t)$ is the computed output value at the t -th sampling instant, and $e(t)$ is the deviation value input at the t -th sampling instant. The working principle of PID is to perform proportional calculation, differential calculation, and integral calculation on the

difference value $e(t)$, and add these three calculation results to obtain the control quantity $u(t)$ of the PID controller.

$$u(t) = K_p * e(t) + K_i \sum_{j=0}^t e(j) + K_d [e(t) - e(t-1)] \quad (29)$$

After determining the expression of $u(t)$, using $u(t) - u(t-1)$ can obtain the increment of the control quantity. The resulting mathematical expression is as follows:

$$\Delta u(t) = (k_p + k_i + k_d)e(t) - (k_p + 2k_d)e(t-1) + k_d e(t-2) \quad (30)$$

In (30), k_p , k_i , and k_d are the proportional coefficient, integral coefficient, and derivative coefficient, respectively; $\Delta u(t)$ is the difference between the controller output at time t and the previous moment; $e(t)$ is the difference between the target value of lysozyme fermentation temperature $r(t)$ at time t and the output value of the fermenter temperature model $y(t)$.

4.2. LSTM-PID Controller

Control systems designed using neural network methods have stronger real-time performance, robustness, and adaptability. However, the BP-PID controller designed using the BP does not effectively utilize target information, previous parameter information, previous control information, etc. Additionally, the global optimization capability of the backpropagation learning algorithm is relatively poor, making it easy to fall into local optima, which fails to achieve good control effects and limits the application of neural networks in PID controllers.

To solve these issues, a PID controller based on the LSTM is designed. An ordinary RNN simply stacks past information, which can easily lead to problems with gradient explosion and gradient vanishing during training. In contrast, an LSTM can control the transmission state through a gating system, remembering information that needs to be stored long-term and ignoring less relevant information. The structure of the LSTM-PID controller designed in this paper is shown in Figure 8. The inputs to the LSTM are set as the target value of the current time point for lysozyme fermentation temperature, the output value of the current time point for the lysozyme fermentation temperature model, and the error values of these two factors for the three time points before and after; the output of the neural network is set as the three parameter values of the PID controller, k_p , k_i , and k_d .

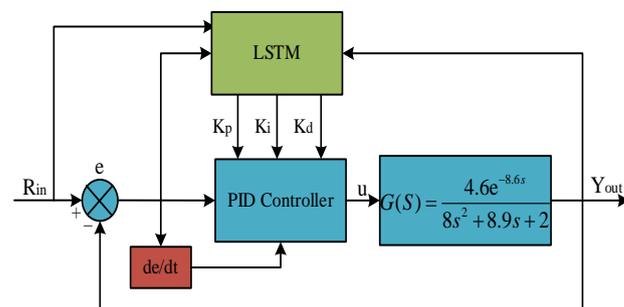


Figure 8. System structure diagram based on LSTM- PID controller.

The structure diagram of the LSTM-PID controller is shown in Figure 9, where:

$$\vec{x} = \begin{cases} h(t-1) \\ c(t-1) \\ x(t) \end{cases} \quad (31)$$

$h(t - 1)$ represents the output value of the network at the previous moment; $c(t - 1)$ denotes the memory unit at the previous moment. $x(t)$ is the current input as a one-dimensional vector.

$$\begin{cases} x_1(t) = r(t) \\ x_2(t) = y(t) \\ x_3(t) = e(t) \\ x_4(t) = e(t - 1) \\ x_5(t) = e(t - 2) \end{cases} \quad (32)$$

where $r(t)$ is the target temperature value for lysozyme fermentation; $y(t)$ is the output temperature value of the fermenter system; $e(t)$ is the difference between $r(t)$ and $y(t)$ at time t .

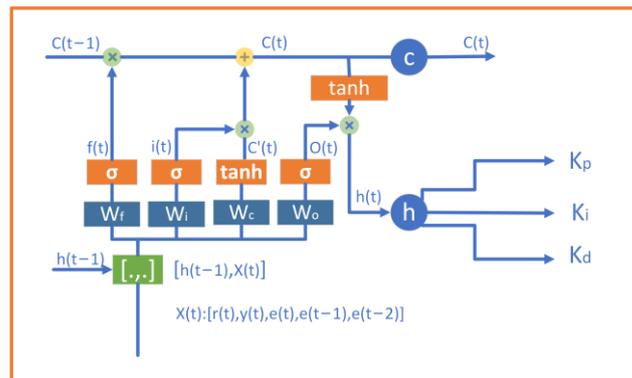


Figure 9. Structure diagram of the LSTM-PID controller.

4.2.1. The Forward Propagation of LSTM

Neural network training is divided into forward propagation and backpropagation. Forward propagation starts from the input layer and continuously calculates the results obtained using each layer of the neural network and the output results through the activation function, finally obtaining the output process. The forward propagation of the LSTM involves the computation of output h and cell state c . The gate system inputs a vector and outputs a value between 0 and 1, allowing useful information flow to pass and blocking useless information flow. Assuming that the weight vector and bias of the gate are represented by W and b respectively, then the gate can be represented by (33):

$$g(\vec{x}) = \sigma(W\vec{x} + b) \quad (33)$$

where, σ is the sigmoid function $\sigma(z) = \frac{1}{1+e^{-z}}$.

The use of gates involves element-wise multiplication of the output vector by the vector to be controlled. Given that the output of the gate is a real vector between 0 and 1, when the gate output is 1, any vector multiplied by it will not change, meaning all information can pass through; when the output is 0, any vector multiplied by it will result in a zero vector, meaning no data can be passed through. As the value range of the sigmoid function is (0, 1), the state of the gate is always half-open and half-closed. The input gate and forget gate in the gating system can be represented by Equations (34) and (35), respectively.

$$i(t) = \sigma(W_i \cdot [h(t - 1), x(t)] + b_i) \quad (34)$$

$$f(t) = \sigma(W_f \cdot [h(t - 1), x(t)] + b_f) \quad (35)$$

where, W_i is the weight matrix of the input gate, $[h(t - 1), x(t)]$ represents concatenating two vectors into a longer vector, b_i is the bias term of the input gate, W_f is the weight matrix of the forget gate, and b_f is the bias term of the forget gate.

The current time input cell state $\tilde{c}(t)$ is calculated from the previous moment's output and the current moment's input, as shown in (36).

$$\tilde{c}(t) = \tanh(W_c \cdot [h(t-1), x(t)] + b_c) \quad (36)$$

In the (36), W_c is the weight matrix of the cell state, and b_c is the bias term of the cell state. The current time cell state $c(t)$ can be obtained by multiplying the forget gate $f(t)$ and the cell state $c(t-1)$ of the last moment element by element, then multiplying the input gate $i(t)$ and the cell state $\tilde{c}(t)$ of the current moment input element by element, and finally adding the two results, the equation is as follows:

$$c(t) = f(t) * c(t-1) + i(t) * \tilde{c}(t) \quad (37)$$

In the (37), the symbol $*$ indicates that the variables are multiplied element by element. It can be seen that the new cell state $c(t)$ is obtained by combining the memory $\tilde{c}(t)$ of the current moment of the LSTM and the memory $c(t-1)$ of the previous moment. Due to the control of the input gate and the forget gate, it can prevent unimportant information from entering the memory and can save previous information. Finally, the output gate is used to control the influence of the LSTM on the current moment output, and the output gate equation is as follows:

$$o(t) = \sigma(W_o \cdot [h(t-1), x(t)] + b_o) \quad (38)$$

In (38), W_o is the weight matrix of the output gate, and b_o is the bias term of the output gate. The final output of the LSTM is determined using the cell state and the output gate. The output is shown in (39). In this paper, the output $h(t)$ of the LSTM neural network is the three parameters of PID: k_p , k_i , and k_d .

$$\begin{cases} h_1(t) = o_1(t) * \tanh[c_1(t)] = k_p \\ h_2(t) = o_2(t) * \tanh[c_2(t)] = k_i \\ h_3(t) = o_3(t) * \tanh[c_3(t)] = k_d \end{cases} \quad (39)$$

4.2.2. The Backpropagation of the LSTM

The loss function is obtained based on the results of forward propagation and input into the backpropagation process to train the neural network. In this section, the backpropagation algorithm used to train the LSTM mainly consists of the following three steps: First, compute the output values of all neurons in the network, namely the five vectors $i(t)$, $f(t)$, $o(t)$, $c(t)$, and $h(t)$. Then, calculate the error values of all neurons in the network in reverse; the backpropagation of the error of the LSTM includes two directions, namely the propagation to the previous layer and the propagation along the time direction. Finally, calculate the gradients of each weight according to the relevant error values.

The error of the LSTM is backpropagated over time, which is a common method for training recursive neural networks. The central idea of this method is to constantly find suitable values along the negative gradient direction of the parameters to be optimized until the model converges. Before performing backpropagation, the error $\delta(t)$ at time t is defined as follows:

$$\delta(t) = \frac{\partial E}{\partial h(t)} \quad (40)$$

To propagate the error back in time, it is necessary to calculate the error $\delta(t-1)$ at time $t-1$. This can be achieved using (40):

$$\delta^T(t-1) = \frac{\partial E}{\partial h(t-1)} = \frac{\partial E}{\partial h(t)} \cdot \frac{\partial h(t)}{\partial h(t-1)} = \delta^T(t) \frac{\partial h(t)}{\partial h(t-1)} \quad (41)$$

By expanding (41) and utilizing the total derivative formula along with Equations (37) and (39), we can derive the formula for error propagation to any time k :

$$\delta^T(k) = \prod_{j=k}^{t-1} \delta_o^T(j)W_{oh} + \delta_f^T(j)W_{fh} + \delta_i^T(j)W_{ih} + \delta_c^T(j)W_{ch} \quad (42)$$

In (42), $\delta_*(t)$ represents the error at moment j for * (* stands for the input gate, output gate, forget gate, and cell state), and $W_{\alpha,\beta}$ represents the weight between α and β .

In the backpropagation of error terms in the LSTM, it is assumed that the current layer is the l th layer. The error term of the $l - 1$ st layer is defined as the derivative of the error function with respect to the weighted input of the $l - 1$ st layer, which is:

$$\delta_t^{l-1} = \frac{\partial E}{\text{net}_t^{l-1}} \quad (43)$$

When calculating the gradients of each weight, for the weight gradients of W_{oh} , W_{fh} , W_{ih} , and W_{ch} , they are actually the sum of the gradients at each moment. Therefore, it is first necessary to find out their gradients at the moment t , and then calculate their final gradients. The error terms δ_o , δ_f , δ_i and δ_c have been obtained in the previous (42), so the gradient values at the moment t can be obtained as follows:

$$\begin{cases} \frac{\partial E}{\partial W_{fh,t}} = \frac{\partial E}{\partial \text{net}_{f,t}} \cdot \frac{\partial \text{net}_{f,t}}{\partial W_{fh,t}} = \delta_{f,t} h_{t-1}^T \\ \frac{\partial E}{\partial W_{oh,t}} = \frac{\partial E}{\partial \text{net}_{o,t}} \cdot \frac{\partial \text{net}_{o,t}}{\partial W_{oh,t}} = \delta_{o,t} h_{t-1}^T \\ \frac{\partial E}{\partial W_{ih,t}} = \frac{\partial E}{\partial \text{net}_{i,t}} \cdot \frac{\partial \text{net}_{i,t}}{\partial W_{ih,t}} = \delta_{i,t} h_{t-1}^T \\ \frac{\partial E}{\partial W_{ch,t}} = \frac{\partial E}{\partial \text{net}_{c,t}} \cdot \frac{\partial \text{net}_{c,t}}{\partial W_{ch,t}} = \delta_{c,t} h_{t-1}^T \end{cases} \quad (44)$$

By summing up the gradients at each moment, the final gradients can be obtained:

$$\begin{cases} \frac{\partial E}{\partial W_{oh}} = \sum_{j=1}^t \delta_{o,j} h_{j-1}^T \\ \frac{\partial E}{\partial W_{fh}} = \sum_{j=1}^t \delta_{f,j} h_{j-1}^T \\ \frac{\partial E}{\partial W_{ih}} = \sum_{j=1}^t \delta_{i,j} h_{j-1}^T \\ \frac{\partial E}{\partial W_{ch}} = \sum_{j=1}^t \delta_{c,j} h_{j-1}^T \end{cases} \quad (45)$$

The following can be obtained based on the corresponding error terms:

$$\frac{\partial E}{\partial W_{ij}} = \frac{\partial E}{\partial \text{net}} \cdot \frac{\partial \text{net}}{\partial W_{ij}} \quad (46)$$

In (46), W_{ij} represents the weight of the i th neuron in the hidden layer and the j th neuron in the output layer; net is the input to the output layer.

However, after setting the output of the neural network as the three parameters of the controller, since the desired PID controller target parameters are uncertain, the external environment cannot provide clear target values for the neural network to learn. Therefore, this paper redesigns the backpropagation algorithm of the LSTM in combination with the formula of the PID controller and sets the difference between the temperature to be controlled in the fermenter and the actual temperature in the fermenter as the new error

value of the neural network. As shown in (47), a new factor is integrated when solving, which is:

$$\frac{\partial E}{\partial W_{ij}} = \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial \Delta u} \cdot \frac{\partial \Delta u}{\partial h} \cdot \frac{\partial h}{\partial net} \cdot \frac{\partial net}{\partial W_{ij}} \quad (47)$$

In (47), E is the mean square error function; y is the actual measured value of the system; Δu is the output of the controller; h is the output of the neural network; since $\partial y / \partial \Delta u$ is unknown, it can be approximately replaced by the sign function sgn which leads to the inaccuracy of the calculation and can be compensated by adjusting the learning rate.

4.2.3. PID Parameter Tuning Based on LSTM

The design of the neural network includes the error calculation and the way of error backpropagation. It leverages the LSTM to optimize the k_p , k_i , and k_d parameters of the PID controller.

The main process of this controller during control is as follows:

Step 1: Build the PID control algorithm and design the LSTM, combine the PID control algorithm and the LSTM, determine the learning rate η and the inertia coefficient α of the LSTM, and set the initial time t to 1.

Step 2: By setting the target temperature $r(t)$ and the actual temperature $y(t)$ calculated using the mathematical model, the error $e(t)$ at this moment is obtained. The calculation formula of $e(t)$ is $e(t) = r(t) - y(t)$.

Step 3: Take $r(t)$, $y(t)$, $e(t)$, $e(t - 1)$, $e(t - 2)$ as the inputs of the LSTM, and set the initial values $r(0)$, $y(0)$, $e(0)$, $e(-1)$, $e(-2)$ to 0.

Step 4: Set the output of the LSTM as the three adjustable parameters of the PID controller.

Step 5: Calculate the output $u(t)$ of the PID controller:

$$\begin{cases} u(t) = u(t - 1) + \Delta u(t) \\ \Delta u(t) = k_p(e(t) - e(t - 1)) + k_i e(t) + k_d(e(t) - 2e(t - 1) + e(t - 2)) \end{cases} \quad (48)$$

Step 6: Set the difference $e(t)$ between $y(t)$ and $r(t)$ as the error of the LSTM, carry out the learning of the LSTM, adjust the weight parameters of the neural network online, and realize the adaptive adjustment of the controller parameters.

Step 7: When $e(t)$ is less than the set threshold, stop adjusting the controller parameters, determine the final controller parameters and input them into the PID controller to obtain the control parameters, otherwise let $t = t + 1$ and return to step 2.

4.3. LSTM-PID Controller Based on LRWOA Algorithm

Although the PID controller based on the LSTM has improved the shortcomings of the classical PID controller, there are still some shortcomings in the training process of the LSTM. For example, when using the backpropagation algorithm to optimize the neural network during training, there is a certain probability that the control system will diverge and not converge. In response to this situation, this article introduces inertia terms into the backpropagation algorithm to alleviate the oscillation problem. In addition, because the initial parameter values of the neural network are initialized in a randomized manner during training, this may cause the neural network to fall into a local optimal solution, resulting in poor results. If suitable parameters can be found for the neural network in the early stages of neural network training, the above problems can be effectively improved. Using the excellent optimization ability of the LRWOA algorithm can find more appropriate parameters in the early stages of neural network training, which can reduce the divergence and non-convergence problems of neural network training.

Using the LRWOA algorithm to solve the problem of initial weights in LSTM has the following two key points: First, the dimension values of the whales in the LRWOA algorithm should be equal to the number of connection weights in the LSTM. Each position of the whale in the LRWOA is set to an initial weight in the neural network. Second, the

fitness function in the LRWOA algorithm needs to be replaced with the mean square error function in the LSTM.

Figure 10 provides an explanation in a two-dimensional space for (16). The position (X, Y) of the whale individual can be updated according to the position (X^*, Y^*) of the whale individual currently obtaining the best solution. By adjusting the values of A and C , the whale individual can be updated to different positions around the current best solution. The possible update positions in three-dimensional space are shown in Figure 11.

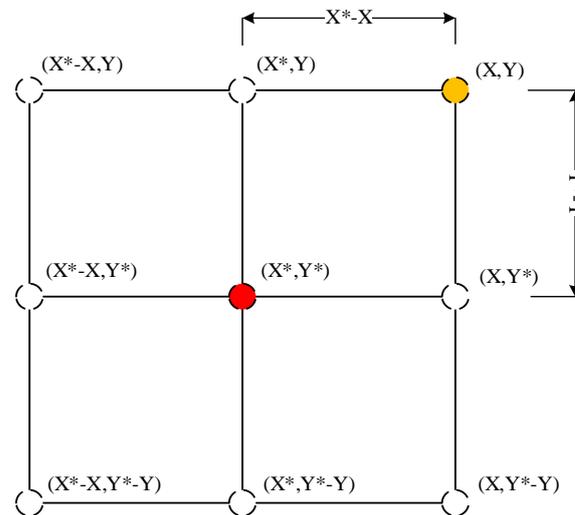


Figure 10. Position vector in two-dimensional space and its possible next position (* represents the current best solution).

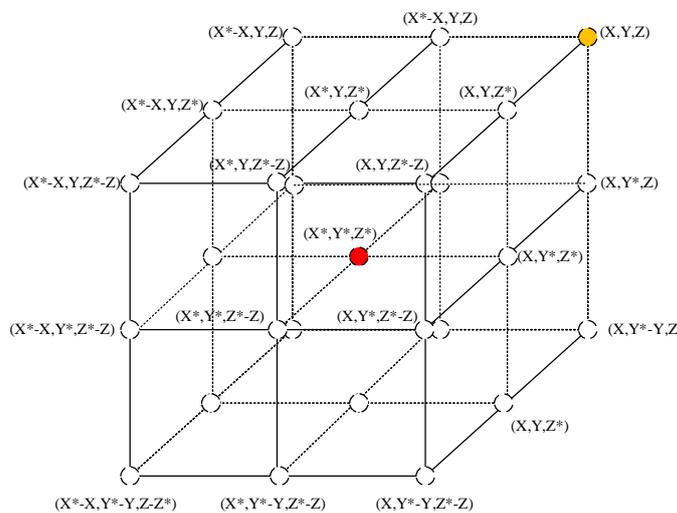


Figure 11. Position vector in three-dimensional space and its possible next position (* represents the current best solution).

By defining a vector r , the individual can be updated to any position around the best solution in Figure 11, therefore, (16) can simulate the behavior of surrounding prey. The above knowledge can also be applied to the n -dimensional search space. Whale individuals will move around the currently obtained best solution in the n -dimensional space. Therefore, when applied to the optimization of initial parameters of LSTM, it is necessary to set the Whale Optimization Algorithm as a four-dimensional search space, corresponding to W_i, W_f, W_c and W_o of the LSTM network, respectively.

The workflow of the optimized controller is as follows:

Step 1: Firstly, determine the topology of the LSTM to find out the number of parameters in each layer of the network and the number of network hyperparameters;

Step 2: Set the dimension of the LRWOA algorithm to the number of parameters in step 1 and randomly initialize the position of the whale population;

Step 3: Use the error between the target value and the output value as the fitness function. According to the fitness function, judge whether the current position of the whale meets the conditions. If it meets the conditions, output the best position of the whale individual in the whale population and jump to step 7, otherwise, jump to step 4;

Step 4: Calculate the objective function value of the whale population;

Step 5: Update the positions of all whale individuals in the whale population;

Step 6: Update the objective function value of the whale population;

Step 7: Use the obtained output result as the optimal initial parameter of the neural network;

Step 8: Set the target temperature $r(t)$ and obtain the actual temperature $y(t)$ through measurement sampling to obtain the current error $e(t)$. The calculation equation of $e(t)$ is $e(t) = r(t) - y(t)$;

Step 9: Use $r(t), y(t), e(t), e(t-1), e(t-2)$ as the input of the LSTM neural network;

Step 10: Set the output of the LSTM as the three adjustable parameters of the PID controller;

Step 11: Calculate the output of the PID controller, set the difference $e(t)$ between $y(t)$ and $r(t)$ as the error of the LSTM, conduct LSTM learning, adjust the weight parameters of the neural network online, and realize the self-adaptive adjustment of the controller parameters;

Step 12: When $e(t)$ is less than the set threshold, stop adjusting the controller parameters, determine the final controller parameters, and obtain the control parameters into the PID controller, otherwise, let $t = t + 1$ and return to step 8.

5. Experiment

5.1. Parameter Description

According to the parameters of the lysozyme fermentation process provided by the project cooperation unit, the relationship between temperature and lysozyme activity is shown in Figure 12. As can be seen from Figure 12, the yield of lysozyme shows a trend of first increasing and then decreasing with temperature. This is because the lysozyme is not active enough at low temperatures, while higher temperatures will cause the enzyme to lose activity. From the figure, it can be seen that the enzyme activity reaches the highest point when the temperature is 33 °C. Therefore, the best temperature in the process of lysozyme fermentation is set as 33 °C. The subsequent experiments will use this temperature as the control target of the control system.

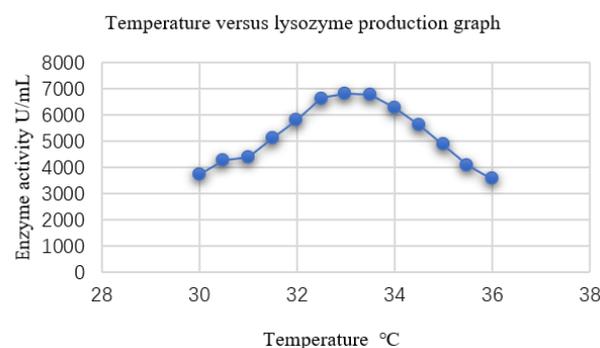


Figure 12. Temperature and lysozyme yield relationship diagram.

The input data of the control system are different from the dataset of traditional neural networks. The input of the neural network in this paper is a time series sequence. The

specific input is as follows: the output sequence of the control system at the current moment, the system target value, and the error sequence of the two in the previous three moments.

In [6,34], PID and BP-PID have been widely applied in the industry and are the two most representative controllers. In [17–22,39], heuristic algorithms are used to optimize the initial parameters of the BP neural network, among which PSO-BP-PID is the most representative. In [37], the method of using the RNN is employed to optimize the PID parameters, which demonstrates excellent position tracking results and robustness against variations in system parameters.

For this purpose, we chose PID [6], BP-PID [34], PSO-BP-PID [39] and RNN-BP-PID [37] as the comparison methods for the experiment.

The control system in this paper is the temperature control system of the fermenter, and its system function is shown in (14) in Chapter 2. The parameters analyzed in the comparative experiment are explained in Figure 13.

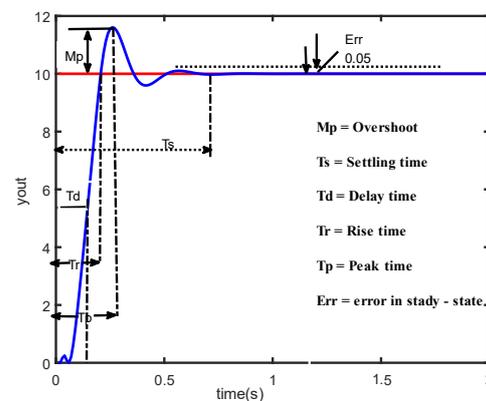


Figure 13. PID experiment parameter explanation diagram.

In Figure 13, M_p represents the overshoot generated by the system; T_s represents the system's settling time; T_d represents the system's delay time; T_r represents the system's rise time; T_p represents the system's peak time; Err represents the system's error value.

5.2. Experimental Results and Analysis

Comparative experiments were conducted on the five controllers, and Figure 14 shows the control performance of the classic PID controller, BP-PID controller, PSO-BP-PID controller, RNN-BP-PID controller, and LSTM-PID controller. R_{in} represents the target temperature. These controllers can complete the control of the fermenter temperature in a short time; the control duration can be kept within 1 s and the overshoot size can be controlled within 2 °C. Among them, the classic PID controller, BP-PID, and PSO-BP-PID controllers have very similar responses, showing an upward and downward fluctuation and gradually stabilizing trend, while the RNN-BP-PID and LSTM-PID controllers stabilize slowly after continuously rising to the target temperature.

In terms of the T_d parameter, the five controllers are basically consistent, thanks to the excellent performance of the PID controller itself. In terms of the T_r and T_p parameters, PID, BP-PID, and PSO-BP-PID show an increasing relationship in sequence, while RNN-BP-PID and LSTM-PID do not produce overshoot due to the addition of recurrent neural network, so that their rising time, peak time and stable time are consistent. In terms of the T_s parameter, it can be observed that LSTM-PID performs best, reaching stability at 0.61 s. The control duration of the classic PID controller is significantly longer than the other four controllers, gradually stabilizing around 1 s. This is because the parameters of the classic PID are fixed and cannot change with the environment. The performances of the other three controllers are comparable, stabilizing the target temperature almost at the same time. In terms of the M_p parameter, controllers with the recurrent neural network perform better than other controllers. Specifically, classic PID, BP-PID, and PSO-BP-PID all have

an overshoot, which are 1.62 °C, 0.79 °C, and 0.1 °C, respectively. However, the recurrent neural network, thanks to its stronger fitting ability and unique associative memory ability, therefore has no overshoot in RNN-BP-PID and LSTM-PID in this system.

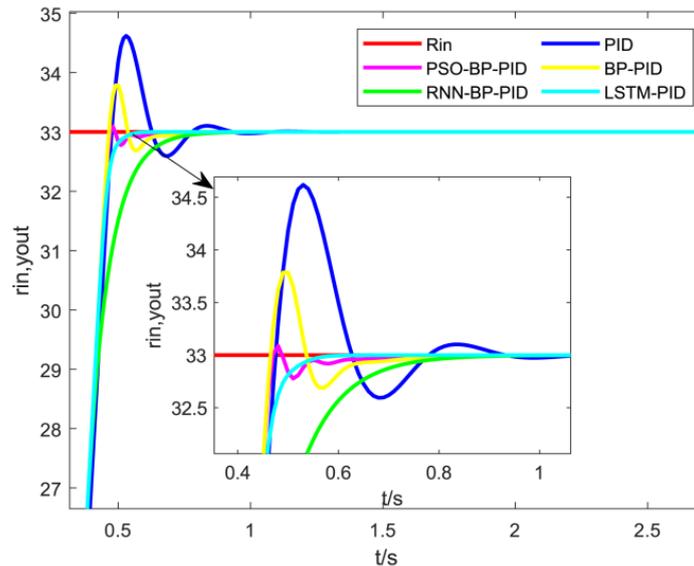


Figure 14. Controller correlation diagram at 33 °C.

Table 2 shows the specific parameters of the five controllers during the control process. It can be seen that LSTM-PID and RNN-BP-PID have the most outstanding performance in terms of overshoot, with no overshoot observed. In terms of settling time, LSTM-PID performs best, reducing by 42.5% compared to the traditional PID, 17.6% compared to BP-PID, 15.3% compared to PSO-BP-PID, and 20.8% compared to RNN-BP-PID. However, in terms of rise time and peak time, since LSTM-PID does not produce overshoot, both of these times are equivalent to the settling time, so they have no comparative significance. From the above analysis, it is clear that LSTM-PID has the best performance.

Table 2. Controller comparison table at 33 °C. Bold represents the best result.

CONTROLLER	PID	BP-PID	PSO-BP-PID	RNN-BP-PID	LSTM-PID
Rise time (s)	0.48	0.46	0.47	0.77	0.61
Settling time (s)	1.06	0.74	0.72	0.77	0.61
Overshoot (°C)	1.62	0.79	0.10	0.00	0.00
Peak (°C)	34.62	33.79	33.10	33.00	33.00
Peak time (s)	0.53	0.50	0.48	0.77	0.61
K_p	4.08	9.91	2.51	14.82	10.01
K_i	0.87	1.92	1.72	1.8	2.09
K_d	12.00	9.46	10.03	10.4	10.06
Error	0	0	0	0	0

To verify the effectiveness and optimization ability of the LRWOA algorithm, a comparative experiment was carried out between the LSTM-PID controller based on the LRWOA algorithm and the PID controller based on the LSTM. The results are shown in Figure 15.

From Figure 15, it can be seen that, compared to LSTM-PID, LRWOA-LSTM-PID can reach the target temperature more quickly during the control process. Overall, LRWOA-LSTM-PID shows stable improvements in rise time, peak time, and settling time compared to LSTM-PID. Specifically, in terms of rise time, peak time, and settling time, the controller designed in this section benefits from the inclusion of the strong optimization ability of the LRWOA algorithm, which can quickly find suitable initial parameter values for the LSTM, thus reducing by 9.8% compared to the LSTM-PID controller. In terms of overshoot, both

LRWOA-LSTM-PID and LSTM-PID benefit from the cyclic neural information feedback mechanism, so both show excellent performance and do not produce overshoot in this system. In terms of controller parameters, both LRWOA-LSTM-PID and LSTM-PID can quickly find a reasonable parameter combination for the PID controller, enabling it to respond quickly to the control system. The specific parameters of the results are shown in Table 3.

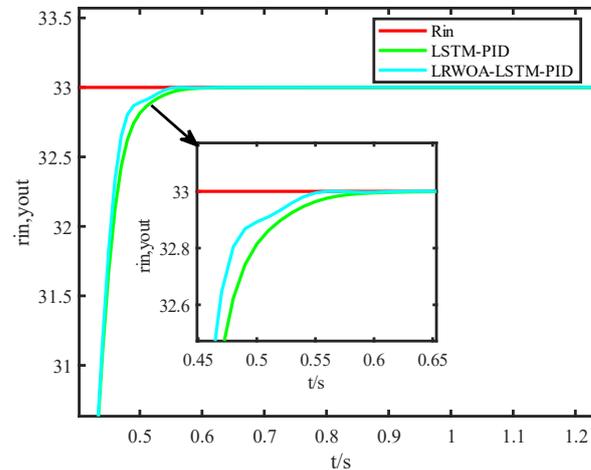


Figure 15. Comparison result diagram of LRWOA-LSTM-PID and LSTM-PID.

Table 3. Comparison result of LRWOA-LSTM-PID and LSTM-PID. Bold represents the best result.

CONTROLLER	LSTM-PID	LRWOA-LSTM-PID
Rise time (s)	0.61	0.55
Settling time (s)	0.61	0.55
Overshoot (%)	0.00	0.00
Peak (°C)	33.00	33.00
Peak time (s)	0.61	0.55
K_p	10.01	12.02
K_i	2.09	3.04
K_d	10.06	11.20
Error	0	0

In summary, the LRWOA-LSTM-PID controller has good performance, can achieve the rapid tuning of PID parameters, improves the convergence speed of the system, and reduces the overshoot of the system, which can meet the strict temperature control requirements in the lysozyme fermentation process, providing a guarantee for high-quality lysozyme production.

6. Conclusions

The paper designs a fermenter temperature mechanism model based on a dynamic equation description, which can reflect the temperature changes in the fermenter well. A method for PID parameter tuning based on LSTM is proposed, which can more reasonably adjust the network weights, speeding up the tuning of PID parameters. A Whale Optimization Algorithm (WOA) based on Levy flight and the random walk strategy (LRWOA) is proposed for the initialization of LSTM parameters, enhancing the global and local optimization capabilities of the neural network parameters. On this basis, a method of temperature control for lysosomal fermentation based on LRWOA-LSTM-PID is proposed. Experimental results show that the method proposed in this paper can achieve the rapid tuning of PID parameters K_p , K_i , and K_d , thus improving the convergence speed of the system and reducing the overshoot of the system.

Considering the further optimization of the proposed algorithm, the following future research is planned: (1) The verification of the LRWOA-LSTM-PID model on a hardware platform, and an attempt to implement this method in a real environment; (2) the enhancement of the real-time capabilities of LRWOA-LSTM-PID through offline parameter tuning; (3) the proposal of better controllers for a more accurate control of parameters such as dissolved oxygen in the lysosomal fermentation process.

Author Contributions: C.D.: Data curation, formal analysis, methodology, software, writing—original draft; X.L.: data curation, funding acquisition, project administration, formal analysis, investigation, writing—original draft; H.Z.: data curation, formal analysis, visualization; J.Y.: formal analysis, visualization, writing—original draft; J.D.: funding acquisition, project administration, resources, visualization; S.Z.: validation. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by Zhejiang Industrial Chain Collaborative Innovation Program of China (2020-330726-13-03-170837), in part by the Key Natural Science Foundation of Zhejiang Province in China (LZ22F010006).

Data Availability Statement: Data are contained within the article.

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

References

- Ferraboschi, P.; Ciceri, S.; Grisenti, P. Applications of Lysozyme, an Innate Immune Defense Factor, as an Alternative Antibiotic. *Antibiotics* **2021**, *10*, 1534. [[CrossRef](#)] [[PubMed](#)]
- Mandal, A.; Herr, J.C. Preparation of Active Human Lysozyme and Transgenic Organisms. Patent No. WO2006028497, 16 March 2006.
- Ercan, D.; Demirci, A. Recent Advances for the Production and Recovery Methods of L-lysozyme. *Crit. Rev. Biotechnol.* **2016**, *36*, 1078–1088. [[CrossRef](#)] [[PubMed](#)]
- Zhu, X.; Zhu, Z. The Generalized Predictive Control of Bacteria Concentration in Marine Lysozyme Fermentation Process. *Food Sci. Nutr.* **2018**, *6*, 2459–2465. [[CrossRef](#)]
- Lasanta, C.; Roldán, A.; Caro, I.; Pérez, L.; Palacios, V. Use of Lysozyme for the Prevention and Treatment of Heterolactic Fermentation in the Biological Aging of Sherry Wines. *Food Control* **2010**, *21*, 1442–1447. [[CrossRef](#)]
- Srikar, K.V.P.; Kumar, Y.V.P.; Pradeep, D.J.; Reddy, C.P. Investigation on PID Controller Tuning Methods for Aircraft Fuselage Temperature Control. In Proceedings of the 2020 International Symposium on Advanced Electrical and Communication Technologies (ISAECT), Kenitra, Morocco, 25–27 November 2020; pp. 1–5.
- Wu, X.; Wang, X.; He, G. A Fuzzy Self-Tuning Temperature PID Control Algorithms for 3D Bio-Printing Temperature Control System. In Proceedings of the 2020 Chinese Control And Decision Conference (CCDC), Hefei, China, 22–24 August 2020; pp. 2846–2849.
- Nasir, A.H.A.; Hambali, N.; Rahiman, M.H.F. Implementation of PRBS & RGS Perturbation Input Signals on Steam Temperature: Model Estimation and PID Control. In Proceedings of the 2023 19th IEEE International Colloquium on Signal Processing & Its Applications (CSPA), Kedah, Malaysia, 3–4 March 2023; pp. 99–104.
- Wu, X.; Wu, J.; Li, D. Designation and Simulation of Environment Laboratory Temperature Control System Based on Adaptive Fuzzy PID. In Proceedings of the 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, China, 12–14 October 2018; pp. 583–587.
- Åström, K.J.; Häggglund, T. Revisiting the Ziegler–Nichols Step Response Method for PID Control. *J. Process Control* **2004**, *14*, 635–650. [[CrossRef](#)]
- Franco, S.S.; Henríquez, J.R.; Ochoa, A.A.V.; da Costa, J.A.P.; Ferraz, K.A. Thermal Analysis and Development of PID Control for Electronic Expansion Device of Vapor Compression Refrigeration Systems. *Appl. Therm. Eng.* **2022**, *206*, 118130. [[CrossRef](#)]
- Muresan, C.I.; De Keyser, R. Revisiting Ziegler–Nichols. A Fractional Order Approach. *ISA Trans.* **2022**, *129*, 287–296. [[CrossRef](#)] [[PubMed](#)]
- Han, J.; Shan, X.; Liu, H.; Xiao, J.; Huang, T. Fuzzy Gain Scheduling PID Control of a Hybrid Robot Based on Dynamic Characteristics. *Mech. Mach. Theory* **2023**, *184*, 105283. [[CrossRef](#)]
- Bu, Q.; Cai, J.; Liu, Y.; Cao, M.; Dong, L.; Ruan, R.; Mao, H. The Effect of Fuzzy PID Temperature Control on The-rmal Behavior Analysis and Kinetics Study of Biomass Microwave Pyrolysis. *J. Anal. Appl. Pyrolysis* **2021**, *158*, 105176. [[CrossRef](#)]
- Singhal, K.; Kumar, V.; Rana, K.P.S. Robust Trajectory Tracking Control of Non-Holonomic Wheeled Mobile Robots Using an Adaptive Fractional Order Parallel Fuzzy PID Controller. *J. Frankl. Inst.* **2022**, *359*, 4160–4215. [[CrossRef](#)]
- Wei, Y.; Wu, Y.; Zhang, X.; Ren, J.; An, D. Fuzzy Self-Tuning PID-Based Intelligent Control of an Anti-Wave Buoy Data Acquisition Control System. *IEEE Access* **2019**, *7*, 166157–166164. [[CrossRef](#)]

17. Jia, L.; Zhao, X. An Improved Particle Swarm Optimization (PSO) Optimized Integral Separation PID and Its Application on Central Position Control System. *IEEE Sens. J.* **2019**, *19*, 7064–7071. [[CrossRef](#)]
18. Tang, G.; Lei, J.; Du, H.; Yao, B.; Zhu, W.; Hu, X. Proportional-Integral-Derivative Controller Optimization by Particle Swarm Optimization and Back Propagation Neural Network for a Parallel Stabilized Platform in Marine Operations. *J. Ocean Eng. Sci.* **2022**. [[CrossRef](#)]
19. Feng, H.; Ma, W.; Yin, C.; Cao, D. Trajectory Control of Electro-Hydraulic Position Servo System Using Improved PSO-PID Controller. *Autom. Constr.* **2021**, *127*, 103722. [[CrossRef](#)]
20. Mohamed, M.A.; Eltamaly, A.M.; Alolah, A.I. PSO-Based Smart Grid Application for Sizing and Optimization of Hybrid Renewable Energy Systems. *PLoS ONE* **2016**, *11*, e0159702. [[CrossRef](#)]
21. Lin, C.L.; Jan, H.Y.; Shieh, N.C. GA-Based Multiobjective PID Control for a Linear Brushless DC Motor. *IEEE/ASME Trans. Mechatron.* **2003**, *8*, 56–65. [[CrossRef](#)]
22. Hussain, M.; Aparna, V.; Shajahan, M.S.M.; Jamal, D.N. Tuning of PID Controller Using GA, SA and QFT for Three Interacting Tank Process. In Proceedings of the 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 11–12 May 2018; pp. 549–553.
23. Elbayomy, K.M.; Zongxia, J.; Huaqing, Z. PID Controller Optimization by GA and Its Performances on the Electro-Hydraulic Servo Control System. *Chin. J. Aeronaut.* **2008**, *21*, 378–384. [[CrossRef](#)]
24. Mousakazemi, S.M.H. Comparison of the Error-Integral Performance Indexes in a GA-Tuned PID Controlling System of a PWR-Type Nuclear Reactor Point-Kinetics Model. *Prog. Nucl. Energy* **2021**, *132*, 103604. [[CrossRef](#)]
25. Sinha, N.; Lai, L.L.; Rao, V.G. Venu Gopal Rao GA Optimized PID Controllers for Automatic Generation Control of Two Area Reheat Thermal Systems under Deregulated Environment. In Proceedings of the 2008 Third International Conference on Electric Utility Deregulation and Restructuring and Power Technologies, Nanjing, China, 6–9 April 2008; pp. 1186–1191.
26. Li, G.; Guo, C.; Li, Y.; Deng, W. Fractional-Order PID Controller of USV Course-Keeping Using Hybrid GA-PSO Algorithm. In Proceedings of the 2015 8th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China, 12–13 December 2015; Volume 2, pp. 506–509.
27. Cheng, C.-H.; Cheng, P.-J.; Xie, M.-J. Current Sharing of Paralleled DC–DC Converters Using GA-Based PID Controllers. *Expert Syst. Appl.* **2010**, *37*, 733–740. [[CrossRef](#)]
28. Mohamad Ali Tousi, S.; Mostafanasab, A.; Teshnehlab, M. Design of Self Tuning PID Controller Based on Competitional PSO. In Proceedings of the 2020 4th Conference on Swarm Intelligence and Evolutionary Computation (CSIEC), Mashhad, Iran, 2–4 September 2020; pp. 022–026.
29. Xue, H.; Bai, Y.; Hu, H.; Xu, T.; Liang, H. A Novel Hybrid Model Based on TVIW-PSO-GSA Algorithm and Support Vector Machine for Classification Problems. *IEEE Access* **2019**, *7*, 27789–27801. [[CrossRef](#)]
30. Agatonovic-Kustrin, S.; Beresford, R. Basic Concepts of Artificial Neural Network (ANN) Modeling and Its Application in Pharmaceutical Research. *J. Pharm. Biomed. Anal.* **2000**, *22*, 717–727. [[CrossRef](#)] [[PubMed](#)]
31. Varghese, M.M.; Vakamalla, T.R.; Gujjula, R.; Mangadoddy, N. Prediction of Solid Circulation Rate in an Internal Circulating Fluidized Bed: An Empirical and ANN Approach. *Flow Meas. Instrum.* **2022**, *88*, 102274. [[CrossRef](#)]
32. Liu, Z.; Liang, J.; Zhang, S.; Kayacan, E. ANN-Based Vibration Control of an Aerial Refueling Hose System with Input Nonlinearity and Prescribed Output Constraint. *J. Frankl. Inst.* **2022**, *359*, 2627–2645. [[CrossRef](#)]
33. Wu, W.; Dandy, G.C.; Maier, H.R. Protocol for Developing ANN Models and Its Application to the Assessment of the Quality of the ANN Model Development Process in Drinking Water Quality Modelling. *Environ. Model. Softw.* **2014**, *54*, 108–127. [[CrossRef](#)]
34. Huang, G.; Yuan, X.; Shi, K.; Wu, X. A BP-PID Controller-Based Multi-Model Control System for Lateral Stability of Distributed Drive Electric Vehicle. *J. Frankl. Inst.* **2019**, *356*, 7290–7311. [[CrossRef](#)]
35. Anh, H.P.H.; Nam, N.T.; Anh, H.P.H.; Nam, N.T. A New Approach of the Online Tuning Gain Scheduling Nonlinear PID Controller Using Neural Network. In *PID Control, Implementation and Tuning*; IntechOpen: Rijeka, Croatia, 2011; ISBN 978-953-307-166-4.
36. Asgharnia, A.; Jamali, A.; Shahnazi, R.; Maheri, A. Load Mitigation of a Class of 5-MW Wind Turbine with RBF Neural Network Based Fractional-Order PID Controller. *ISA Trans.* **2020**, *96*, 272–286. [[CrossRef](#)] [[PubMed](#)]
37. Dhaouadi, R.; Jafari, R. Adaptive PID Neuro-Controller for a Nonlinear Servomechanism. In Proceedings of the 2007 IEEE International Symposium on Industrial Electronics, Vigo, Spain, 4–7 June 2007; pp. 157–162.
38. Zhang, G.; Lu, R.; Chen, M. A Stable Self-Learning DRNN-PID Control of Multi-Point Mooring System. In Proceedings of the Advances in Guidance, Navigation and Control, Harbin, China, 5–7 August 2022; Yan, L., Duan, H., Yu, X., Eds.; Springer: Singapore, 2022; pp. 4769–4779.
39. Meng, Z.; Zhang, L.; Wang, H.; Ma, X.; Li, H.; Zhu, F. Research and Design of Precision Fertilizer Application Control System Based on PSO-BP-PID Algorithm. *Agriculture* **2022**, *12*, 1395. [[CrossRef](#)]
40. Jin, L.; Fan, J.; Du, F.; Zhan, M. Research on Two-Stage Semi-Active ISD Suspension Based on Improved Fuzzy Neural Network PID Control. *Sensors* **2023**, *23*, 8388. [[CrossRef](#)] [[PubMed](#)]
41. Mirjalili, S.; Lewis, A. The Whale Optimization Algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [[CrossRef](#)]
42. Dubkov, A.A.; Spagnolo, B.; Uchaikin, V.V. Lévy Flight Superdiffusion: An Introduction. *Int. J. Bifurc. Chaos* **2008**, *18*, 2649–2672. [[CrossRef](#)]
43. Geim, A.K. Nobel Lecture: Random Walk to Graphene. *Rev. Mod. Phys.* **2011**, *83*, 851–862. [[CrossRef](#)]

44. Shi, Y.; Eberhart, R.C. Parameter Selection in Particle Swarm Optimization. In Proceedings of the Evolutionary Programming VII, San Diego, CA, USA, 25–27 March 1998; Porto, V.W., Saravanan, N., Waagen, D., Eiben, A.E., Eds.; Springer: Berlin/Heidelberg, Germany, 1998; pp. 591–600.
45. Zhang, M.; Zhang, W.; Wang, J.; Xiong, X.; Cheng, Y.; Lv, X. Reconstructions of Strain Distribution along FBG Sensor Using GWO Algorithm and Interrelated Simulation Verification. In Proceedings of the 2022 IEEE International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC), Chongqing, China, 5–7 August 2022; pp. 1–5.
46. Luo, J.; Chen, H.; Wang, K.; Tong, C.; Li, J.; Cai, Z. LGWO: An Improved Grey Wolf Optimization for Function Optimization. In Proceedings of the Advances in Swarm Intelligence, Fukuoka, Japan, 27 July 27–1 August 2017; Tan, Y., Takagi, H., Shi, Y., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 99–105.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.