mathematics

MDPI

Article

Application of Fuzzy PID Based on Stray Lion Swarm Optimization Algorithm in Overhead Crane System Control

Jie Fu [1,*], Jian Liu [2], Dongkai Xie [3] and Zhe Sun [4,5,*]

[1] Rail Transit Department, Zhejiang Institute of Communications, Hangzhou 311112, China
[2] College of Information Engineering, Nanjing University of Finance and Economics, Nanjing 210023, China; liujian@nufe.edu.cn
[3] Alibaba Cloud Intelligence Business Group, Alibaba Co., Ltd., Hangzhou 310024, China; 3039911153@zju.edu.cn
[4] Post Industry Technology Research and Development Center of the State Posts Bureau (Internet of Things Technology), Nanjing University of Posts and Telecommunications, Nanjing 210023, China
[5] Post Big Data Technology and Application Engineering Research Center of Jiangsu Province, Nanjing University of Posts and Telecommunications, Nanjing 210023, China
* Correspondence: fujie@zjvtit.edu.cn (J.F.); zhesunny@njupt.edu.cn (Z.S.); Tel.: +86-0571-8848-1929 (J.F.); +86-1816-8407-332 (Z.S.)

Abstract: To solve the problem of crane anti-swing, fuzzy PID is a common method. However, the parameter configuration of fuzzy PID requires a lot of time and effort from professionals. Based on this, we introduce the LSO algorithm and add the stray operator, which effectively improves its global search performance. By combining SLSO and fuzzy PID and comparing them with other methods, this paper confirms that even without the targeted optimization by professionals, the optimization algorithm can find the appropriate parameter configuration for fuzzy PID which can be effectively used in the crane anti-swing problem.

Keywords: overhead crane; anti-swing control; fuzzy PID; SLSO algorithm

MSC: 93-10; 93B45; 93C42

check for updates

Citation: Fu, J.; Liu, J.; Xie, D.; Sun, Z. Application of Fuzzy PID Based on Stray Lion Swarm Optimization Algorithm in Overhead Crane System Control. *Mathematics* **2023**, *11*, 2170. https://doi.org/10.3390/math11092170

Academic Editor: António Lopes

Received: 29 March 2023
Revised: 22 April 2023
Accepted: 23 April 2023
Published: 5 May 2023

1. Introduction

Fuzzy logic control provides a system theory method for experts to construct language information and convert it into control strategies, which can solve many complex control problems that cannot establish accurate mathematical model systems, so it is an effective method to deal with imprecision and uncertainty in reasoning system and control systems [1–5]. Because of this, the parameter setting of fuzzy control has always depended on the personal ability of experts.

PID control is widely used in various fields because of its advantages of simple structure, strong stability, and convenient adjustment [6–10]. As the core content of control system design, PID parameter tuning is the key factor to determine the control effect. In general, PID parameter tuning can be divided into theoretical calculation tuning and engineering tuning. The former is mainly based on the mathematical model of the system and determines the controller parameters through theoretical calculation, but it still needs to be adjusted and modified on site. The latter mainly adjusts the parameters manually according to field operation experience. Because operator experience is not easy to accurately describe, various semaphores and evaluation indicators in the control process are not easy to quantitatively express, the adjustment process is not controllable, and the optimization effect is extremely dependent on personal ability.

Overhead cranes are indispensable in the construction of bridges, docks, and other buildings. An overhead crane is a typical underactuated system. During the usage, because the sling cannot fully control the load, the swing of the load may collide with other objects.

At present, an overhead crane must be operated by experienced workers. However, due to the inaccuracy of manual operation, it is still impossible to avoid safety accidents. Therefore, the construction industry needs to design a stable and efficient anti-interference controller for the bridge crane system and realize automatic control of the bridge crane system. There are many studies in this field [11–14].

There are many examples of the combination of fuzzy control and PID [15–17], among which Sun et al. applied it to anti-swing control of an overhead crane, and achieved good control results [18]. This avoids the problem of the parameter optimization of PID, but also ushers in new problems. The parameter optimization of fuzzy control is more difficult than PID. This not only requires optimization personnel to have rich practical experience but also requires certain knowledge of fuzzy mathematics, which undoubtedly raises the application threshold of fuzzy control.

In 2016, the lion swarm optimization algorithm was proposed [19]. The lion swarm optimization algorithm is a nature-inspired algorithm based on the special lifestyle of lions and their cooperative behaviors [20]. Compared with some algorithms, the lion swarm optimization algorithm is a new meta-heuristic algorithm, which has the characteristics of simple operation, fast convergence speed, and a small amount of calculation. Subsequently, LSO has been continuously optimized or applied in various fields [21–24]. Although LSO has the characteristics of fast convergence speed and a small amount of calculation, it is still easy to fall into local optimization to a certain extent. Strengthening the algorithm is very important for the optimization performance of complex systems, and there are many papers worthy of reference in this regard [25–29].

PID does not perform well in the control of overhead cranes due to the difficulty in handling the control of nonlinear systems. After the introduction of fuzzy PID, although the x-performance is improved, the dependence of fuzzy PID on expert experience limits its control performance and generalization capability. To improve the performance of the fuzzy PID controller, we proposed an SLSO algorithm-based fuzzy PID controller for overhead crane systems. The contribution can be summarized as follows: (1) a stray operation is introduced to improve the LSO algorithm, which can enhance the population diversity and further reduce the risk of falling into local optimization, to improve the convergence accuracy of the algorithm. (2) Adaptive parameter adjustment based on the SLSO algorithm is designed to eliminate the dependency on experts. In addition, the effectiveness of adaptive fuzzy parameter configuration was verified via anti-swing experiments of the overhead crane.

This paper is divided into six sections. The Section 1 is a general introduction. Section 2 elaborates on the theoretical model and control formulation of an overhead crane. Section 3 introduces the LSO algorithm and our improvement, and conducts a comparison experiment with other algorithms on the test function. Section 4 describes how the SLSO algorithm is combined with fuzzy PID and applied to the control of overhead cranes. Section 5 gives the results and analysis of the simulation experiments. Section 6 provides a summary of this paper and an outlook for future work.

2. Introduction of the Overhead Crane System Model

As shown in Figure 1, the control system controls the horizontal movement of the trolley on the bridge, and the movement and swing of the load can only be indirectly controlled by controlling the movement of the trolley.

According to Euler Lagrange method, the dynamic model of an overhead crane is as follows:

$$(m_l + m_c)\ddot{x} + m_l l(\ddot{\theta}\cos\theta - \dot{\theta}^2 \sin\theta) = u \tag{1}$$

$$m_l \cos\theta + m_l l\ddot{\theta} + m_l g \sin\theta = 0 \tag{2}$$

where x and θ denote the displacement of the trolley and the swing angle of the load, m_l and m_c, respectively, represent the mass of the load and the trolley, l is the length of the sling, u is the control force exerted on the trolley, and g is the acceleration of gravity.

Setting $\dot{x} = x_1, \ddot{x} = x_2, \dot{\theta} = x_3, \ddot{\theta} = x_4$, the following differential equations can be obtained by transformation:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \frac{m_l g \cos x_3 \sin x_3 + m_l l \dot{x}_3^2 \sin x_3 + u}{m_l + m_c - m_l \cos^2 x_3} \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = \frac{m_l l \dot{x}_3^2 \cos x_3 \sin x_3 + (m_l + m_c) g \sin x_3 + u \cos x_3}{l(m_l + m_c - m_l \cos^2 x_3)} \end{cases} \quad (3)$$

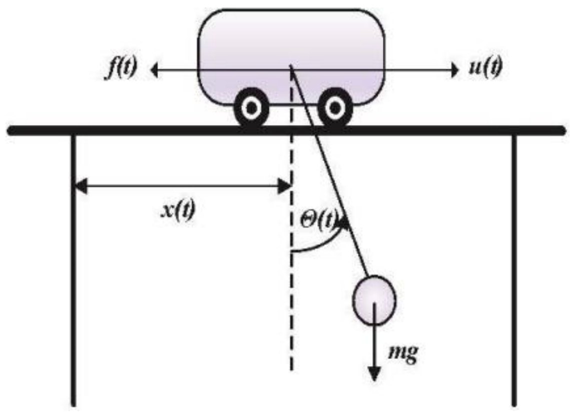

Figure 1. The structure diagram of an overhead crane.

3. The Stray Lion Swarm Optimization Algorithm

This section will employ many symbols, and their meanings are shown in Table 1.

Table 1. The nomenclature list.

Symbols/Abbreviations	Meaning
x_i^k	the i-th individual in the k-th generation population
p_i^k	the historical best position of the i-th individual from the 1st to the k-th generation
g^k	the best position of the k-th generation population
p_c^k	the individual randomly selected from the k-th generation lioness group
p_m^k	the individual randomly selected from the k-th generation lion group
q,γ	random value, $q \sim N(0,1), \gamma \sim U(0,1)$
α_f,α_c	disturbance factor
$f^k[i]$	the value of the stray individual at the i-th dimension in the k-th generation

3.1. Standard Lion Swarm Optimization Algorithm

In order to search for better solutions, the Lion King will conduct a range search based on the historical optimal solution. The formula for updating the position is as follows:

$$x_i^{k+1} = g^k(1 + \gamma \| p_i^k - g^k \|) \quad (4)$$

A lioness randomly selects another lioness to cooperate with, and the formula for updating the position is as follows:

$$x_i^{k+1} = \frac{p_i^k + p_c^k}{2}(1 + \alpha_f \gamma) \quad (5)$$

There are three updating strategies for young lions, namely follow the lioness, follow the lion king, or leave the location of the group to search for the updated location in reverse. The formula is as follows:

$$x_i^{k+1} = \begin{cases} \frac{g^k+p_i^k}{2}(1+\alpha_c\gamma), 0 \le q \le \frac{1}{3} \\ \frac{p_m^k+p_i^k}{2}(1+\alpha_c\gamma), \frac{1}{3} < q \le \frac{2}{3} \\ \frac{\overline{g}^k+p_i^k}{2}(1+\alpha_c\gamma), \frac{2}{3} < q \le 1 \end{cases} \tag{6}$$

where x_i^k refers to the i-th individual in the k-th generation population; γ is a random number generated according to the normal distribution N(0, 1); p_i^k is the historical best position of the i-th individual from the 1st to the k-th generation; g^k is the best position of the k-th generation population; p_c^k is randomly selected from the k-th generation lioness group; p_m^k is randomly selected from the k-th generation lion group; q is the uniform random value generated according to the uniform distribution U [0, 1]; $\overline{g}^k = \overline{low} + \overline{up} - g^k$; $\overline{low}$ and $\overline{up}$ are the minimum value and maximum value of each dimension within the range of lion activity space; α_f and α_c are the disturbance factor. The calculation method is as follows:

$$\alpha_f = 0.1(\overline{up} - \overline{low}) \times \exp\,(-\frac{30t}{T})^{10} \tag{7}$$

$$\alpha_c = 0.1(\overline{up} - \overline{low}) \times (\frac{T-t}{T}) \tag{8}$$

where t is the current number of iterations and T is the maximum number of iterations.

3.2. Stray Lion Swarm Optimization Algorithm

Although LSO has the advantages of high search efficiency and fast convergence speed, it still cannot solve the problem where swarm intelligence can easily fall into local optimization. In the LSO, most individuals will iterate with the lion king as the core, so it is difficult to escape when they fall into the local optimal solution.

In this paper, a stray individual is introduced as optimization interference, which can effectively avoid falling into local optimal solutions and obtain better optimization results on the premise of ensuring the population size.

3.2.1. Stray Operation in SLSO

The stray individual introduced in this paper deviates from the algorithm as far as possible in scope. At the same time, to avoid the individual falling into an extremely bad state, it is necessary to introduce a certain random quantity to ensure the effect. The formula for each generation of the stray individual is as follows:

$$f^k[i] = ((up[i] - low[i]) \times \gamma + 2low[i] + up[i] - g^k[i])/2 \tag{9}$$

where $f^k[i]$ is the value of the stray individual at the i-th dimension in the k-th generation. $up[i]/low[i]$ is the upper/lower limit at the i-th dimension. $g^k[i]$ is the value of the lion king at the i-th dimension in the k-th generation.

3.2.2. Iteration Strategy of the Lion Swarm in SLSO

To avoid the negative impact of dissociated individuals on the population, this paper sets a participation probability. When the probability is met, the cubs and females will be updated according to the new formula. In general, this probability is set to 0.1. Formula (11) is the new iterative strategy of the female lion, and Formula (12) is the new iterative strategy of the young lion.

$$x_i^{k+1} = \frac{p_i^k + f^k}{2} \tag{10}$$

$$x_i^{k+1} = \begin{cases} \frac{g^k+f^k}{2}, 0 \le q \le \frac{1}{3} \\ \frac{x_i^k+f^k}{2}, \frac{1}{3} < q \le \frac{2}{3} \\ \frac{\overline{x}_i^k+f^k}{2}, \frac{2}{3} < q \le 1 \end{cases} \quad (11)$$

where f^k is the stray individual in the k-th generation. $\overline{x}_i^{k+1} = \overline{low} + \overline{up} - x_i^k$.

3.2.3. Convergence Proof of SLSO

The SLSO algorithm is improved based on the LSO algorithm, and this paper first makes a proof of the convergence of the LSO algorithm. The proof refers to reference [30].

(1) Markov chain model of the LSO algorithm

The position update of each individual is obtained by Gaussian sampling, where the position update distribution of the lion king is $x_i(t+1) \sim N(g, |p_i, g|_2)$.

The position update distribution of the lioness is $x_i(t+1) \sim N(\frac{P_i+P_c}{2}, \alpha_f^2)$.

The position update distribution of the young lion is as follows:

$$x_i(t+1) \sim \begin{cases} N(\frac{g+p_i}{2}, \alpha_c^2), q < 1/3 \\ N(\frac{p_c+p_i}{2}, \alpha_c^2), 1/3 \le q < 2/3 \\ N(\frac{\overline{g}+p_i}{2}, \alpha_c^2), 2/3 \le q < 1 \end{cases}$$

In which q = rand [0,1).

To illustrate the Markov chain model of the LSO algorithm, the following definitions and mathematical descriptions are given.

Definition 1. *Lion swarm state and state space. The set of all states in the pride constitutes the state space of the pride, denoted as follows:*

$$|s = (x_1, x_2, \cdots, x_i, \cdots x_N)|x_i = (x_{i1}, x_{i2}, \cdots, x_{id}, \cdots, x_{iD}), 1 \le i \le N, 1 \le d \le D$$

Definition 2. *State transfer of individuals, For $\forall x_i \in s, x_j \in s$, the lion is transferred from state x_i to state x_j in one step, denoted as $T_s(x_i) = x_j$.*

Theorem 1. *Transfer probability of the LSO algorithm:*

$$P(T_s(x_i) = x_j) = \begin{cases} P_m(T_s(x_i) = x_j), lion - king \\ P_f(T_s(x_i) = x_j), lioness \\ P_c(T_s(x_i) = x_j), young - lion \end{cases}$$

Proof. The corresponding one-step transfer probabilities are different because of the different ways to update the positions of the three lions. The lions' positions can be viewed as a set of points in the hyperspace, and the position update process is a point set transformation in the hyperspace. For computational convenience, let the changed point set obey a uniform distribution U(−1,1) so that the transfer probability of the male lion can be obtained. The transfer probability of the lion king is shown as follows:

$$P_m(T_s(x_i) = x_j) = \begin{cases} \frac{1}{2(|g-p_i|)}, x_j \in [g - \left|g - p_i\right|, g + \left|g - p_i\right|] \\ 0, esle \end{cases} \quad (12)$$

The transfer probability of the lioness is shown as follows:

$$P_m(T_s(x_i) = x_j) = \begin{cases} \frac{1}{2\alpha_f(|p_c - p_i|)}, x_j \in [\frac{p_i + p_c}{2} - \alpha_f \left| p_c - p_i \right|, \frac{p_i + p_c}{2} + \alpha_f \left| p_c - p_i \right|] \\ 0, esle \end{cases} \tag{13}$$

The transfer probability of the young lion is shown as follows:

$$P_c(T_s(x_i) = x_j) = \begin{cases} \frac{1}{2\alpha_c(|g - p_i|)}, x_j \in [\frac{p_i + g}{2} - \alpha_c \left| g - p_i \right|, \frac{p_i + g}{2} + \alpha_c \left| g - p_i \right|] \\ \frac{1}{2\alpha_c(|p_m - p_i|)}, x_j \in [\frac{p_i + g}{2} - \alpha_c \left| p_m - p_i \right|, \frac{p_i + g}{2} + \alpha_c \left| p_m - p_i \right|] \\ \frac{1}{2(|\overline{g} - p_i|)}, x_j \in [\frac{p_i + \overline{g}}{2} - \left| \overline{g} - p_i \right|, \frac{p_i + \overline{g}}{2} + \left| \overline{g} - p_i \right|] \end{cases} \tag{14}$$

□

Definition 3. *State transfer probabilities of lion swarm. For* $\forall s_i \in S$ *and* $\forall s_j \in S$*, S is the set of lion pride states, and the probability of a lion pride transferring from* s_i *to* s_j *in one step, denoted as* $T_s(s_i) = s_j$*, is* $P(T_s(s_i) = s_j) = \prod_{i=1}^{N} P(T_s(s_i) = x_j^i)$.

where N is the number of individuals in the pride, and x_j^i is the state corresponding to individual x_i. The one-step transfer probability of the pride state in the LSO algorithm is the simultaneous transfer of the states of all lions in the pride.

(2) Convergence analysis of the LSO algorithm

According to the authors in [31], the definitions of Markov chain, finite Markov chain, and chi-square Markov chain are no longer given in this paper; see [31] for details.

Theorem 2. *The population sequence generated by the LSO algorithm* $\{s(t), t \geq 0\}$ *is a finite chi-square Markov chain, where t is the number of iterations.*

Proof.

1. According to Definition 3, in the population sequence $\{s(t), t \geq 0\}$, $\forall s(t) \in S$ and $\forall s(t+1) \in S$, the transfer probability $P(T_s(s(t)) = s(t+1))$ is determined by the transfer probability $P(T_s(x(t)) = x(t+1))$ of all lions.
2. According to Theorem 1, the state transfer probability of any lion in the pride is only related to the state at moment t and other randomly selected individuals in the population at moment t. Therefore, $P(T_s(x(t)) = x(t+1))$ is only related to the state at moment t, but not to t.
3. According to 1 and 2, it can be seen that the population sequence generated by the LSO algorithm has Markov property, and because the state space $\{s(t), t \geq 0\}$ of the lion population is finite, according to the definition of finite Markov chain, the population sequence $\{s(t), t \geq 0\}$ generated by the LSO algorithm constitutes a finite Markov chain.
4. According to Theorem 1, $P(T_s(s(t)) = s(t+1))$ is also only related to the state at moment t of s, but not to t. Therefore, the population sequence produced by the LSO algorithm $\{s(t), t \geq 0\}$ is a finite chi-square Markov chain.

□

According to the authors in [32], it is known that the stochastic algorithm converges globally, and the LSO algorithm is a stochastic search algorithm, so this paper will determine the convergence of the LSO algorithm according to the convergence criterion of the stochastic algorithm.

(3) Convergence proof of LSO algorithm

Definition 4. *The set of optimal states of the lion population is G. Let the optimal solution of the optimization problem* $< A, f >$ *be* g^**, and define the set of optimal states of the lion swarm as follows:* $G = \{s = (x_1, x_2, \cdots x_i, \cdots, x_N) | f(x_i = f(g^*), x_i \in S, s \in S)\}$*. If G = S, then every solution in the feasible space is not only a feasible solution, but also an optimal solution. At this point, the iteration is meaningless, the following discussion of* $G \subset S$*.*

Theorem 3. *The optimal set of lion states G of the lion group algorithm is a closed set on the state space S.*

Proof. $\forall s_i \in G, s_j \notin G, s_j \in S$, the transfer probability of $T_s(s_i) = s_j$ is $P(T_s(s_i) = s_j) = \prod_{i=1}^{N} P(T_s(s_i) = x_j^i)$.

At least one lion state in G is optimal, and let $g^* \sim x_{i_0k}$ be the optimal state, i.e., at least $\exists x_{i_0k} \in G$, $P(T_S(x_{i_0k}) = x_{jk}) = 0$.

At this point, $P(T_s(s_i) = s_j) = 0$, so the set of optimal lion swarm states G is a closed set on the state space S. □

Theorem 4. *There is no nonempty closed set M in the state space S of the lion population such that* $M \cap G = \varphi$*.*

Proof. Suppose there exists a nonempty closed set M in the state space S, and $M \cap G = \varphi$, let $s_i = s(g^*, g^*, \cdots, g^*) \in G$, $s_j = (x_{j1,} x_{j2}, \cdots, x_{jd}) \in M$, and we have $f(x_j) > f(g^*)$.

According to the Chapman–Kolmogorov equation, we can obtain the result as follows:

$$P^l_{s_j,s_i} = \sum_{s_{r1} \in S} \cdots \sum_{s_{rl-1} \in S} P(T_S(s_j) = s_{r1}) P(T_S(s_{r1}) = s_{r2}) \cdots P(T_S(s_{rl-1}) = s_{ri}) \tag{15}$$

The algorithm will satisfy the conditions (12)–(14) in Theorem 1 after finitely many iterations of m. Therefore, the one-step transfer probability of each term of the expansion in Equation (15) satisfies $P(T_s(s_{r_{c+j}}) = s_{r_{c+j+1}}) > 0$ when the step size is large enough.

Therefore, $P^l_{s_j,s_i} > 0$, which yields that M is not a closed set. Thus, the Markov chain of lion group states is not approximately separable, and the z-state space S does not contain closed sets other than G. □

Theorem 5. *Assume that the Markov chain has a nonempty closed set E and there does not exist another nonempty closed set O, such that* $E \cap O = \varphi$*, when* $j \in E$*, there is* $\lim_{k \to \infty} P(x_k = j) = \pi_j$*. When* $j \notin E$*, there is* $\lim_{k \to \infty} P(x_k = j) = 0$*.*

Proof. For the proof process, please refer to [33]. □

Theorem 6. *When the iterations within the lion group tend to infinity, the lion group state must enter the optimal set of states G.*

Proof. From Theorems 3–5, Theorem 6 holds. □

Theorem 7. *The LSO algorithm can converge to the global optimum.*

Proof. The LSO algorithm is stochastic, so the LSO algorithm satisfies the condition of global convergence of stochastic algorithms H1 [33], and we know from Theorem 6 that the probability that the LSO algorithm does not search for the global optimal solution for an infinite number of consecutive times is 0. Then, we have $\prod_{k=0}^{\infty} (1 - u_k[B]) = 0$.

where $u_k[B]$ is the probability measure of the k-th iteration of the LSO algorithm to search for a solution to the set B, which satisfies the global convergence condition of the most taboo algorithm H2 [33]. For the LSO algorithm at each iteration, the update of the individual historical optimum takes the retention mechanism of the optimal individual, when the iteration tends to infinity. $\lim_{k\to\infty} P(x_k \in R_{\varepsilon,M}) = 1$. $\{x_k\}_{k=0}^{\infty}$ is the sequence generated by the iteration of the LSO algorithm, according to the global convergence of the stochastic search algorithm. It can be concluded that the LSO algorithm is globally convergent. □

Theorem 8. *The SLSO algorithm is globally convergent.*

Proof. The dissociation operator only sets up a stray individual outside the population and jointly searches for non-optimal solutions within the population at low probability (probability = 0.1). This means that the population sequence convergence of the SLSO algorithm with size n is equivalent to the population sequence convergence of the LSO algorithm with size 10n/9. The SLSO algorithm proposed in this article still meets the following requirements:

1. The population evolution direction in the SLSO algorithm is monotonic, i.e., $F(X(n+1)) \leq F(X(n))$
2. The population sequence of the SLSO algorithm $\{X(n), n \notin N^+\}$ is a homogeneous Markov chain
3. The Markov chain of the SLSO algorithm $\{X(n), n \notin N^+\}$ converges with probability 1 to a subset of the satisfactory population M $M_0 = M_0^* = \left\{ Y = (y_1, \ldots, y_{N_p}) \middle| y_i \in M^* \right\}$ in the solution space, i.e., $\lim_{n\to\infty} P\left(X(n) \in M_0^* \middle| X(0) = X_0\right) = 1$.

Therefore, it can be inferred that the SLSO algorithm in this paper converges.

The relevant symbols are common symbols for the convergence proof of swarm intelligence algorithms, and there will be no expansion explanation here. □

3.3. Numerical Experiments

To verify the performance of the SLSO presented in this paper, six well-known benchmark functions are used. For comparison, the standard LSO, standard GA, and standard PSO algorithms are adopted during the test process.

For a fair comparison, the population = 50, dimension = 10, number of iterations = 100, and each algorithm runs 50 times for each test function. Some of these benchmark functions are lower than 10 dimensions. Since the goal of this paper is to optimize the parameter configuration of ADRC, all benchmark functions have been increased to 10 dimensions for calculation. We take the average of the results of 50 runs as the result to eliminate the uncertain factors in the search process. The final iteration result is compared with the number of iterations used to achieve the optimized iteration result, and the results are shown in Table 2. Information on these benchmark functions is shown in Table 2, too.

Table 2. Comparative results of benchmark functions.

Function Name	SLSO		Standard LSO		GA		PSO		Min Value
	Avg_Result	Std	Avg_Result	Std	Avg_Result	Std	Avg_Result	Std	
Schwefel	3.55×10^3	5.05×10^4	3.66×10^3	6.36×10	3.64×10^3	5.75	3.61×10^3	1.44×10	0
Styblinski Tang	-2.69×10	6.01×10^{-4}	-1.48×10	4.21	-8.58	2.92	-2.68×10	1.02×10^{-10}	-2.903534
Beale	1.01×10^{-5}	1.53×10^{-5}	2.23×10^{-5}	1.58×10^{-1}	2.69×10^{-3}	4.37×10^{-3}	5.69×10^{-5}	2.81×10^{-11}	0
Easom	-9.99×10^{-1}	3.15×10^{-5}	-9.84×10^{-1}	8.89×10^{-2}	-1.07×10^{-2}	7.57×10^{-2}	-2.00×10^{-2}	1.41×10^{-1}	-1
Eggholder	-9.42×10^2	2.66×10	-8.95×10^2	6.58×10	-8.66×10^2	1.12×10^2	-7.12×10^2	9.21×10	-959.6407
Holder_table	-1.85×10	4.74×10^{-1}	-1.82×10	1.13	-1.74×10	1.52	-1.83×10	4.15×10^{-1}	-19.2085

Please note that it is not that there are no more experiments, but that most of the test functions are not difficult to optimize for LSO. Therefore, we only select functions with poor LSO performance to show the improvement effect.

Figure 2 shows the convergence curves of four algorithms with benchmark functions, and shows the performance of the SLSO more intuitively and clearly. From Figures 2–7, we can find that in Function Eason, the LSO obtains a worse result than GA, but the SLSO obtains a better result than the other three methods.

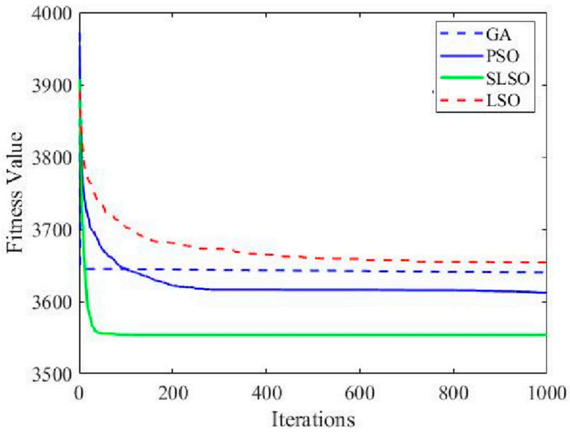

Figure 2. Schwefel.

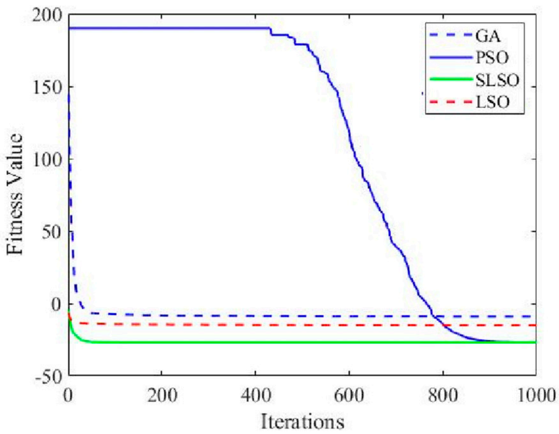

Figure 3. Styblinski-Tang.

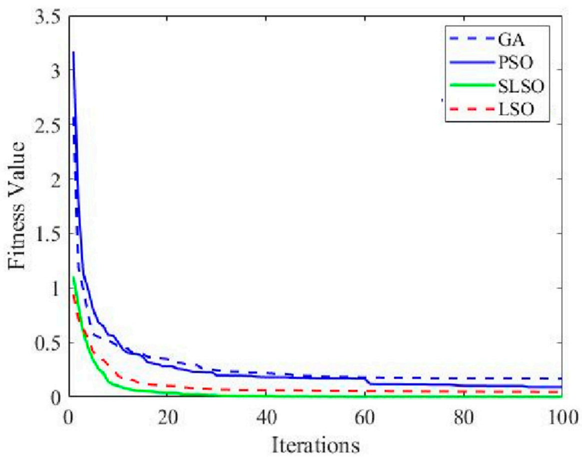

Figure 4. Beale.

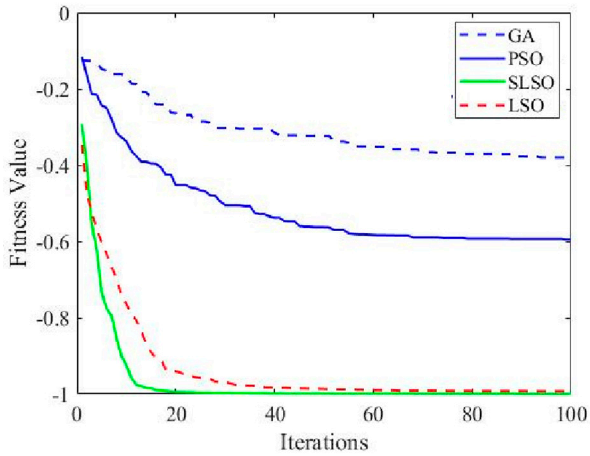

Figure 5. Eason.

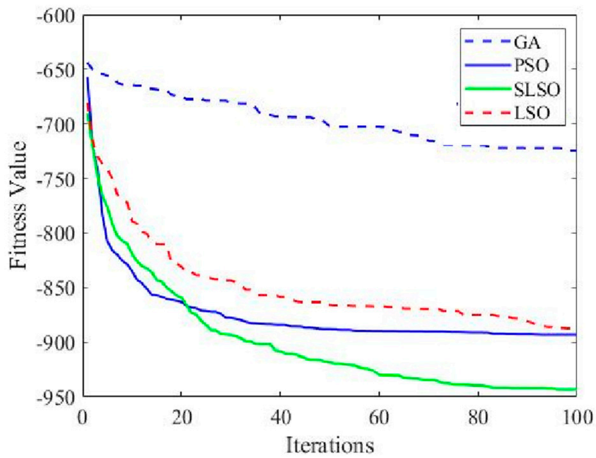

Figure 6. Eggholder.

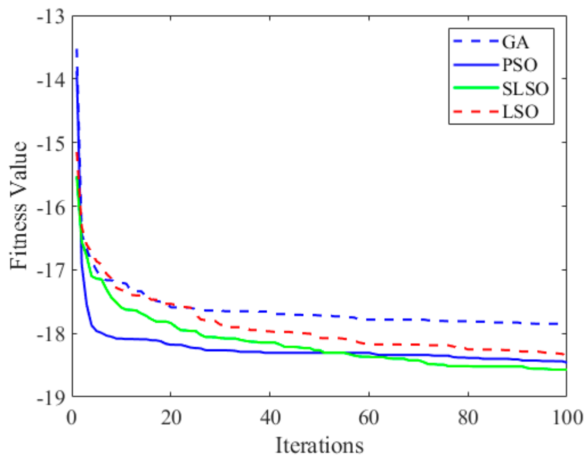

Figure 7. Holder-table.

In Function Styblinski_Tang, Beale, Eggholder, and Holder_table, the LSO works worse than PSO, but better than GA. However, after being improved, it works better than PSO.

In Function Schwefel, we can find that LSO works worst, and the SLSO works best.

We can find that in Table 2, the standard deviation of the results of the function based on the SLSO algorithm for finding the best is smaller than that of the LSO-based, which means that SLSO can find the optimal solution more stably, rather than relying on randomness.

In conclusion, SLSO can not only further improve the optimization results of LSO, but also perform well in the face of functions where LSO is not good at optimizing. Therefore, we can think that the improvement in this paper not only improves the accuracy of the algorithm, but also improves the applicability of the algorithm, and the improvement effect is ideal.

4. Designing of SLSO-Based Fuzzy PID

4.1. Fuzzy PID of Overhead Crane

The structure of fuzzy PID is shown in Figure 8. By calculating the system error $e(t)$ and error change rate $ec(t)$, and combining them with expert experience, the change rates of K_p, K_i and K_d can be deduced through fuzzy rules.

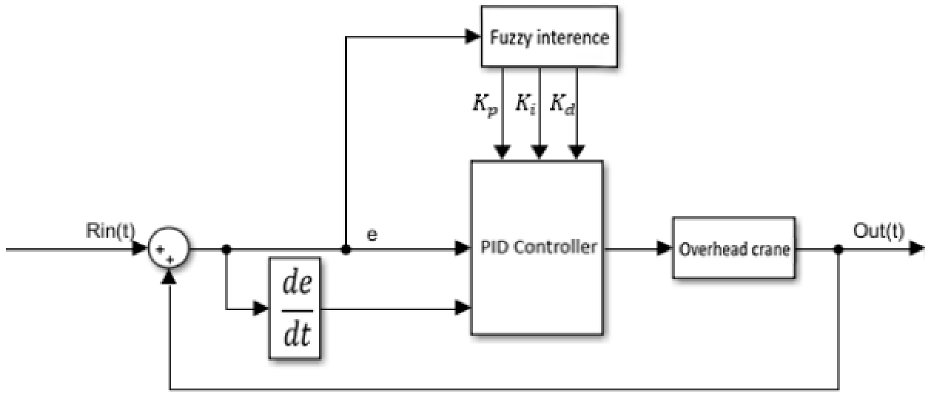

Figure 8. The diagram of fuzzy PID.

The value ranges of $e(t)$, $ec(t)$ and the fuzzy domains of K_p, K_i, and K_d are [−10, 10], [−10, 10], and [−6, 6], respectively. Generally speaking, fuzzy rules include {NB, NM, NS, NZ, ZO, PZ, PS, PM, PB}, and may contain different numbers of fuzzy rules according to different situations. The update method of K_p, K_i, and K_d is shown in Formula (16).

$$K_p = K'_p + \Delta K_p, K_i = K'_i + \Delta K_i, K_d = K'_d + \Delta K_d \tag{16}$$

The fuzzy rules of K_p, K_i, and K_d are shown in Tables 3–5.

Table 3. Fuzzy RULES of Kp.

e(t)() Δkp ec(t)	**NB**	**NM**	**NS**	**Z**	**PS**	**PM**	**PB**
NB	PB	PB	PB	PB	PM	PS	Z
NM	PM	PM	PS	PS	PS	Z	Z
NS	PM	PS	Z	Z	Z	NS	NM
Z	PS	PS	Z	Z	Z	NM	NB
PS	NM	NS	Z	Z	Z	PS	PM
PM	Z	Z	PS	PM	PM	PB	PB
PB	Z	PS	PB	PB	PB	PB	PB

Table 4. Fuzzy RULES of Ki.

e(t)() Δki ec(t)	**NB**	**NM**	**NS**	**Z**	**PS**	**PM**	**PB**
NB	PB	PB	PB	PB	PM	PS	Z
NM	PB	PB	PB	PM	PS	Z	Z
NS	PB	PM	PS	PS	Z	NS	NM
Z	NM	NS	Z	Z	Z	NS	NM
PS	NM	NS	Z	PS	PS	PM	PB
PM	Z	Z	PS	PM	PM	PB	PB
PB	Z	PS	PB	PB	PB	PB	PB

Table 5. Fuzzy RULES of Kd.

e(t)() Δkd ec(t)	**NB**	**NM**	**NS**	**Z**	**PS**	**PM**	**PB**
NB	PB	PM	PS	PB	NB	NB	NB
NM	PM	PS	Z	PS	NB	NS	Z
NS	PB	PM	Z	PS	PS	PM	PB
Z	PB	Z	PS	PS	PS	PM	PB
PS	PB	PM	PS	Z	PS	PM	PB
PM	Z	NS	NM	NS	Z	PS	PM
PB	NB	NB	NB	NS	PS	PM	PB

4.2. SLSO-Based Fuzzy PID

The fuzzy rule setting of fuzzy PID can be obtained quickly according to expert experience, but the value setting needs repeated debugging.

In this paper, SLSO is introduced to the interval design of fuzzy numbers.

For PID control, three rules require fuzzy control, each with two input parameters and one output parameter. Each parameter has seven situations and is controlled by seven arrays. Due to the symmetry of the parameters themselves, each parameter requires six numbers to control.

The interval of parameters is determined; therefore, the optimization of fuzzy rules can be transformed into the segmentation of the interval, and the number of segmented nodes is the number we need. Therefore, we patrol through seven numbers, with a range of values in the range (0–100). The ratio of the seven numbers is equivalent to the length ratio between the divided partitions within the interval. In this way, we can obtain the six numbers of the segmentation interval.

In summary, each interval needs 7 parameters, each fuzzy rule needs 3 intervals, this experiment has 3 fuzzy rules, so each individual needs to have 63 dimensions to optimize the parameters of the fuzzy rule. Then, the basic PID parameters also need to be optimized, which means that the length of each individual is 66.

The objective function is set as the product of the actual travel distance and the cumulative swing angle.

The steps are as follows:

Step 1 Establish the overhead crane control system.

Step 2 Set the individual length to 66, where the first and the second mean K_p and K_d, the left mean values of fuzzy rules. The population number to 30, and the number of iterations to 1000 to initialize the population.

Step 3 Analyze individual values, generate FIS files, read them into the base workspace, start Simulink, read the output of the simulation, and calculate individual fitness.

Step 4 Set $i = i + 1$, update individual values according to SLSO.

Step 5 If the obtained parameters meet the termination criteria or i = I_itermax, stop the algorithm and output the result. Otherwise, return to Step 3.

Step 6 Save the best individual as FIS files.

where the FIS file is a file type, which is used to save fuzzy rules and values.

5. Simulation Experiment

Generally speaking, in fuzzy PID, a set of fuzzy rules contains dozens of intervals.

To verify the validity of the proposed method, fuzzy PID control based on SLSO, fuzzy PID control without adjustment, adaptive PID based on the DE algorithm (hereinafter referred to as PID-DE), and the traditional PID control method are simulated under different conditions. The conditions are shown in Table 6.

Table 6. Specific experimental conditions.

Conditions	1	2	3	4	5	6
m_l/kg	7	7	7	12	12	12
x_d	6	12	20	6	12	20

The parameters of the overhead crane are set as M_T = 22 kg, l = 1 m, g = 9.81 m/s^2. The parameters of traditional PID are set to (60, 0, 60), and the parameters of PID-DE refer to the relevant paper [34]. In addition, the relevant parameters of other methods are listed below.

The values of fuzzy PID are based on SLSO and which, without adjustment, are shown in Tables 7–10. In addition, the parameters of fuzzy PID based on SLSO are optimized and set to (31.7, 0, 44.5). The parameters of fuzzy PID without optimization are set to (50, 0, 50) because the up limit is 100 and the low limit is 0 in the process of optimizing the PID parameters using the SLSO algorithm.

Table 7. Values of fuzzy PID rules Kp based on SLSO.

Name	NB	NM	NS	Z
e(t)	[−5.829,−4.457]	[−5.829,−4.457,−4.114,−2.742]	[−4.114,−2.742,−2.4,−1.658]	[−2.4,−1.658,1.658,2.4]
ec(t)	[−5.143,−4.286]	[−5.143,−4.286,−3.429,−2.571]	[−3.429,−2.571,−1.714,−0.857]	[−1.714,−0.857,0.857,1.714]
Δk_p	[−8.571,−7.143]	[−8.571,−7.143,−5.714,−4.286]	[−5.714,−4.286,−2.857,−1.429]	[−2.857,−1.429,1.429,2.857]
Name	PS	PM	PB	
e(t)	[1.658,2.4,2.743,4.114]	[2.742,4.114,4.457,5.829]	[4.457,5.829]	
ec(t)	[0.857,1.714,2.571,3.429]	[2.571,3.429,4.286,5.143]	[4.286,5.143]	
Δk_p,	[1.429,2.857,4.286,5.714]	[4.286,5.714,7.143,8.571]	[7.143,8.571]	

Table 8. Values of fuzzy PID rules Ki based on SLSO.

Name	NB	NM	NS	Z
e(t)	[−5.143,−4.286]	[−5.143,−4.286,−3.429,−2.571]	[−3.429,−2.571,−1.714,−0.857]	[−1.714,−0.857,0.857,1.714]
ec(t)	[−5.073,−4.226]	[−5.073,−4.226,−3.339,−2.54]	[−3.339,−2.54,−1.614,−0.757]	[−1.614,−0.757,0.757,1.614]
Δk_i	[−9.667,−7.0]	[−9.667,−7.0,−6.334,−3.666]	[−6.334,−3.666,−3.0,−0.333]	[−3.0,−0.333,0.333,3.0]
Name	PS	PM	PB	
e(t)	[0.857,1.714,2.571,3.429]	[2.571,3.429,4.286,5.143]	[4.286,5.143]	
ec(t)	[0.757,1.614,2.54,3.339]	[2.54,3.339,4.226,5.073]	[4.226,5.073]	
Δk_i,	[0.333,3.0,3.666,6.334]	[3.666,6.334,7.0,9.667]	[7.0,9.667]	

Table 9. Values of fuzzy PID rules Kd based on SLSO.

Name	NB	NM	NS	Z
e(t)	[−5.8,−4.2]	[−5.8,−4.2,−3.832,−2.232]	[−3.832,−2.232,−1.832,−0.232]	[−1.832,−0.232,0.232,1.832]
ec(t)	[−5.76,−3.84]	[−5.76,−3.84,−3.36,−1.44]	[−3.36,−1.44,−0.96,0.96]	[−1.44,−0.96,0.96,1.44]
Δk_d	[−9.667,−7.0]	[−9.667,−7.0,−6.334,−3.666]	[−6.334,−3.666,−3.0,−0.3333]	[−3.0,−0.333,0.333,3.0]
Name	PS	PM	PB	
e(t)	[0.2,1.832,2.232,3.832]	[2.232,3.832,4.2,5.8]	[4.2,5.8]	
ec(t)	[−0.96,0.96,1.44,3.36]	[1.44,3.36,3.84,5.76]	[3.84,5.76]	
Δk_d,	[0.333,3,3.0,3.666,6.334]	[3.666,6.334,7.0,9.667]	[7.0,9.667]	

Table 10. Values of fuzzy PID rules without optimization.

Name	NB	NM	NS	Z
e(t)	[−5.25,−4.5]	[−5.25,−4.5,−3.75,−3]	[−3.75,−3,−2.25,−1.5]	[−2.25,−0.75,0.75,2.25]
ec(t)	[−5.15,−4.3]	[−5.15,−4.3,−3.44,−2.58]	[−3.44,−2.58,−1.72,−0.86]	[−1.72,−0.86,0.86,1.72]
Δk_p	[−8.58,−7.15]	[−8.58,−7.15,−5.72,−4.28]	[−5.72,−4.29,−2.86,−1.43]	[−2.86,−1.43,1.43,2.86]
Name	PS	PM	PB	
e(t)	[1.5,2.25,3,3.75]	[3,3.75,4.5,5.25]	[4.5,5.25]	
ec(t)	[0.86,1.72,2.58,3.44]	[2.58,3.44,4.3,5.15]	[4.3,5.15]	
Δk_p,	[1.43,2.86,4.29,5.72]	[4.29,5.72,7.15,8.58]	[7.15,8.58]	

The clear values of fuzzy PID rules are shown in Tables 7–9. In addition, the values of fuzzy PID without optimization are shown in Table 10, where parameters are the average points of the interval. Because the values of fuzzy PID without optimization are set by average, we just show the values of Kp; the values of Ki and Kd are the same as Kp.

Introducing the parameters obtained by the SLSO algorithm, the comparative simulation experiment is implemented, and the simulation results are shown below.

From Figures 9–14, we can see that, compared to PID without optimized parameters, fuzzy PID without targeted configuration of fuzzy rules has advantages over ordinary PID in swing-angle control, but its anti-swing performance is worse than the PID-DE. In terms of distance control, its oscillation amplitude is larger than that of ordinary PID. This can prove that fuzzy control lacks usability without parameter optimization.

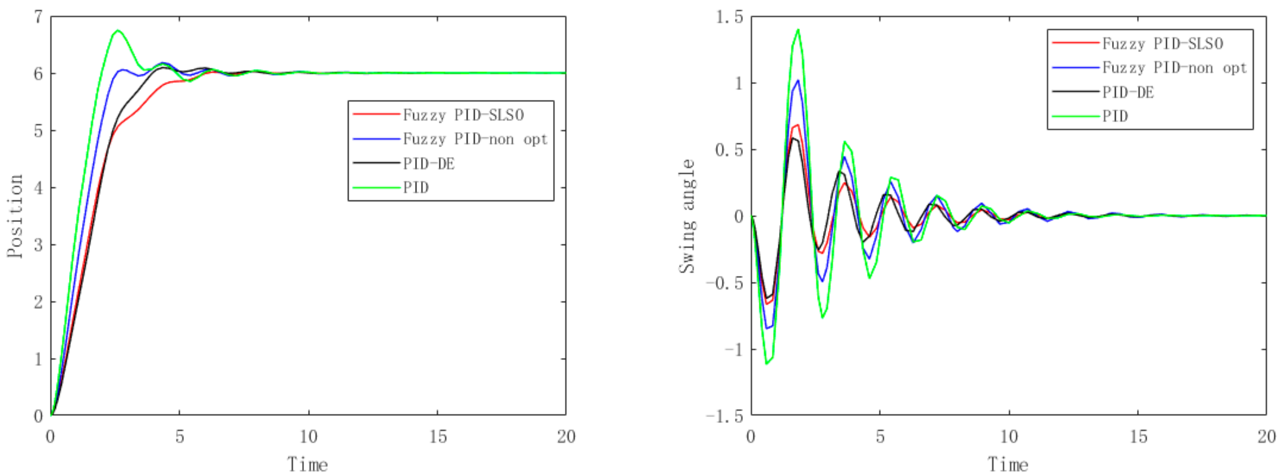

Figure 9. The results of condition 1.

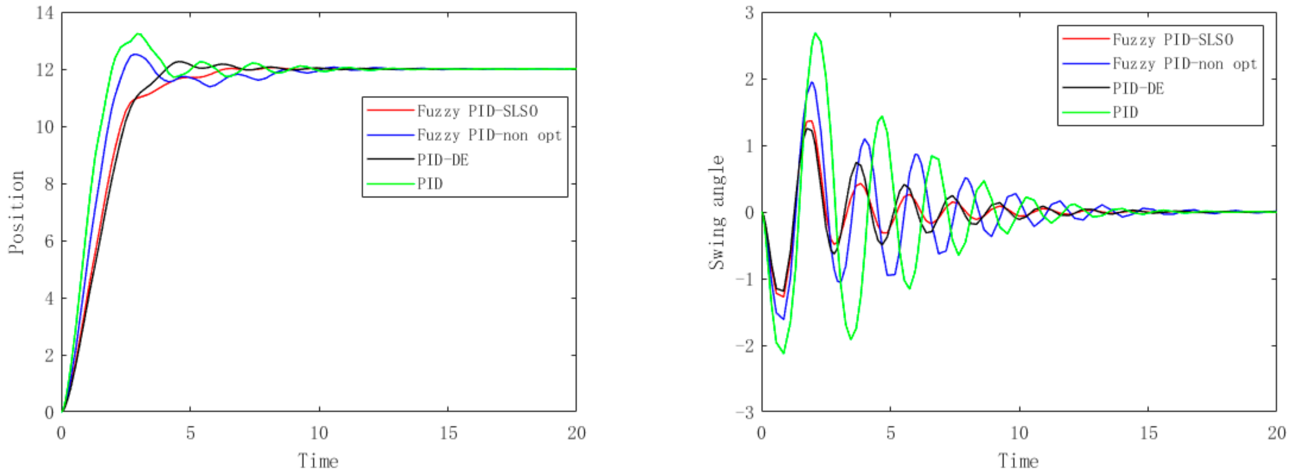

Figure 10. The results of condition 2.

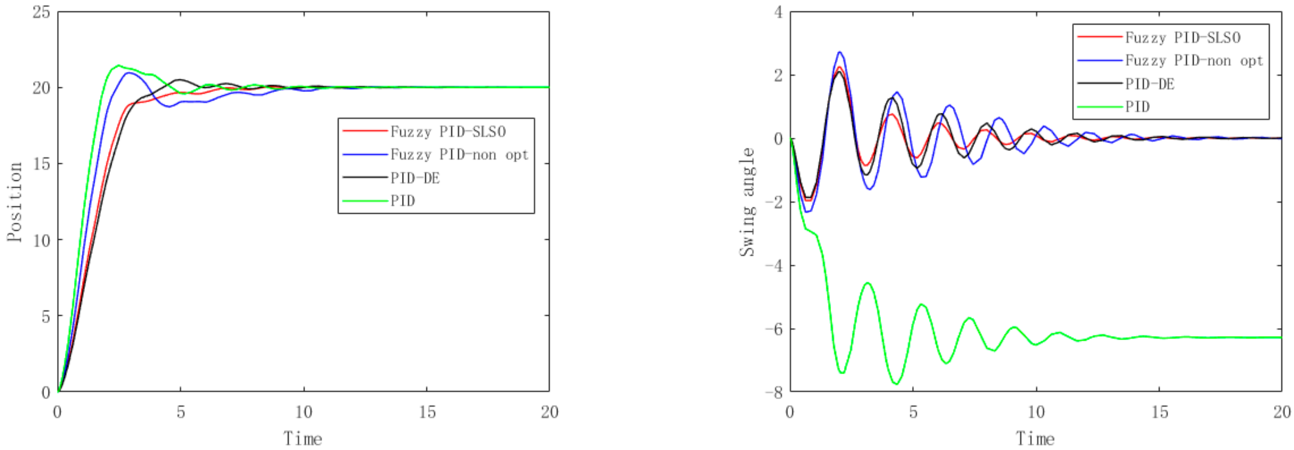

Figure 11. The results of condition 3.

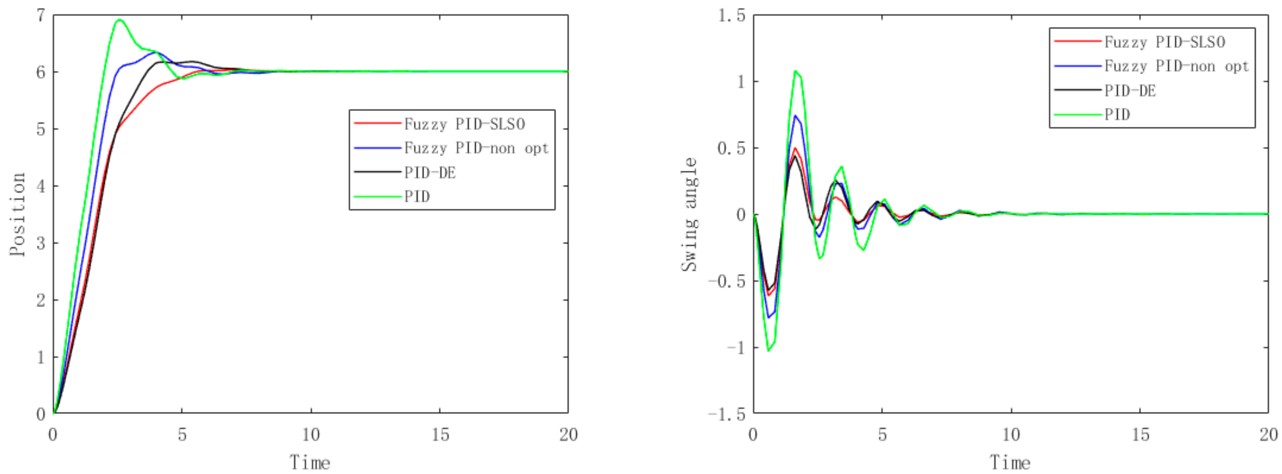

Figure 12. The results of condition 4.

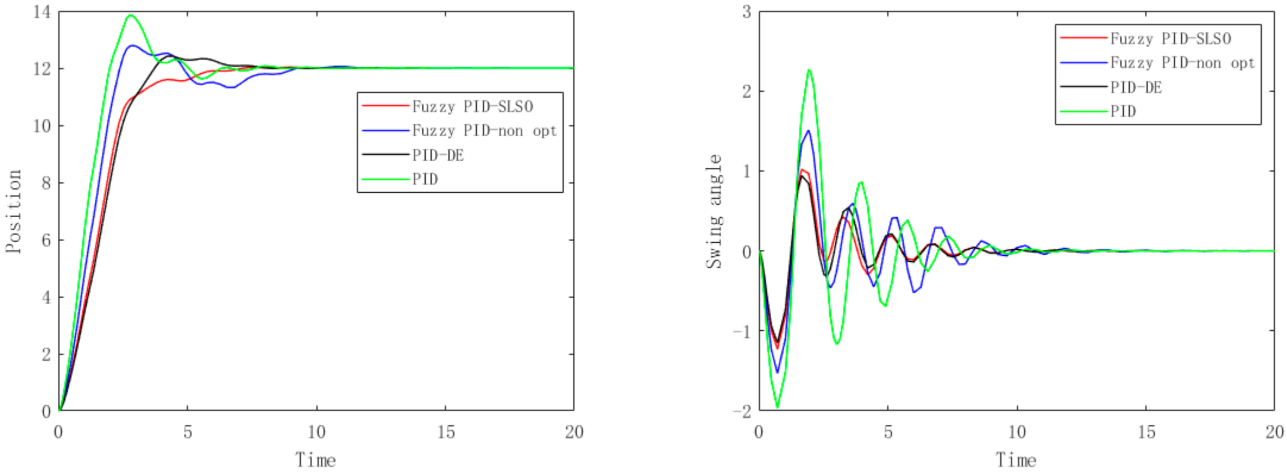

Figure 13. The results of condition 5.

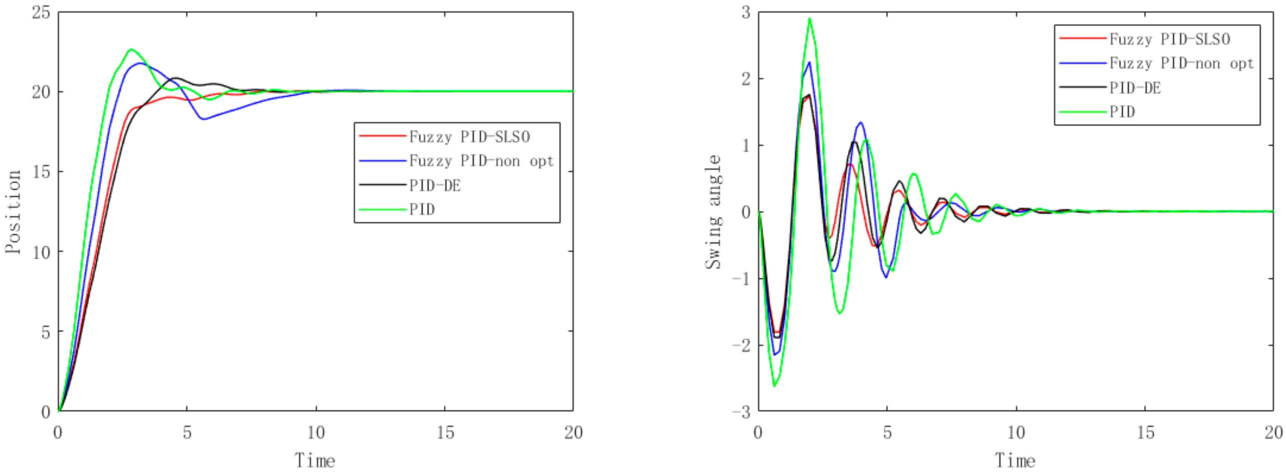

Figure 14. The results of condition 6.

However, in the fuzzy PID, where only basic rules are specified and specific parameters are optimized by the algorithm, its distance control completely exceeds the PID-DE. Most intuitively, the distance of the fuzzy PID almost does not exceed the maximum distance, which is very important in practical applications, meaning that collisions will not occur.

At the same time, the swing angle of the fuzzy PID is also well controlled, which means that the suspended object can reach the endpoint in a very stable attitude.

We conducted comparative experiments on both distance and counterweight dimensions, and the experimental results showed that in both cases, the fuzzy PID control based on SLSO parameter configuration can achieve a good anti-swing effect.

6. Conclusions

In this paper, an SLSO-based fuzzy PID controller is designed to suppress the swing of load during the operation of an overhead crane. To configure the parameter effectively, a modified lion swarm algorithm, which is based on the stray strategy, verifies the effectiveness of the improvement on several functions. By implementing simulation experiments and compared to other adaptive PID methods, the proposed method can dampen the load angle amplitude and residual swing. More precisely, in distance control, the percentages of invalid distance for the four methods of fuzzy PID-SLSO, fuzzy PID-non-optimization, PID-DE, and PID are 3.31%, 35.83%, 10.85%, and 37.03%, respectively. In addition, in the swing control, the swing angle of PID is set to 1, and the swing amplitudes of the four methods are 52.87%, 79.63%, 66.37%, and 100%, respectively. The numerical results show that the fuzzy PID-SLSO algorithm proposed in this paper has an excellent anti-swing control effect in an overhead crane system. This proposed method can also be applied to other under-actuated control systems, such as inverted pendulum systems, pendulum robots, and autonomous surface vehicles.

Author Contributions: Conceptualization, J.F. and Z.S.; methodology, Z.S.; software, D.X.; validation, J.L. and J.F.; formal analysis, Z.S.; investigation, J.F. and J.L.; resources, J.L.; data curation, D.X.; writing—original draft preparation, J.F.; writing—review and editing, Z.S.; visualization, D.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (grant nos. 61972208, 62272239, and 62022044) and the National Natural Science Foundation of Jiangsu Province (grant nos. BK20201043).

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request. Some models and codes used during the study are proprietary or confidential in nature.

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

References

1. Kuo, K.Y.; Linb, J. Fuzzy logic control for flexible link robot arm by singular perturbation approach. *Appl. Soft Comput.* **2003**, *2*, 24–38. [CrossRef]
2. Lu, Y. Adaptive-Fuzzy Control Compensation Design for Direct Adaptive Fuzzy Control. *IEEE Trans. Fuzzy Syst.* **2018**, *26*, 3222–3231. [CrossRef]
3. Ma, M.; Wang, T.; Qiu, J.; Karimi, H.R. Adaptive Fuzzy Decentralized Tracking Control for Large-Scale Interconnected Nonlinear Networked Control Systems. *IEEE Trans. Fuzzy Syst.* **2021**, *29*, 3186–3191. [CrossRef]
4. Chiou, Y.C.; Lan, L.W. Genetic fuzzy logic controller: An iterative evolution algorithm with new encoding method. *Fuzzy Sets Syst.* **2005**, *152*, 617–635. [CrossRef]
5. Jiang, H.; An, T.; Ma, B.; Li, Y.; Dong, B. Value Iteration-based Decentralized Fuzzy Optimal Control of Modular Reconfigurable Robots via Adaptive Dynamic Programming. In Proceedings of the 2022 5th International Conference on Robotics, Control and Automation Engineering (RCAE), Changchun, China, 28–30 October 2022; pp. 186–190.
6. Sabir, M.M.; Ali, T. Optimal PID controller design through swarm intelligence algorithms for sun tracking system. *Appl. Math. Comput.* **2016**, *274*, 690–699. [CrossRef]
7. Ismayil, C.; Kumar, R.S.; Sindhu, T.K. Optimal fractional order PID controller for automatic generation control of two—Area power systems. *Int. Trans. Electr. Energy Syst.* **2016**, *25*, 3329–3348. [CrossRef]
8. Zamani, A.A.; Tavakoli, S.; Etedali, S. Fractional order PID control design for semi-active control of smart base-isolated structures: A multi-objective cuckoo search approach. *Isa Trans.* **2017**, *67*, 222. [CrossRef]
9. Garrido, J.; VãZquez, F.; Morilla, F. Multivariable PID control by decoupling. *Int. J. Syst. Sci.* **2016**, *47*, 1054–1072. [CrossRef]
10. Xie, M.; Li, X.; Wang, Y.; Liu, Y.; Sun, D. Saturated PID Control for the Optical Manipulation of Biological Cells. *IEEE Trans. Control Syst. Technol.* **2017**, *26*, 1909–1916. [CrossRef]
11. Gao, P.; Wang, Z.; Zhang, Y.; Li, M. Prediction System for Overhead Cranes Based on Digital Twin Technology. *Appl. Sci.* **2023**, *13*, 4696. [CrossRef]
12. Lisperguier, N.; López, Á.; Vielma, J.C. Seismic Performance Assessment of a Moment-Resisting Frame Steel Warehouse Provided with Overhead Crane. *Materials* **2023**, *16*, 2815. [CrossRef] [PubMed]
13. Ungureanu, M.; Medan, N.; Ungureanu, N.S.; Pop, N.; Nadolny, K. Tribological Aspects Concerning the Study of Overhead Crane Brakes. *Materials* **2022**, *15*, 6549. [CrossRef] [PubMed]
14. Mustapää, T.; Tunkkari, H.; Taponen, J.; Immonen, L.; Heeren, W.; Baer, O.; Brown, C.; Viitala, R. Secure Exchange of Digital Metrological Data in a Smart Overhead Crane. *Sensors* **2022**, *22*, 1548. [CrossRef] [PubMed]
15. Shi, Q.; Lam, H.K.; Xuan, C.; Chen, M. Adaptive Neuro-Fuzzy PID Controller based on Twin Delayed Deep Deterministic Policy Gradient Algorithm. *Neurocomputing* **2020**, *402*, 183–194. [CrossRef]
16. Wu, W.; Gong, G.; Chen, Y.; Zhou, X. Performance Analysis of Electro-Hydraulic Thrust System of TBM Based on Fuzzy PID Controller. *Energies* **2022**, *15*, 959. [CrossRef]
17. Sun, L.; Ma, J.; Yang, B. Fuzzy PID Design of Vehicle Attitude Control Systems. In Proceedings of the 2020 Chinese Control and Decision Conference (CCDC), Hefei, China, 22–24 August 2020; pp. 1826–1830.
18. Sun, Z.; Ling, Y.; Sun, Z.; Bi, Y.; Tan, S.; Ding, L. Designing and Application of Fuzzy PID Control for Overhead Crane Systems. In Proceedings of the 2019 2nd International Conference on Information Systems and Computer Aided Education (ICISCAE), Dalian, China, 28–30 September 2019; pp. 411–414.
19. Yazdani, M.; Jolai, F. Lion Optimization Algorithm (LOA): A Nature-Inspired Metaheuristic Algorithm. *J. Comput. Des. Eng.* **2016**, *3*, 24–36. [CrossRef]

20. Almezeini, N.; Hafez, A. Task Scheduling in Cloud Computing using Lion Optimization Algorithm. *Int. J. Adv. Comput. Sci. Appl.* **2017**, *8*, 77–83. [CrossRef]
21. Zhang, D.Q.; Jiang, M.Y. Parallel discrete lion swarm optimization algorithm for solving traveling salesman problem. *J. Syst. Eng. Electron.* **2020**, *31*, 751–760.
22. Qu, S.; Dou, Y.; Wang, Y.; Sun, R.; Liu, J.; Yang, W. Path Planning of Electric Power Inspection Robot Based on Improved Lion Swarm Algorithm. In Proceedings of the 2021 IEEE 5th Conference on Energy Internet and Energy System Integration (EI2), Taiyuan, China, 22–24 October 2021; pp. 3335–3339.
23. Ji, F.; Jiang, M. Tabu Annealing Lion Swarm Optimization Algorithm. In Proceedings of the 2021 International Conference on Computer Engineering and Artificial Intelligence (ICCEAI), Shanghai, China, 27–29 August 2021; pp. 422–426.
24. Wang, Z.; Wang, Q.; He, D.; Liu, Q.; Zhu, X.; Guo, J. An Improved Particle Swarm Optimization Algorithm Based on Fuzzy PID Control. In Proceedings of the 2017 4th International Conference on Information Science and Control Engineering (ICISCE), Changsha, China, 21–23 July 2017; pp. 835–839.
25. Fu, J.; Wang, N.; Zhao, J.H.; Zang, S.C. A Membrane Computing Optimization Algorithm with Multi-Subsystem for Parameter Estimation of Heavy Oil Thermal Cracking Model. *Int. J. Intell. Robot. Appl.* **2022**, *1*, 139–151. [CrossRef]
26. Wang, N.; Wang, D.X.; Xing, Y.Z.; Shao, L.M.; Afzal, S. Application of co-evolution RNA genetic algorithm for obtaining optimal parameters of SOFC model. Renew. *Energy* **2020**, *150*, 221–233.
27. Fu, J.; Zhao, J.H.; Yu, L.D. Self-adaptive membrane computing algorithm and its application in ABS system. *Con. Eng. China* **2019**, *26*, 155–161.
28. Zhu, X.H.; Wang, N. Cuckoo search algorithm with membrane communication mechanism for modeling overhead crane systems using RBF neural networks. *Appl. Soft. Comput.* **2017**, *56*, 458–471. [CrossRef]
29. Debbah, A.; Kherfane, H.; Kelaiaia, R. Gas Turbine Aerodynamics Improvement Via a Design of Intelligent Fractional Control. *Stroj. Časopis J. Mech. Eng.* **2021**, *71*, 85–100.
30. Liu, S.; Yang, Y.; Zhou, Y. A swarm intelligence algorithm-lion swarm optimization. *Pattern Recognit. Artif. Intell.* **2018**, *31*, 431–441.
31. Wu, D.H.; Kong, F.; Ji, Z.C. Convergence Analysis of Chicken Swarm Optimization Algorithm. *J. Cent. South Univ.* **2017**, *48*, 2105–2112. [CrossRef]
32. Solis, F.J.; West, J.B. Minimization by Random Search Techniques. *Math. Oper. Res.* **1981**, *6*, 19–30. [CrossRef]
33. Zhang, W.X.; Liang, Y. *Mathematical Foundation of Genetic Algorithm*; Xi'an Jiaotong University Press: Xi'an, China, 2003.
34. Sun, Z.; Wang, N.; Bi, Y.; Zhao, J. A DE based PID controller for two dimensional overhead crane. In Proceedings of the 2015 34th Chinese Control Conference (CCC), Hangzhou, China, 28–30 July 2015; pp. 2546–2550.