

Article

Research on Active Disturbance Rejection Control with Parameter Autotuning for a Moving Mirror Control System Based on Improved Snake Optimization

Liangjie Zhi ^{1,2,3} , Min Huang ^{1,2,3}, Lulu Qian ^{1,3}, Zhanchao Wang ^{1,3} , Qin Wen ^{1,3} and Wei Han ^{1,3,*}

¹ Aerospace Information Research Institute, Chinese Academy of Sciences, No. 9 Dengzhuang South Road, Haidian District, Beijing 100094, China; zhiliangjie20@mailsucas.ac.cn (L.Z.)

² School of Optoelectronics, University of Chinese Academy of Sciences, No. 19(A) Yuquan Road, Shijingshan District, Beijing 100049, China

³ Department of Key Laboratory of Computational Optical Imagine Technology, Chinese Academy of Sciences, No. 9 Dengzhuang South Road, Haidian District, Beijing 100094, China

* Correspondence: hw@aircas.ac.cn; Tel.: +86-157-6210-7786

Abstract: In order to improve the control of a moving mirror control system and enhance the anti-interference ability of the system, active disturbance rejection control (ADRC) with parameter autotuning is proposed and applied to control a rotary voice coil motor (RVCM). Improved snake optimization (I-SO) was applied to tune and optimize ADRC's key parameters. To obtain excellent parameters efficiently, in the population initialization phase of SO, the quality and diversity of initial solutions were improved through a chaotic elite opposition learning algorithm. In the local search phase, a sine and cosine (SC) search mode was introduced to enhance the local search ability of SO. The simulation results show that I-SO can effectively find the ideal parameters. I-SO has excellent search capability and stability. The experimental control system of a moving mirror was established, and the effectiveness of the parameters optimized by I-SO was verified. ADRC with parameter autotuning showed excellent control in the moving mirror control system, and the stability of the optical path scanning speed reached 99.2%.

Keywords: active disturbance rejection control; rotary voice coil motor; snake optimization; parameter autotuning; chaotic elite opposition learning; sine and cosine search mode



Citation: Zhi, L.; Huang, M.; Qian, L.; Wang, Z.; Wen, Q.; Han, W. Research on Active Disturbance Rejection Control with Parameter Autotuning for a Moving Mirror Control System Based on Improved Snake Optimization. *Electronics* **2024**, *13*, 1650. <https://doi.org/10.3390/electronics13091650>

Academic Editor: Davide Astolfi

Received: 19 March 2024

Revised: 16 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/>).

1. Introduction

As the core component of the Fourier-transform infrared spectrometer (FTIR), the Michelson interferometer generates an optical path difference by means of a moving mirror [1]. Therefore, the control accuracy of the moving mirror control system determines the performance of the FTIR [2]. Today, swing-arm interferometers have the advantage of high stability and are widely used in FTIR [3]. Moving mirrors of swing-arm interferometers are mostly driven by rotary voice coil motors (RVCMs). RVCMs are special motors that only work in a limited angle range, and they are commonly used for reciprocating drives with small inertial load [4]. Compared with traditional motors, RVCMs have the exceptional advantages of small dimensions, high accuracy, large thrust, and fast response [5]. Therefore, RVCMs are widely used in the industrial, aerospace, and precision instrument fields [6–9].

However, in RVCM-based control systems, external disturbances act directly on the motor. External random disturbances (such as vibration disturbances) are not considered in the mechanical balance equation of the model, and parameter perturbations are not considered in the electrical balance equation. These disturbances and parameters change as the system runs for a long time and is affected by factors such as temperature, device aging, and working environment [10]. These uncertainties lead to poor robustness and weak anti-interference ability of the control system. In recent years, with the rapid

development of control theory, some advanced control methods have been proposed to improve the performance of VCM-based control systems, such as the control method with a disturbance observer [11], a position estimator [12], adaptive control [13,14], sliding mode control [15–17], active disturbance rejection control (ADRC) [18], and other advanced control technologies.

ADRC is a nonlinear control method that does not depend on the precise mathematical model of the controlled object. It can estimate the total disturbance of the system in real time and then compensate the system to improve its anti-interference ability [19]. At present, ADRC is widely used in the field of motor control. In [20], it is verified that ADRC can effectively observe the bounded time-varying disturbance of voice coil motor control systems. In [21], linear ADRC is used to reduce the calculation load of the controller and is suitable for real-time control systems. In [22], improved ADRC applied to high-precision positioning workbench is proposed, which effectively improves the positioning accuracy and robustness of the system. The majority of the above ADRC studies focus on the controller's structure rather than how to adjust its parameters, which limits the promotion and application of ADRC to a certain extent.

Conventional ADRC has many parameters, and each parameter directly and significantly affects the controller's performance. To fully exploit the advantages of ADRC, its parameters must be properly adjusted. The manual adjustment of ADRC parameters requires the expertise of engineers, which is time-consuming and labor-intensive, so it is difficult to achieve ideal control. Second, the range of ADRC parameters varies as a result of the various controlled objects. Manual adjustment is not conducive to the widespread application of ADRC. Therefore, it is necessary to find a parameter tuning method that is convenient for application.

To address this issue, numerous scholars have proposed the use of heuristic algorithms to tune the parameters of ADRC. In [23], genetic algorithm (GA)-based parameter tuning ADRC was proposed to improve the stabilization accuracy of the three-axis inertial platform. However, since the GA is prone to premature convergence and has poor local search capabilities, it can only find suboptimal solutions, as opposed to the optimal one. Particle swarm optimization (PSO) is also a commonly used parameter optimization algorithm. However, the search ability of PSO is weaker in the later stage of the iterations, and it can easily fall into local optima. In [24], an improved PSO was proposed to balance the global and local search abilities of PSO. Reference [25] proposed using the butterfly optimization algorithm to tune the parameters of ADRC to achieve the best performance of the controller. In [26], ant colony optimization (ACO) was applied to the control of current controllers. However, setting ACO parameters is complex, and it is easy to deviate from the high-quality solution if the parameters are not set properly. Compared with ACO, gray wolf optimization (GWO) has the advantages of a simple structure and fewer parameters [27]. However, GWO has the drawbacks of low accuracy and slow convergence when facing complex problems. The chimp optimization algorithm suffers from the same shortcomings [28]. In [29], a whale optimization algorithm (WOA) was applied to fine-tune the parameters of ADRC. However, the WOA easily falls into local optima, and its optimization ability is weak. The sparrow search algorithm (SSA) has also been used in ADRC parameter tuning [30,31]. It has strong local search ability and fast convergence, but its global search ability is weak and it does not easily escape the local optimum. In addition, some hybrid algorithms have also been proposed, like the algorithm composed of differential evolution and PSO [32] and that composed of PSO and bacterial foraging optimization (BFO) [33].

The snake optimization (SO) algorithm is a heuristic algorithm that simulates the feeding and breeding behavior of snakes [34]. This algorithm has the advantages of a strong global search ability, high efficiency, and fast convergence. As a reference, the SO algorithm was tested using 10 functions at the Congress on Evolutionary Computation (CEC) 2020. The SO algorithm performed poorly on tests of four functions due to its weak local search capabilities in the later stages of the search. To address this issue, this study

designed an improved SO algorithm (I-SO). It adopts the chaotic elite opposition learning algorithm and introduces the sine and cosine (SC) search mode, which enhances the global search capability of SO and its ability to escape local optima.

In this paper, to improve the control of ADRC for a moving mirror control system and eliminate the time-consuming and labor-intensive process of manual parameter adjustment, a novel ADRC system with parameter autotuning based on I-SO is proposed. ADRC with parameter autotuning based on I-SO is more robust than conventional ADRC. As a population-based optimization algorithm, I-SO does not rely on specific mathematical models, thus enhancing the applicability of ADRC with parameter autotuning based on I-SO in the control field. The simulation results show the feasibility and the effectiveness of the proposed method.

The rest of this paper is organized as follows. Section 2 describes the structure of the moving mirror system and the mathematical model of an RVCM. The principle of ADRC is also explained in this section. Section 3 provides a brief overview of the original SO and describes the proposed I-SO in detail. Further, Section 3 describes the implementation process of I-SO applied in ADRC. Section 4 demonstrates the superiority of I-SO and verifies the advantages of I-SO by comparison with various optimization algorithms. Section 5 simulates a moving mirror control system and verifies the effectiveness of the ADRC with parameter autotuning based on I-SO. Section 6 presents the conclusions of this research.

2. Mathematical Model of the RVCM and Principle of ADRC

2.1. Mathematical Model of the RVCM

This study designed a swing-arm interferometer with dual cube-corner mirrors. Its three-dimensional structure diagram and optical path diagram are shown in Figure 1a,b, respectively. The swing of the two cube-corner mirrors is driven by an RVCM.

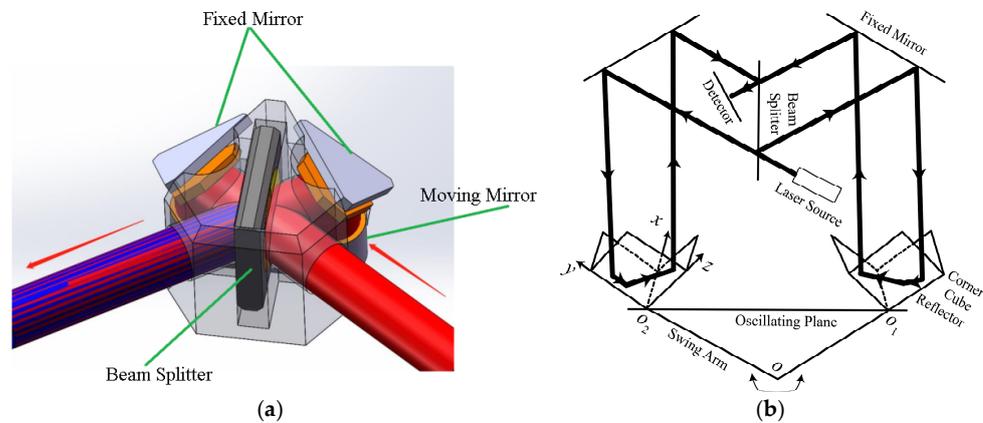


Figure 1. Swing-arm interferometer with dual cube-corner mirrors: (a) 3D structure diagram; (b) optical path diagram.

Under ideal conditions, the transfer function $G(s)$ between θ and u of the RVCM can be obtained as per [35]:

$$G(s) = \frac{\theta(s)}{u(s)} = \frac{k_a}{LJs^3 + (JR + Lk)s^2 + (k_a^2 + kR)s + Rb} \tag{1}$$

where J is the moment of inertia of the rotor, k is the friction damping coefficient, k_a is the torque coefficient, u is the voltage of the motor armature, i , L , and R are the current, inductance, and resistance of the motor armature, respectively, and θ is the angular displacement of the motor rotor of RVCM.

During normal operation, the inductance value of RVCM is negligible. The specific transfer function $G(s)_s$ of RVCM can be obtained by means of frequency sweeping:

$$G(s)_s = \frac{0.36}{0.23s^2 + 6.29s + 0.07} \tag{2}$$

The model shown in Equation (2) is obtained under ideal conditions. The RVCM is interfered with by a variety of nonlinear factors during motion control, such as friction, mechanical resonance and hysteresis characteristics. Therefore, there are inevitable modeling errors between the mathematical model and the actual system. To address this problem, an ADRC algorithm was used to control RVCM in this study. The overall block diagram of the control system is shown in Figure 2. The ADRC algorithm is composed of a tracking differentiator (TD), a linear extended state observer (LESO), and state error feedback control law (SEF).

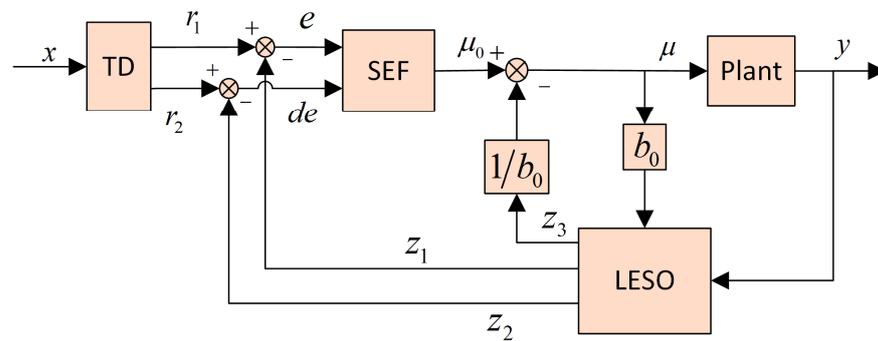


Figure 2. Overall block diagram of the control system.

2.2. Tracking Differentiator

For the jump behavior of the command signal, TD arranges a transition process for the command signal and reduces the initial impact of the signal on the system. A discrete mathematical formula of TD that is different from traditional TD is given in [36]:

$$\begin{cases} r_1(k+1) = r_1(k) + hr_2(k) + 0.5h^2 f_c(r_1(k) - x(k), r_2(k), r, h) \\ r_2(k+1) = r_2(k) + hf_c(r_1(k) - x(k), r_2(k), r, h) \end{cases} \tag{3}$$

where h is the sampling period, $x(k)$ is the input signal at the k th time, $r_1(k)$ is the tracking signal of $x(k)$, $r_2(k)$ is the derivative of $r_1(k)$, r is a parameter that affects the tracking speed, and $f_c(\cdot)$ is an improved discrete form of the fastest control comprehensive function, and its solution is shown in Equation (4):

$$\begin{cases} y = 2r_1 + hr_2, s = \text{sgn}(y) \\ k_0 = \frac{1}{2}(\sqrt{1 + \frac{4|y|}{h^2 r}} - 1) \\ k = \text{fix}(k_0) + 1 \\ f_c(r_1(k) - x(k), r_2(k), r, h) = -\text{sat}(\frac{r_2}{h} + \frac{(k-1)rs}{2} + \frac{y}{2kh^2}, r) \end{cases} \tag{4}$$

where $\text{sgn}(\cdot)$ is the sign function, $\text{fix}(\cdot)$ is the rounding function, and the saturation function $\text{sat}(\cdot)$ is defined as:

$$\text{sat}(x_1, \delta) = \begin{cases} \delta \text{sgn}(x), & |x| > \delta \\ x, & |x| \leq \delta \end{cases} \tag{5}$$

Compared with the discrete form of the traditional fastest control synthesis function, this has more concise expression and reduces the phase delay of the output signal.

2.3. Linear Extended State Observer

LESO can observe the position, speed, and unknown parts of the controlled object and compensate for the unknown uncertainty and external system disturbance by estimating the total disturbance of the system. Compared with NLESO, LESO has fewer parameters, which reduces the calculation load of the controller and is more suitable for real-time control systems. The discrete mathematical formula of LESO is as follows:

$$\begin{cases} e(k) = z_1(k) - y(k) \\ z_1(k+1) = z_1(k) + h(z_2(k) - \beta_{01}e(k)) \\ z_2(k+1) = z_2(k) + h(z_3(k) - \beta_{02}e(k)) + b_0u(k) \\ z_3(k+1) = z_3(k) - h\beta_{03}e(k) \end{cases} \quad (6)$$

where $y(k)$ is the feedback signal from the controlled object, $z_1(k)$ is the estimated value of $r_1(k)$ in the TD, $z_2(k)$ is the estimated value of $r_2(k)$, and $z_3(k)$ is the estimated value of the total system disturbance $f(k)$. There are three important parameters in the LESO: β_{01} , β_{02} and β_{03} . β_{01} and β_{02} affect the estimated value of the state $r_1(k)$ and $r_2(k)$, respectively, while β_{03} affects the estimated value of total disturbance. In addition, b_0 is also a key parameter that will affect LESO and b_0 affects the value of the compensation.

2.4. State Error Feedback Control Law

In this paper, the SEF is a PD controller with disturbance compensation term. The discrete mathematical formula is given as follows:

$$\begin{cases} e_1(k) = r_1(k) - z_1(k), e_2(k) = r_2(k) - z_2(k) \\ u_0(k) = k_p e_1(k) + k_d e_2(k) \\ u(k) = u_0(k) - z_3(k)/b_0 \end{cases} \quad (7)$$

where e_1 and e_2 represent the error of position and speed, respectively, while k_p and k_d are the parameters of the controller. In order to facilitate parameter tuning, the parameters in Equation (7) can be defined as per [37]:

$$k_p = \omega_c^2, k_d = 2\zeta\omega_c \quad (8)$$

where ω_c represents the bandwidth of the controller. In order to further improve the control of the closed-loop system, the damping ratio coefficient ζ is introduced in the parameter setting process to reduce the oscillation of the closed-loop system.

Since there are many important parameters in ADRC, manually adjusting them based on expertise is a time-consuming and labor-intensive process. Additionally, the controller does not always perform at its peak efficiency when manually adjusting the parameters empirically. In this paper, the I-SO algorithm is used to tune the important and key parameters β_{01} , β_{02} , β_{03} and ω_c to achieve ideal performance.

3. ADRC with Parameter Autotuning Based on the I-SO Algorithm

3.1. Basic Principles of SO

The SO algorithm is inspired by the mating behavior of snakes. If the temperature is low and food is available, snakes will try to find the best mate and then mate. Otherwise, they will only look for food or eat existing food. Each snake has a position X in the search space, which is modified by looking for food, fighting, and mating to find the globally optimal position. Therefore, the mechanism of SO consists of the exploration phase without food and the exploitation phase of existing food, explained as follows.

Before exploring, the temperature of the environment, the food quantity in the environment, and the population of snakes are first defined.

The temperature $temp$ is defined as:

$$temp = \exp\left(\frac{-t}{T}\right) \quad (9)$$

where t is the current number of iterations and T is the maximum number of iterations. The food quantity Q is defined as:

$$Q = c_1 \times \exp\left(\frac{t-T}{T}\right) \quad (10)$$

where c_1 is a constant.

For the snake population, the initialization of snakes is performed by generating a uniformly distributed random population, as shown in Equation (11):

$$X_i = X_{\min} + rand \times (X_{\max} - X_{\min}) \quad (11)$$

where X_i is the position of the i -th individual, X_{\min} and X_{\max} are, respectively, the lower and upper bounds of the environment, and $rand$ is a random number between 0 and 1.

Assuming that the number of females and males in the population of snakes is the same, then:

$$N_m = N_f = N/2 \quad (12)$$

where N_m is the number of males, N_f the number of females, and N the total number of individuals in the snake population.

3.1.1. Exploration Phase without Food

If $Q < \text{threshold}$ (threshold of food quantity), the snakes look for food by selecting any random position and update their position accordingly. The position update equations of male and female snakes are shown in Equation (13):

$$\begin{cases} X_{i,m}(t+1) = X_{rand,m}(t) \pm c_2 \times A_m \times (rand \times (X_{\max} - X_{\min}) + X_{\min}) \\ X_{i,f}(t+1) = X_{rand,f}(t) \pm c_2 \times A_f \times (rand \times (X_{\max} - X_{\min}) + X_{\min}) \end{cases} \quad (13)$$

where $X_{i,m}$ is the i -th male snake position, $X_{i,f}$ is the i -th female snake position, $X_{rand,m}$ and $X_{rand,f}$ refer to the positions of random male and female, respectively, c_2 is a constant, and A_m is the male ability to find food. Correspondingly, A_f is the female ability to find food. These can be calculated using Equation (14):

$$\begin{cases} A_m = \exp(-f_{rand,m}/f_{i,m}) \\ A_f = \exp(-f_{rand,f}/f_{i,f}) \end{cases} \quad (14)$$

where $f_{rand,m}$ is the fitness of $X_{rand,m}$ and $f_{i,m}$ is the fitness of the i -th individual in the male group, and the same can be seen for the female group in Equation (14).

3.1.2. Exploitation Phase of Existing Food

Under the condition of $Q \geq \text{threshold}$, if $temp > \text{threshold}$ (temperature threshold), the environment is in a hot state. The snakes only look for food, and the position update equation is shown in Equation (15):

$$X_{i,j}(t+1) = X_{food} \pm c_3 \times temp \times rand \times (X_{food} - X_{i,j}(t)) \quad (15)$$

where $X_{i,j}$ is the position of an individual, X_{food} is the position of the best individual, and c_3 is a constant.

If $temp < \text{threshold}$ (temperature threshold), the temperature is cold. The snakes will either be in fight mode or in mating mode. The fight mode can be described by Equation (16):

$$\begin{cases} X_{i,m}(t+1) = X_{i,m}(t) + c_3 \times FM \times rand \times (Q \times X_{best,f} - X_{i,m}(t)) \\ X_{i,f}(t+1) = X_{i,f}(t) + c_3 \times FF \times rand \times (Q \times X_{best,m} - X_{i,f}(t)) \end{cases} \quad (16)$$

where $X_{i,m}$ is the i -th male position, $X_{best,f}$ refers to the position of the best individual in the female group, c_3 is a constant, and FM is the fighting ability of the male snake. Similarly, $X_{i,f}$ and $X_{best,m}$ can be understood. FM and FF can be obtained from Equation (17):

$$\begin{cases} FM = \exp(-f_{best,f}/f_i) \\ FF = \exp(-f_{best,m}/f_i) \end{cases} \quad (17)$$

where $f_{best,f}$ is the fitness of the best snake of the female group, $f_{best,m}$ is the fitness of the best snake of the male group, and f_i is the fitness of the i -th snake position.

The mating mode can be described by Equation (18):

$$\begin{cases} X_{i,m}(t+1) = X_{i,m}(t) + c_3 \times M_m \times rand \times (Q \times X_{i,f}(t) - X_{i,m}(t)) \\ X_{i,f}(t+1) = X_{i,f}(t) + c_3 \times M_f \times rand \times (Q \times X_{i,m}(t) - X_{i,f}(t)) \end{cases} \quad (18)$$

where $X_{i,m}$ is the i -th male position and $X_{i,f}$ is the i -th female position in their respective groups. M_m and M_f refer to the mating ability of the males and females, respectively, and they can be calculated as follows:

$$\begin{cases} M_m = \exp(-f_{i,f}/f_{i,m}) \\ M_f = \exp(-f_{i,m}/f_{i,f}) \end{cases} \quad (19)$$

where $f_{i,f}$ is the fitness of the i -th female of the female group and $f_{i,m}$ is the fitness of the i -th male of the male group.

When two snakes complete mating, their egg may or may not hatch. If the egg hatches, the worst male and female are replaced, as described in Equation (20):

$$\begin{cases} X_{worst,m} = X_{min} + rand \times (X_{max} - X_{min}) \\ X_{worst,f} = X_{min} + rand \times (X_{max} - X_{min}) \end{cases} \quad (20)$$

where $X_{worst,m}$ is the worst male individual and $X_{worst,f}$ is the worst female.

3.2. I-SO

3.2.1. Chaotic Elite Opposition Learning

To enhance the global search ability of the SO algorithm, we aimed at improving the quality of its initial solution. In this paper, the chaotic elite opposition learning algorithm is applied to the initialization phase of SO. The chaotic values generated by the tent chaotic map are uniformly distributed in the range of 0–1 [38]. Elite opposition learning exploits elite individuals having more effective information to construct their opposition population and selects multiple excellent individuals from the current population and the opposition population as the initial solution, thus improving the quality of the initial population [39]. The definition of a tent chaotic map is shown in Equation (21):

$$x_{n+1} = \begin{cases} x_n/\alpha, x_n \in [0, \alpha) \\ (1 - x_n)/(1 - \alpha), x_n \in [\alpha, 1] \end{cases} \quad (21)$$

where α is a random number within the range of 0–1.

The distribution and frequency of chaotic values of the tent chaotic map are shown in Figure 3a,b, where it can be seen that the chaotic values generated by the tent chaotic map are uniformly distributed between 0 and 1 and show rich diversity.

Suppose that $s_i = \{s_{i,1}, s_{i,1}, \dots, s_{i,d}\}$ is an elite individual in the search space. The definition of its reverse solution $\tilde{s}_i = \{\tilde{s}_{i,1}, \tilde{s}_{i,1}, \dots, \tilde{s}_{i,d}\}$ is shown in Equation (22):

$$\begin{cases} \tilde{s}_{i,j} = \lambda(lb_j + ub_j) - s_{i,j}, & lb_j \leq \tilde{s}_{i,j} \leq ub_j \\ \tilde{s}_{i,j} = rand(lb_j, ub_j), & \tilde{s}_{i,j} > ub_j \text{ or } \tilde{s}_{i,j} < lb_j \end{cases} \quad (22)$$

where lb_j and ub_j are the lower and upper boundaries of the j dimension search space, respectively, and λ is a random number between 0 and 1.

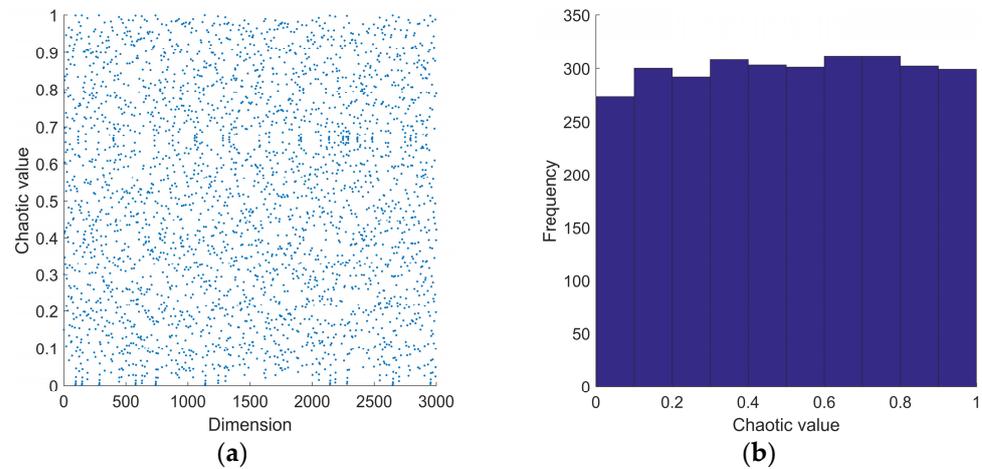


Figure 3. Tent chaotic map: (a) distribution of values; (b) frequency of values.

The steps to initialize the population using the chaotic elite opposition learning algorithm are as follows.

- (1) Use the tent chaotic map to initialize population S , calculate the fitness of individual, and select the first $N/2$ individuals with better fitness to form elite population E .
- (2) Calculate the opposition population OE of elite population E .
- (3) The new population $\{S, OE\}$ is obtained by merging populations S and OE , and the fitness of individuals in the new population is calculated. The final initial population is composed of the first $N_{\{S, OE\}}/2$ individuals with better fitness.

This paper combines a tent chaotic map with the elite opposition learning algorithm to maintain the diversity of the population and improve the quality of the initial population.

3.2.2. SC Search Mode

In the exploration phase of existing food, in order to enhance the ability of the SO algorithm to escape the local optimum, the algorithm introduces a SC search mode [40], which requires the algorithm to use a mathematical model based on sine and cosine functions to fluctuate outward. We modify Equation (15), (16), and (18), and the modified part is shown in Equation (23).

$$rand \rightarrow \begin{cases} \sin(rand(0, 2\pi)) \\ \cos(rand(0, 2\pi)) \end{cases} \quad (23)$$

The position update formula for males introduces the sine function, and for females, the cosine function is introduced.

The chaotic elite opposition learning algorithm improves the quality of the initial solution, and the introduction of the SC search mode enhances the ability of SO to escape the local optimum. These two strategies are helpful for the SO to find the global optimum solution.

3.3. Implementation of I-SO Applied in ADRC

In this paper, I-SO is implemented to optimize the parameters of ADRC in the control system of an RVCM. The schematic diagram of the control system including I-SO is shown in Figure 4. In detail, the tunable and key parameters β_{01} , β_{02} , β_{03} , and ω_c are optimized via I-SO under the consideration of the stability and accuracy of the system. The fitness function determines the changing trend of the snake's position. Therefore, an appropriate fitness function to connect I-SO and ADRC is needed to obtain appropriate parameters for ADRC.

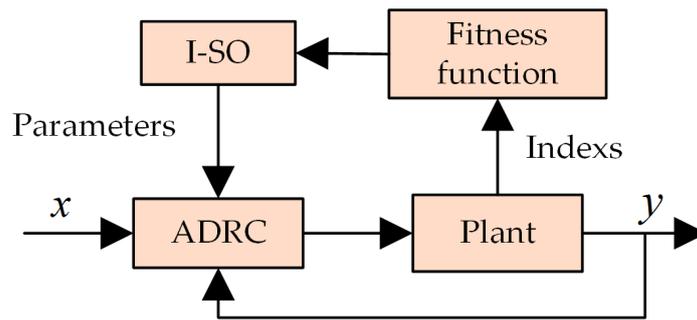


Figure 4. Schematic diagram of control system including I-SO.

Generally, the control indicators of the RVCM and other control systems include stable and dynamic performance indicators. When the input of the system is a step response, the important dynamic performance indicators of the system include overshoot σ and rise time t_r . The steady-state performance indicator is mainly stable error e_{ss} . A system with excellent performance needs to consider stable and dynamic performance.

Integrated time and absolute error (ITAE) is an indicator used to measure system performance. In this paper, the $ITAE_d$ is used to measure the LESO's observation error of total disturbance to the system. The expression of $ITAE_d$ is:

$$ITAE_d = \int_0^T t|e_{z_3}(t)|dt \tag{24}$$

where T is the response period of the system and $e_{z_3}(t)$ is the observation error of the total disturbance to the system. The expression of $e_{z_3}(t)$ is:

$$e_{z_3}(t) = f(t) - z_3(t) \tag{25}$$

Finally, the fitness function Y is composed of the weighted sum of the dynamic and stable control indicators and $ITAE_d$, as shown in Equation (26):

$$Y = a \cdot \sigma + b \cdot t_r + c \cdot e_{ss} + d \cdot ITAE_d \tag{26}$$

where $a, b, c,$ and d are the weights.

The dynamic and stable control indicators and $ITAE_d$ share the same characteristic that the smaller their value is, the better the control of the RVCM that will be achieved. Thus, the ideal parameters can cause the system to have the best control and enable the fitness function Y to reach the minimum. In each iteration, the optimization process of I-SO is the process of moving the position of each snake towards the optimal position.

In the experiment, the detailed optimization process of I-SO computing the optimization parameters $\beta_{01}, \beta_{02}, \beta_{03},$ and ω_c can be described as follows.

Step 1. The parameters in I-SO are initialized, including the number of snakes N , the maximum number of iterations T , the dimension dim , the upper limit and the lower limit of the search space, and thresholds of food quantity and temperature, etc. The position X of each snake in the search space is initialized via the chaotic elite opposition learning algorithm.

Step 2. The snake population is divided into two groups, one for males and the other for females.

Step 3. ADRC is run with the parameters of Step 1, and the RVCM is controlled by the ADRC. The indicators of control of the system are calculated. These indicators are substituted into Equation (26) to calculate the value of the fitness function. The results are compared and the best males and females are defined.

Step 4. The iteration is executed and the food quantity and temperature are calculated according to Equations (9) and (10).

Step 5. The stage of exploring food is entered and the positions of males and females are updated according to Equation (13).

Step 6. When the amount of food is sufficient, the next stage is entered to judge whether the temperature is appropriate. The position of the snake is updated using Equations (15), (16) and (18), which introduce the SC search mode.

Step 7. The stopping criterion is checked. If the number of iterations i is greater than the maximum number of iterations T , the optimal parameters are output and the algorithm is stopped. Otherwise, the process moves to Step 4.

The pseudocode of Algorithm 1 I-SO is as follows.

Algorithm 1 I-SO

```

1: Initialize parameters of I-SO
2: Initialize the snakes using Equation (22)
3: Equally divide the population into two groups
4: while ( $i \leq T$ )
5:     Evaluate the fitness of individuals in each group
6:     Find best male  $X_{best,m}$  and best female  $X_{best,f}$ 
7:     Define Temp using Equation (9).
8:     Define food Quantity Q using Equation (10).
9:     if ( $Q < 0.25$ ) then
10:        Perform exploration using Equation (13).
11:     else
12:        if ( $temp > 0.6$ ) then
13:            Perform exploitation Equations (15) and (23)
14:        else
15:            if ( $rand > 0.6$ ) then
16:                Snakes in Fight Mode Equations (16) and (23)
17:            else
18:                Snakes in Mating Mode Equations (18) and (23)
19:                Change the worst male and female using Equation (20)
20:            end if
21:        end if
22:    end if
23: end while
24: Return best solution.

```

4. Simulation Results and Analysis

To verify the effectiveness of the I-SO, an ADRC model of the RVCM was established using MATLAB, and the key parameters of ADRC were tuned through I-SO. The environment configuration of this simulation experiment was as follows: a 64-bit Windows 10 operating system, an Intel (R) Core (TM) i7-10710u CPU, a main frequency of 1.10 GHz, 16 GB of memory, and simulation software MATLAB R2016b.

According to the commissioning experience, the general parameters of the control system were set as follows:

(1) ADRC:

TD: $r = 1000$, $h_0 = 0.001$; LESO: $b = 0.35$; SEF: $\zeta = 1.25$

In order to test the observation ability of the LESO for the total disturbance of system, a disturbance was introduced into the speed link of the system. The anti-interference ability of the system can be verified by adding different disturbances and noises in different links of the system. When conducting vibration tests on instruments or models in the field or in the laboratory, sinusoidal vibration tests are often used. Introducing sinusoidal periodic disturbances can simulate disturbances in the environment by adjusting the frequency, vibration amplitude, and test duration. Therefore, the expression for the disturbance was set to:

$$d_{nl} = 2 + \sin(4\pi t) \quad (27)$$

(2) SO algorithm: the parameter settings of I-SO algorithm are shown in Table 1.

Table 1. Parameter settings of I-SO algorithm.

No.	Parameter Name	Value
1	Threshold of food quantity	0.48
2	Threshold of temperature	0.6
3	C1	1
4	C2	0.05
5	C3	2
6	Population size	30
7	Dim	4
8	Max number of iterations	30
9	lower limit	{100, 1000, 10,000, 10}
10	upper limit	{10,000, 100,000, 10,000,000, 250}

The minimum value of the fitness function, the convergence speed, and the standard deviation (STD) of multiple different minimum fitness function values obtained after repeatedly running the algorithm need to be considered important indicators of the intelligent optimization algorithm. The minimum value of the fitness function reflects the search ability of the algorithm, and the standard deviation reflects its stability.

The optimization algorithm was run 50 times independently, and the average of 50 convergence speed values was calculated, as well as the average and standard deviation of 50 fitness function values. Among the 50 running results, the operation result of the algorithm whose convergence speed and fitness function value were closest to their respective averages (the average value of the fitness function was first considered) was selected as the operation result of optimization algorithm (each optimization algorithm mentioned below was handled in this same way).

4.1. Verification of Control Performance of ADRC Tuned by I-SO

After selection, we finally determined that the optimal solution output by I-SO algorithm is {178, 49,642, 7,898,000, 145}.

Figure 5 shows the response curves and error curves. Table 2 shows fitness function values and some control indicators. It can be seen from Figure 5 and Table 2 that the parameters calculated by I-SO can enable the simulation system to achieve outstanding dynamic and stable performance, as well as an exceptional response to position and speed signals. When there is no disturbance in the system, the position overshoot is 4.66%, both the position and speed steady-state error are 0, and the rise time is shorter than 0.1 s. When there is disturbance in the system, the system has a small steady-state error of about 0.02%.

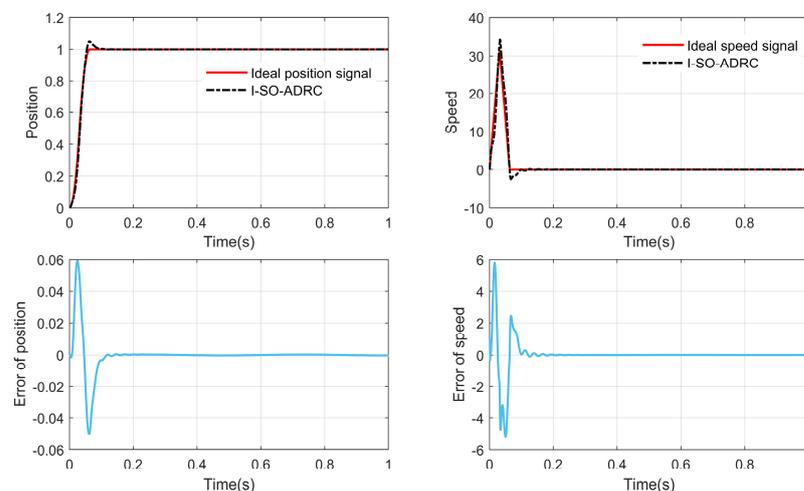


Figure 5. Response curves and error curves.

Table 2. Values of fitness function and control indicators.

	Y	$\sigma(\%)$	t_r	$e_{ss}(\%)$	$ITAE_d$
Without disturbance	61.48	4.66	0.05	0	14.68
With disturbance	72.20	5.00	0.05	0.02	21.96

Figure 6 shows the observation results provided by the LESO with disturbance in the system. It can be seen from Figure 6 that the observation results of the position signal and speed signal are satisfactory. It can be seen from Figure 6 that the LESO cannot effectively observe the disturbance at the initial stage due to the large initial error when the system tracks the step signal, but after about 0.1 s, the LESO can effectively observe the disturbance introduced by the system.

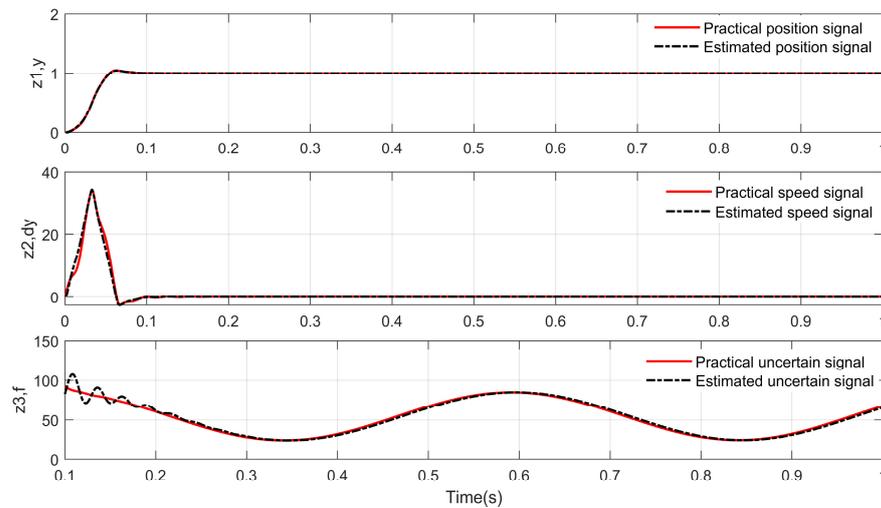


Figure 6. Observation curves provided by LESO.

4.2. Convergence Verification of I-SO–ADRC

As a control algorithm combined with an intelligent optimization algorithm, it is necessary to verify the convergence of I-SO and ADRC.

Figure 7 shows the convergence curve of the fitness function in the I-SO optimization process. It can be seen from Figure 7 that the value of the fitness function gradually decreases with the increase in the number of iterations and does not change after iteration 22, indicating that I-SO is convergent. Figure 8 depicts the optimization process of each parameter by I-SO in the form of a scatter diagram, which is complex. By observing the distribution of scatter points, it can be seen that the value of each parameter changes with the value of the fitness function, and finally converges to the optimal parameter value.

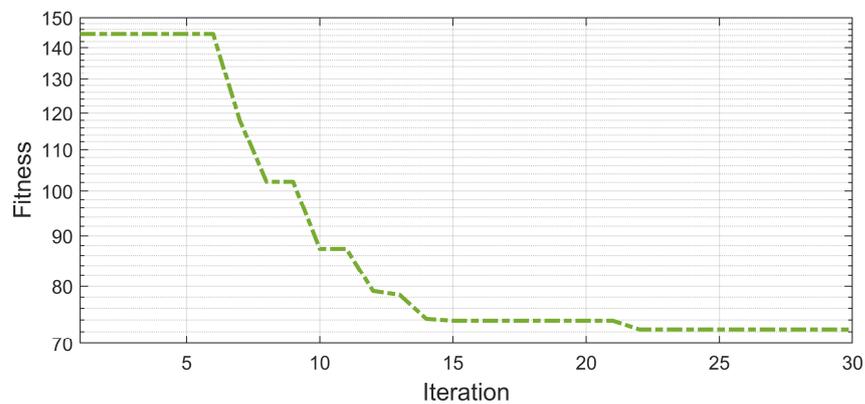


Figure 7. Convergence curve of the fitness function.

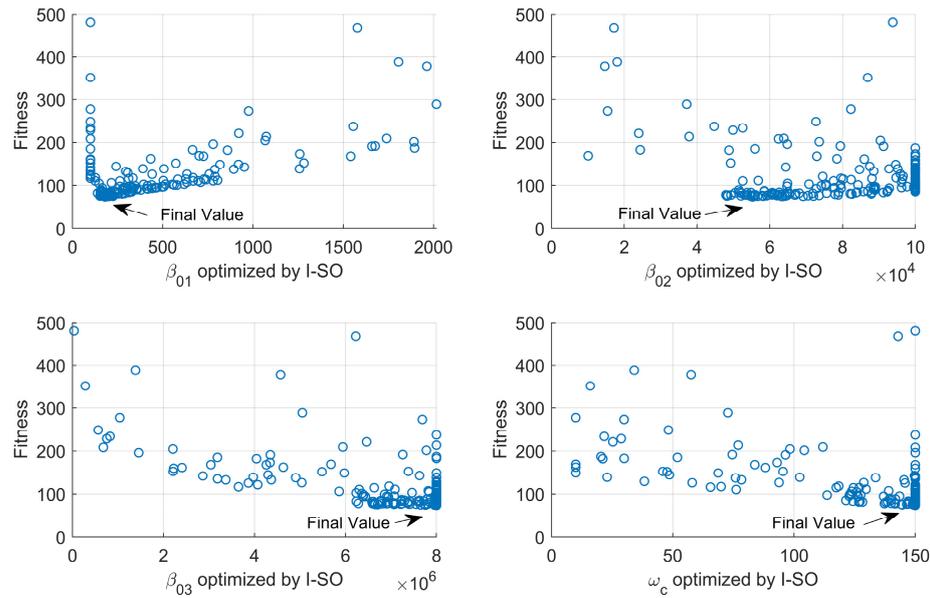


Figure 8. Parameter optimization process by I-SO.

In Figure 7, it can be seen that the fitness is greatly reduced at iterations 6, 9, and 22, which is representative of the iterative process. The parameter sets obtained for iterations 6, 9, and 22 are {950, 99,317, 7,442,311, 75}, {217, 50,420, 7,277,566, 80}, and {178, 49,642, 7,898,000, 145}, respectively. In order to verify the I-SO–ADRC convergence and the applicability of I-SO to tune the parameters of ADRC, the optimization results of iterations 6, 9, and 22 are compared in Figure 9. Table 3 shows the fitness values and the dynamic and stable control indicators of the three iterations.

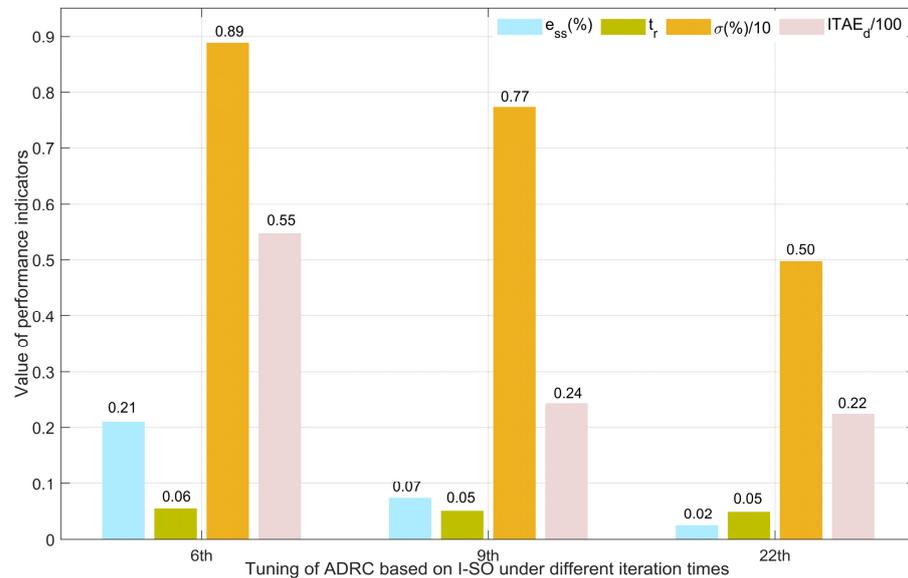


Figure 9. Indicator comparison histogram under different iteration times.

Table 3. Values of fitness function and control indicators of different iterations.

Iteration Time	γ	$\sigma(\%)$	t_r	$e_{ss}(\%)$	$ITAE_d$
6th	144.47	8.87	0.06	0.21	54.73
9th	102.14	7.74	0.05	0.07	24.33
22nd	72.20	5.00	0.05	0.02	21.96

Based on Table 3 and Figure 9, it can be seen that as the number of iterations increases, the performance of the system improves steadily, while the values of the indicators and fitness function generally decline, especially overshoot and $ITAE_d$, with significant reductions. The parameter sets obtained for iterations 6, 9, and 22 are applied to the control system, respectively, and the system response and error curves are obtained, as shown in Figure 10. It can be seen in Figure 10 combined with Table 3 that the performance of the control system using the parameter set obtained in iterations 6 and 9 is not ideal. Compared with iterations 6 and 9, the performance of the control system is much better when using the parameter set obtained in iteration 22, which is also the final optimal parameter set.

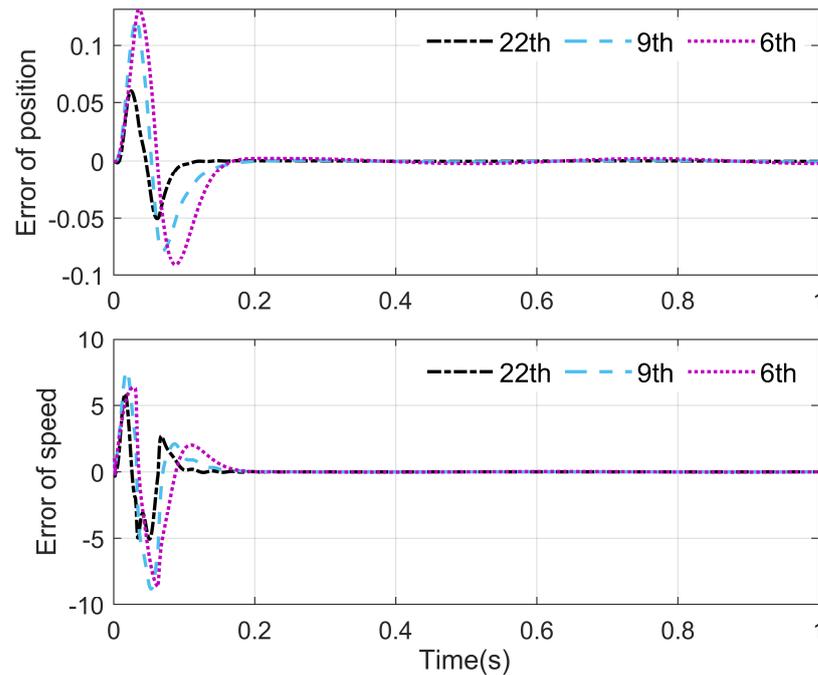


Figure 10. Response curves and their respective errors at iterations 6, 9 and 22.

4.3. Comparison of I-SO–ADRC with SO–ADRC

The simulation results presented in Figures 5 and 6 demonstrate that I-SO is feasible and effective for ADRC parameter tuning. In order to verify the advantages of the proposed I-SO in solving the parameter tuning problem of ADRC, under the same experimental environment and parameter conditions, the SO algorithm was used to tune the parameters of ADRC. The calculation of indicators adopted the method proposed in Section 4. Table 4 lists the indicators of the two SO algorithms. Figure 11 depicts a broken-line chart of the indicators of the two SO algorithms corresponding to Table 4. Figure 12 shows the convergence curves of the two SO algorithms. It can be seen from Table 4, Figures 11 and 12 that many of the indicators of I-SO are better than those of SO. Compared with SO, although I-SO has more convergence iterations, its running time is shorter and its fitness value is smaller, which shows that I-SO has stronger optimization ability and is less likely to fall into local optimality. I-SO has a much smaller STD than SO, indicating that the stability of I-SO is excellent.

Table 4. Indicators of different SO algorithms.

	Average	Iteration	STD	Time(s)
SO	87	22	12.1	6.34
I-SO	76	26	7.2	5.28

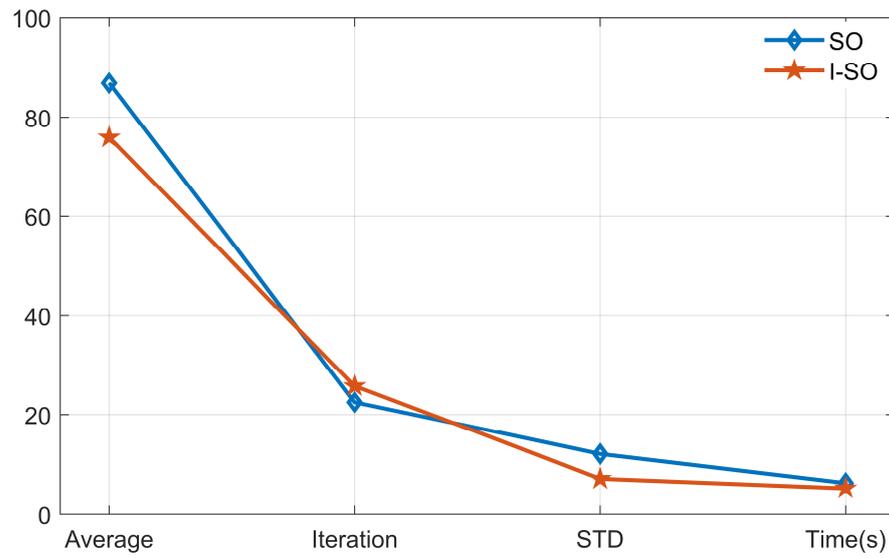


Figure 11. Indicator comparison of different SO algorithms.

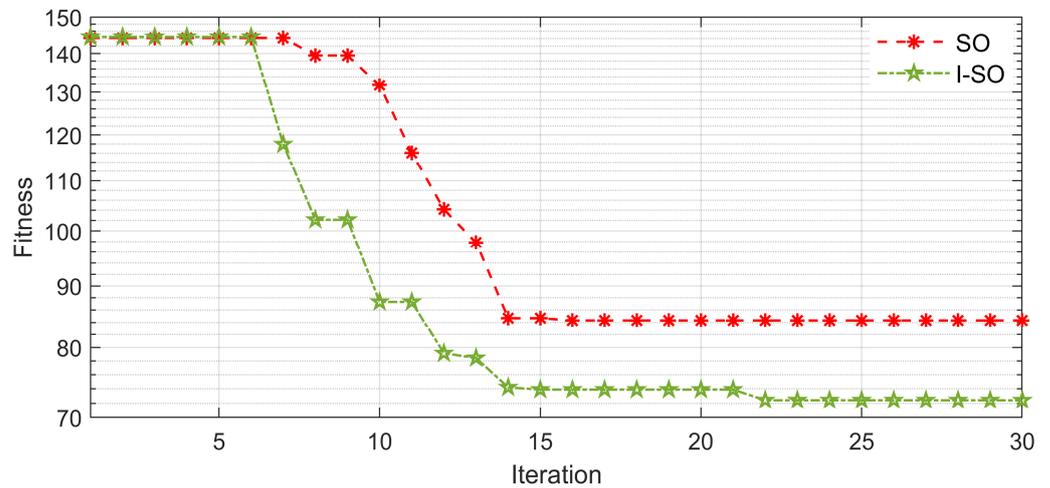


Figure 12. Convergence curves of different SO algorithms.

Figure 13 presents a column comparison chart of the four indicators that constitute the fitness function. Figure 14 shows the error curves obtained by applying the optimal parameter set corresponding to the two algorithms to the control system. According to Figures 13 and 14, two different SO algorithms can be applied to the parameter tuning of ADRC and provide ideal control, but the performance of the system when applying the I-SO is better.

Considering indicators of the control system and optimization algorithm comprehensively, compared with the SO algorithm, I-SO has the advantages of strong search ability, short running time, and good stability.

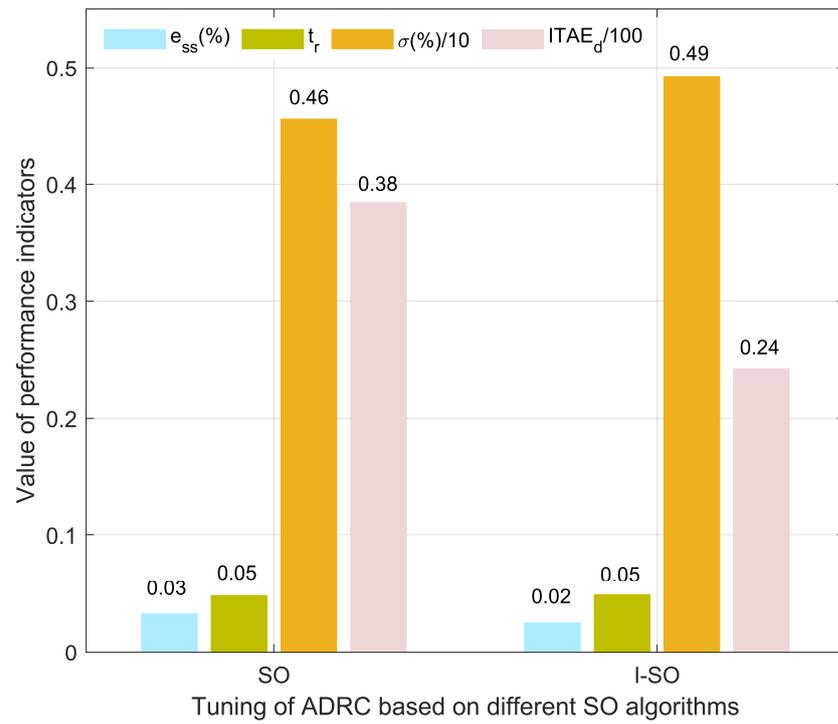


Figure 13. Indicator comparison histogram of ADRC based on different SO algorithms.

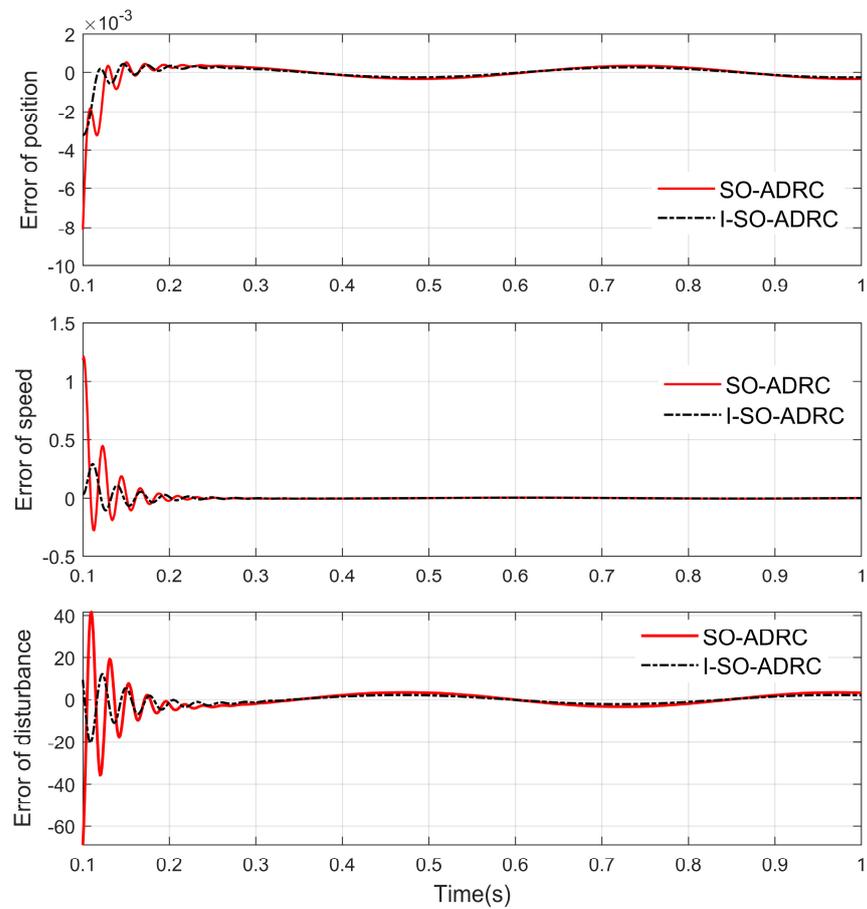


Figure 14. Error curves of ADRC based on different SO algorithms.

4.4. Comparison between I-SO–ADRC and PSO–ADRC, GA–ADRC and SSA–ADRC

To verify the effectiveness and superiority of I-SO compared to other intelligent optimization algorithms, three well-known optimization algorithms—PSO, GA, and SSA—were applied to the parameters tuning of ADRC, and the results were compared with those of I-SO.

In this experiment, if the final fitness was less than 100, the final solution was considered the global optimal solution. Otherwise, the final solution was considered the local optimal solution. Finally, the unique parameter settings of the three optimization algorithms were determined in this paper, as shown in Table 5. Other parameters like population size, upper limit, and lower limit were the same as those of I-SO. The initial number of iterations was set to 30.

Table 5. Parameter settings of the three optimization algorithms.

	Parameter	Parameter Description	Value
PSO	w	Inertia weight	[0.1,0.9]
	C_1	Self-best factor	1.5
	C_2	Global-best factor	1.5
GA	p_c	Crossover probability	0.9
	p_m	Mutation probability	0.1
SSA	ST	Security threshold	0.8
	p_p	Proportion of producers	0.2
	p_a	Proportion of sparrows who perceive the danger	0.5

Table 6 is a comparison of the indicators of the three algorithms against I-SO. According to [21], for evaluation of the performance of improved evolutionary algorithms, statistical tests are required. In this paper, we chose to use the Wilcoxon rank-sum test and conduct it at the 5% significance level. Assuming that I-SO is the best algorithm among the four, expressed as N/A, the p-values of the three algorithms can be obtained, as shown in Table 6. The p-values of the three algorithms are less than 0.05, which shows that the superiority of the I-SO algorithm is statistically significant.

Table 6. Indicators of different optimization algorithms.

	GA	PSO	SSA	I-SO
Average	96	90	79	76
STD	11.7	16.5	9.0	7.2
Iteration	23	29	29	26
Time(s)	5.50	5.72	8.73	5.28
p-value for Wilcoxon’s rank-sum test	2.446×10^{-6}	0.0018	9.660×10^{-7}	N/A

Figure 15 provides a radar map comparing the four algorithms, where the four axes, respectively, correspond to the four indicators in Table 6. The quadrangles with different colors represent the different algorithms. As seen in Figure 15, the green quadrangle representing the I-SO algorithm is almost completely encircled by the other quadrangles. Figure 16 shows the convergence curves of the four algorithms. According to Table 6, Figures 15 and 16, among the four algorithms, the average, STD, and time of the I-SO algorithm are the smallest, and only the convergence speed is slightly higher than that of GA. Compared with the other three algorithms, I-SO has stronger search ability, better stability, faster convergence, and lower complexity.

Figure 17 exhibits columnar comparison charts of the four indicators constituting the fitness function. Figure 18 shows the error curves obtained by applying the optimal parameter set corresponding to the four algorithms to the control system. According to Figures 17 and 18, compared with SSA and GA, the control system corresponding to I-SO shows a more ideal control effect and several control indicators achieve smaller values.

The control effect of the PSO-based system is also ideal. Compared with PSO, the I-SO algorithm is advantageous in its stability and disturbance observation ability. To sum up, the I-SO algorithm proposed in this paper can adjust ADRC parameters more effectively than the other three algorithms.

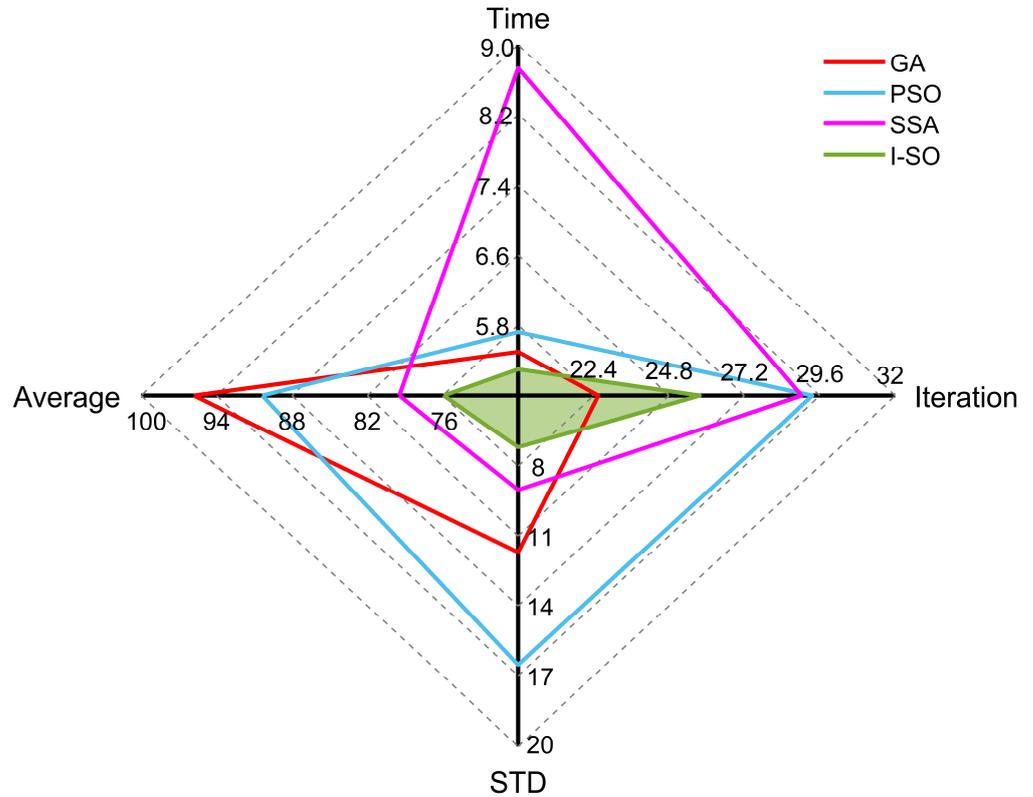


Figure 15. Radar map comparing indicators of the four algorithms.

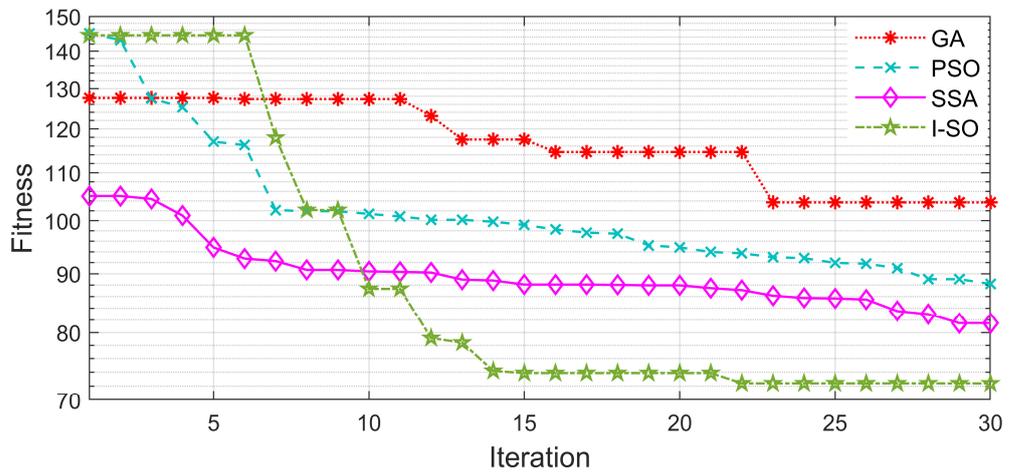


Figure 16. Convergence curves of the four algorithms.

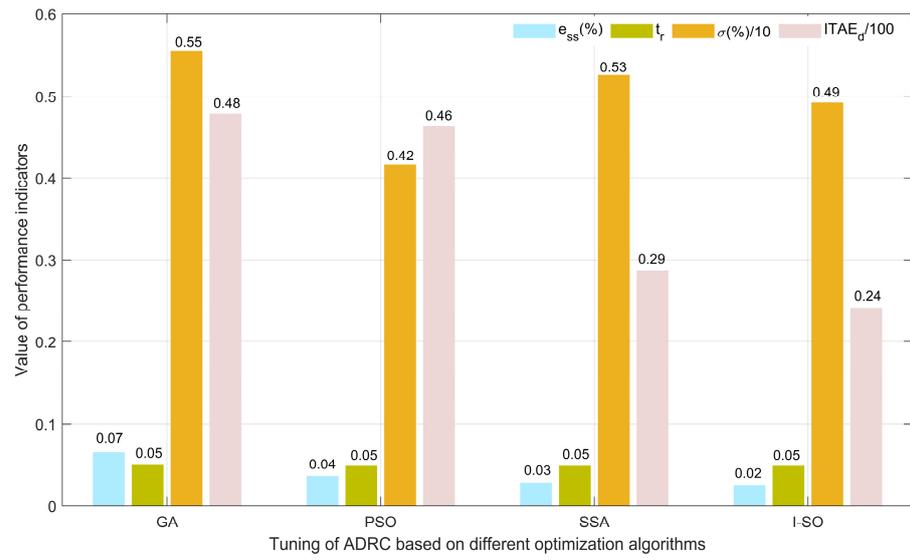


Figure 17. Indicator comparison histogram of ADRC based on different optimization algorithms.

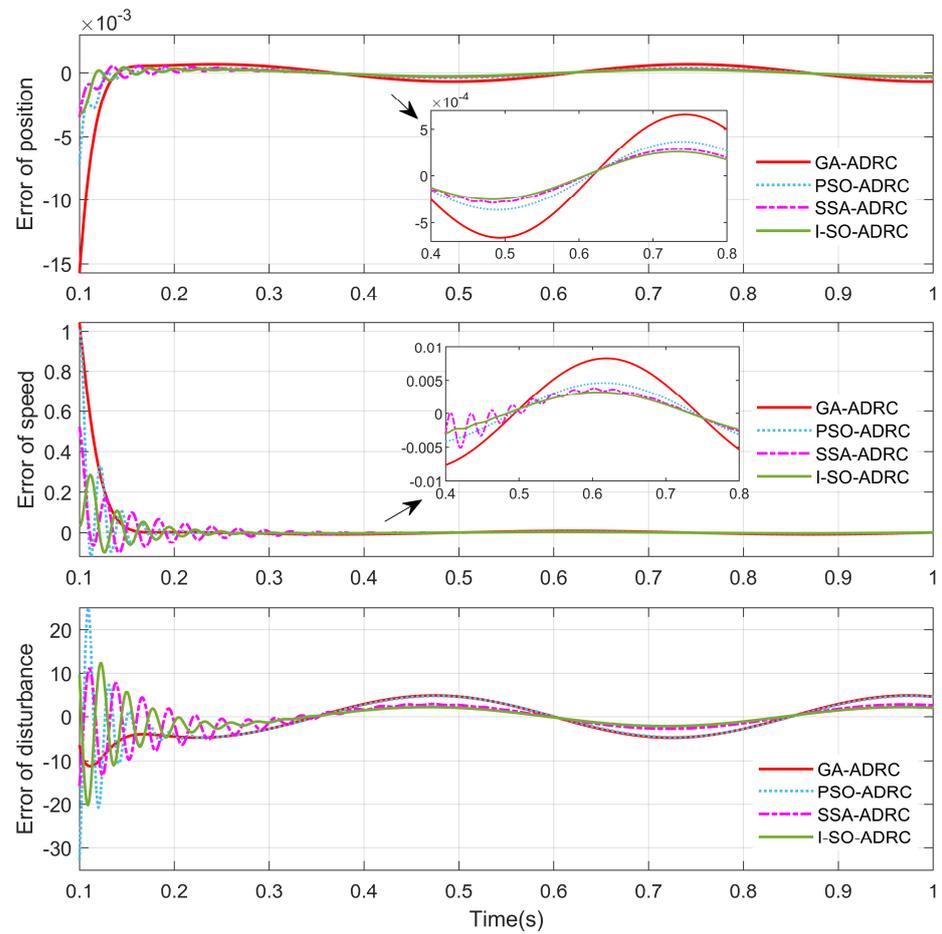


Figure 18. Error curves of ADRC based on different optimization algorithms.

5. Moving Mirror Control System

For the swing-arm interferometer in Figure 1, after derivation and a reference review, this paper obtained the equation of the swing angular displacement θ of the RVCМ and the optical path scanning speed v_0 of the Fourier transform, as shown in Equation (28):

$$\theta(t) = \sqrt{2}tg\left(\frac{\sqrt{2}v_0}{8r}t\right) \tag{28}$$

where r is the length of the swing arm. According to Equation (28) and the interferometer parameter, $v_0 = 0.64\text{cm/s}$, $r = 4.5\text{cm}$, the motion curve of RVCМ can be designed as shown in Figure 19.

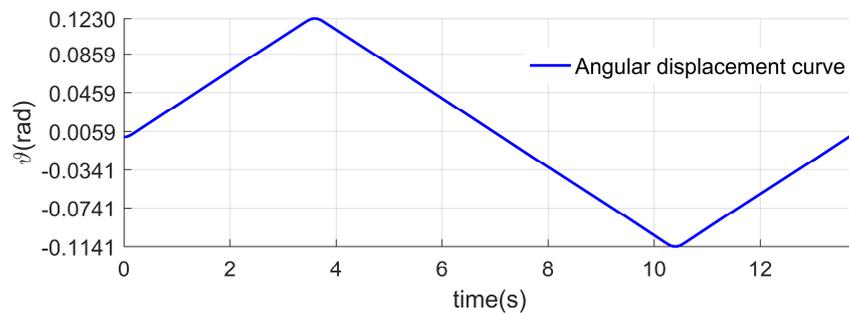


Figure 19. Planned motion curve of the swing arm.

Saber and MAST languages were used to build the RVCМ, H-bridge, and DSP models, a complete moving mirror control system was built, and joint simulation with Simulink 2022b was conducted to verify the effectiveness of ADRC with parameters tuned through I-SO. The simulation system is shown in Figure 20. The controller uses the optimal solution obtained by the I-SO. Considering that an H-bridge is used in the system, the parameters of SEF may need to be modified. After debugging, the value of ω_c is modified to 90. The parameters of LESO are not modified.

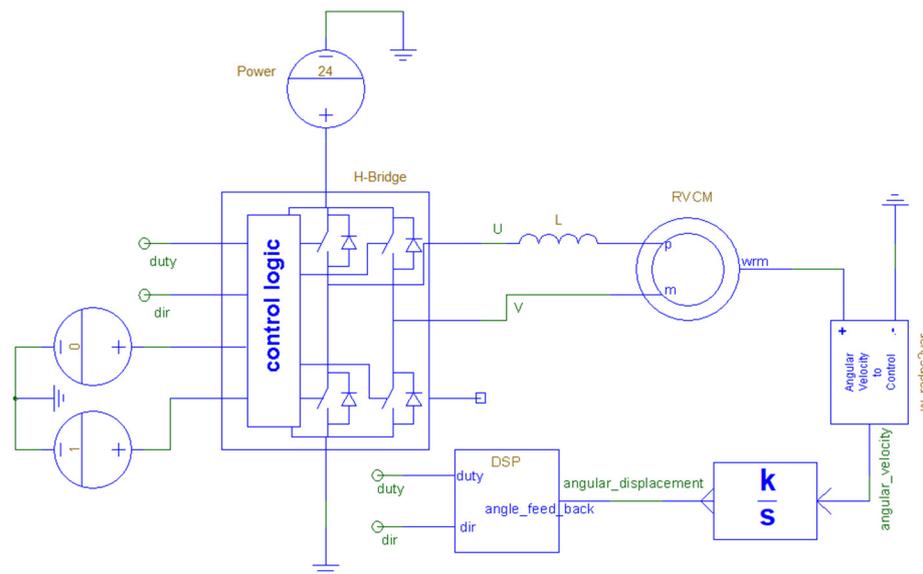


Figure 20. Overall block diagram of moving mirror control system.

The simulation results are shown in Figure 21 and Table 7. Table 7 shows the following error of the system. According to Figure 21a and Table 7, it can be seen that the maximum following error of angular displacement is 0.17 mrad, and large following errors appear during the starting and commutation stages of the RVCМ.

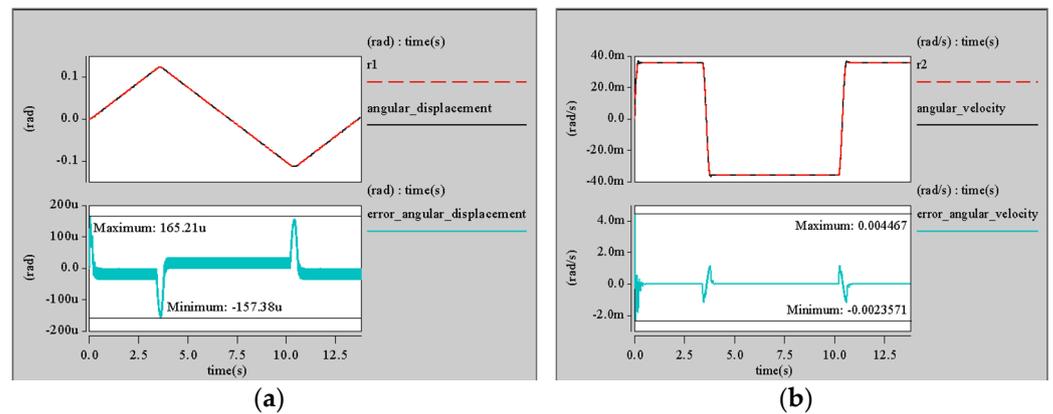


Figure 21. Following curves and following error curves: (a) angular displacement; (b) angular velocity.

Table 7. Following error.

	Maximum Error	Average Error	STD of Error
Angular displacement (mrad)	0.17	0.01	0.03
Angular velocity (mrad/s)	4.47	0.09	0.38

In Figure 21a, we can see that the curve experiences jittering, but the value of the jitter is extremely small, close to 0. There may be two reasons for the jitter shown in Figure 21a: first, the simulation step size of the Saber 2019 in this paper may be set too low, which affects the simulation accuracy; second, the current loop in the simulation system may cause the simulation results to be unstable, causing the curve to jitter.

The following curve and error curve of the angular velocity are shown in Figure 21b. It can be seen that large angular velocity following errors and fluctuations occur during the starting stage of the RVCM. However, during the steady-operation stage of the RVCM, the angular velocity following error is close to 0, achieving an excellent following effect. The stability of the optical path scanning speed of this system is calculated to be 99.2% in 4–10 s. The simulation results prove that the ADRC system tuned using I-SO has excellent control for the moving mirror control system and has a good following effect on the planned motion trajectory. Using I-SO to adjust the parameters of ADRC is efficient and effective, which provides certain reference significance for the use of the ADRC algorithm in moving mirror control systems.

6. Conclusions

A novel ADRC system with parameter autotuning based on I-SO was applied to a moving mirror control system. The key parameters of the ADRC system were continuously tuned through the optimization mechanism of I-SO. The simulation results show that the parameter set obtained by I-SO can achieve the best control compared to the original SO, PSO, GA, and SSA. A moving mirror control system was built. The results show that ADRC with parameter autotuning based on I-SO can successfully complete the motion requirements of the moving mirror. The STD of the error of the angular displacement and angular velocity was 0.03 and 0.38, respectively. The following effect of the angular displacement and the angular velocity during the smooth operation stage was excellent.

The chaotic elite opposition learning algorithm can improve the quality and maintain the diversity of initial solutions, which is conducive to improving the global search ability and stability of the SO. The introduction of the SC search mode can improve the SO's ability to jump out of local optima. The I-SO proposed in this paper effectively improves the applicability of ADRC and greatly shortens the time required for its parameter tuning. Compared with the original SO, PSO, GA, and SSA, I-SO has the best search capability and stability. ADRC with parameter autotuning based on I-SO shows ideal control.

Author Contributions: Conceptualization, M.H., W.H. and L.Z.; methodology, L.Z.; writing—original draft preparation, L.Z.; writing—review and editing, W.H. and L.Q.; visualization, Z.W. and Q.W.; project administration, M.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China (grant 62101533), in part by the Key Deployment of Chinese Academy of Sciences under grant E3Z204010F, and in part by the Key Deployment of Chinese Academy of Sciences under grant E1Z204020F.

Data Availability Statement: All data used to support the findings of this study are available from the corresponding author upon request.

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

References

1. Zhang, M.; Zhang, J.; Yang, H. Research on speed stability of moving mirror in FTIR spectrometer. *Infrared Laser Eng.* **2014**, *43*, 1240–1246.
2. Liang, X.-w.; Shi, L. Design of a Moving Mirror Scanning System for Portable Interferometer. *Spectrosc. Spectr. Anal.* **2017**, *37*, 3255–3259.
3. Shi, Y.; Zhou, C.; Yin, Q.; Zhou, Y. Simulation on Interference Control System in Fourier Transform Spectrograph. *J. Syst. Simul.* **2013**, *25*, 753–759. [[CrossRef](#)]
4. Hu, J.; Wu, J.; Chen, J.; Li, Y.; Ren, L.; Wang, Q.; Liu, C. Calculation, simulation and fast estimation of thermal resistance of rotary voice coil actuators. *IET Electr. Power Appl.* **2022**, *16*, 675–684. [[CrossRef](#)]
5. Li, Y.; Li, Y.; Ren, L.; Lin, Z.; Wang, Q.; Xu, Y.; Zou, J. Analysis and Restraining of Eddy Current Damping Effects in Rotary Voice Coil Actuators. *IEEE Trans. Energy Convers.* **2017**, *32*, 309–317. [[CrossRef](#)]
6. Roemer, D.B.; Bech, M.M.; Johansen, P.; Pedersen, H.C. Optimum Design of a Moving Coil Actuator for Fast-Switching Valves in Digital Hydraulic Pumps and Motors. *IEEE/ASME Trans. Mechatron.* **2015**, *20*, 2761–2770. [[CrossRef](#)]
7. Smith, K.J.; Graham, D.J.; Neasham, J.A. Design and Optimization of a Voice Coil Motor with a Rotary Actuator for an Ultrasound Scanner. *IEEE Trans. Ind. Electron.* **2015**, *62*, 7073–7078. [[CrossRef](#)]
8. Gokhan, C.; Reza, Z.; Ozan, K. Design and Optimization of Reduced Torque Ripple Rotary Voice Coil Motor. In Proceedings of the 2018 XIII International Conference on Electrical Machines (ICEM), Alexandroupoli, Greece, 3–6 September 2018; pp. 663–669.
9. Hsieh, C.-L.; Liu, C.-S.; Cheng, C.-C. Design of a 5 degree of freedom–voice coil motor actuator for smartphone camera modules. *Sens. Actuators A Phys.* **2020**, *309*, 112014. [[CrossRef](#)]
10. Okyay, A.; Erkorkmaz, K.; Khamesee, M.B. Mechatronic design, actuator optimization, and control of a long stroke linear nano-positioner. *Precis. Eng.* **2018**, *52*, 308–322. [[CrossRef](#)]
11. Al-Jodah, A.; Shirinzadeh, B.; Ghafarian, M.; Kumar Das, T.; Tian, Y.; Zhang, D. A fuzzy disturbance observer based control approach for a novel 1-DOF micropositioning mechanism. *Mechatronics* **2020**, *65*, 102317. [[CrossRef](#)]
12. Shewale, M.; Razban, A.; Deshmukh, S.; Mulik, S. Design, Development and Implementation of the Position Estimator Algorithm for Harmonic Motion on the XY Flexural Mechanism for High Precision Positioning. *Sensors* **2020**, *20*, 662. [[CrossRef](#)] [[PubMed](#)]
13. Wang, L.; Wang, Z.; Wang, F.; Shi, G.; Xu, R. Robust finite-time adaptive control for high performance voice coil motor-actuated fast steering mirror. *Rev. Sci. Instrum.* **2022**, *93*, 125003. [[CrossRef](#)] [[PubMed](#)]
14. Hsu, C.; Chen, B.; Wu, B. Fuzzy broad learning adaptive control for voice coil motor drivers. *Int. J. Fuzzy Syst.* **2022**, *24*, 1696–1707. [[CrossRef](#)]
15. Barzegar-Kalashani, M.; Tousi, B.; Mahmud, M.A.; Farhadi-Kangarlu, M. Robust nonlinear sliding mode controllers for single-phase inverter interfaced distributed energy resources based on super twisting algorithms. *ISA Trans.* **2022**, *123*, 61–75. [[CrossRef](#)] [[PubMed](#)]
16. Gong, Z.; Huo, D.; Niu, Z.; Chen, W.; Cheng, K. A novel long-stroke fast tool servo system with counterbalance and its application to the ultra-precision machining of microstructured surfaces. *Mech. Syst. Signal Process.* **2022**, *173*. [[CrossRef](#)]
17. Chen, S.Y.; Wu, G.W.; Yu, C.J. Robust fractional-order integral terminal sliding mode control with adaptive uncertainty observation for a VCM-driven X–Y motion stage. *IET Electr. Power Appl.* **2022**, *17*, 161–180. [[CrossRef](#)]
18. Tian, C.; Yan, P.; Zhang, Z. Inter-sample output predictor based sampled-data ADRC supporting high precision control of VCM servo systems. *Control Eng. Pract.* **2019**, *85*, 138–148. [[CrossRef](#)]
19. Han, J. Auto Disturbances Rejection Control Technique. *Front. Sci.* **2007**, *1*, 24–31.
20. Huang, Y.; Wu, S.; Jin, H.; Jiao, Z. Research on Voice Coil Motor Control Considering Interference. In Proceedings of the 16th IEEE Conference on Industrial Electronics and Applications (ICIEA), Chengdu, China, 1–4 August 2021; pp. 906–911.
21. Bernat, J.; Kolota, J. Active Disturbance Rejection Control for Dielectric Electroactive Polymer Actuator. *IEEE Access* **2021**, *9*, 95218–95227. [[CrossRef](#)]
22. Chen, Q.; Li, L.; Wang, M.; Pei, L. The precise modeling and active disturbance rejection control of voice coil motor in high precision motion control system. *Appl. Math. Model.* **2015**, *39*, 5936–5948. [[CrossRef](#)]
23. Zhou, X.; Gao, H.; Zhao, B.; Zhao, L. A GA-based parameters tuning method for an ADRC controller of ISP for aerial remote sensing applications. *ISA Trans.* **2018**, *81*, 318–328. [[CrossRef](#)] [[PubMed](#)]

24. Du, C.; Yin, Z.; Zhang, Y.; Liu, J.; Sun, X.; Zhong, Y. Research on Active Disturbance Rejection Control with Parameter Autotune Mechanism for Induction Motors Based on Adaptive Particle Swarm Optimization Algorithm with Dynamic Inertia Weight. *IEEE Trans. Power Electron.* **2019**, *34*, 2841–2855. [[CrossRef](#)]
25. Abdul-Kareem, A.; Hasan, A.; Al-Qassar, A.; Humaidi, A.; Hassan, R.; Ibraheem, I.; Azar, A. Rejection of wing-rock motion in delta wing aircrafts based on optimal LADRC schemes with butterfly optimization algorithm. *J. Eng. Sci. Technol.* **2022**, *17*, 2476–2495.
26. Ahmad, A.; Kashif, S.A.R.; Nasir, A.; Rasool, A.; Liaquat, S.; Padmanaban, S.; Mihet-Popa, L. Controller Parameters Optimization for Multi-Terminal DC Power System Using Ant Colony Optimization. *IEEE Access* **2021**, *9*, 59910–59919. [[CrossRef](#)]
27. Ren, J.; Chen, Z.; Yang, Y.; Sun, M.; Sun, Q.; Wang, Z. Grey Wolf Optimization Based Active Disturbance Rejection Control Parameter Tuning for Ship Course. *Int. J. Control. Autom. Syst.* **2022**, *20*, 842–856. [[CrossRef](#)]
28. Amlashi, A.G.; Rezvani, M.; Radmehr, M.; Ghafouri, A.; Razmjoo, N. Optimizing Adaptive Disturbance Rejection Control Models Using the Chimp Optimization Algorithm for Ships' Hybrid Renewable Energy Systems. *Comput. Intell. Neurosci.* **2022**, *2022*, 3569261. [[CrossRef](#)] [[PubMed](#)]
29. Ali, S.; Xia, Y.; Khan, Z.; Ali, A.; Navid, Q.; Aurangzeb, K.; Anwar, M. Frequency regulation in interconnected power system through Enhanced beluga whale optimized flatness-based active disturbance rejection control. *IEEE Access* **2024**, *12*, 15348–15367. [[CrossRef](#)]
30. Zhu, Q.; Zhuang, M.; Liu, H.; Zhu, Y. Optimal Control of Chilled Water System Based on Improved Sparrow Search Algorithm. *Buildings* **2022**, *12*, 269. [[CrossRef](#)]
31. Gao, B.; Shen, W.; Guan, H.; Zheng, L.; Zhang, W. Research on Multistrategy Improved Evolutionary Sparrow Search Algorithm and its Application. *IEEE Access* **2022**, *10*, 62520–62534. [[CrossRef](#)]
32. Pak, Y.; Kong, Y.; Ri, J. Robust pid optimal tuning of a delta parallel robot based on a hybrid optimization algorithm of particle swarm optimization and differential evolution. *Robotica* **2023**, *41*, 1159–1178. [[CrossRef](#)]
33. Kang, C.; Wang, S.; Ren, W.; Lu, Y.; Wang, B. Optimization Design and Application of Active Disturbance Rejection Controller Based on Intelligent Algorithm. *IEEE Access* **2019**, *7*, 59862–59870. [[CrossRef](#)]
34. Hashim, F.A.; Hussien, A.G. Snake Optimizer: A novel meta-heuristic optimization algorithm. *Knowl.-Based Syst.* **2022**, *242*, 108320. [[CrossRef](#)]
35. Shi, Y. Research on the Control Technology for the Interferometer Subsystem in Space-Borne Infrared Spectrometer. Ph.D. Thesis, Huazhong University of Science and Technology, Wuhan, China, 2012.
36. Liu, X.; Sun, X.; Hao, Z.; Liu, Y. A New Discrete-time Form of Optimal Tracking Differentiator. *Inf. Control* **2013**, *42*, 729–734.
37. Han, W.; Tan, W. Tuning of linear active disturbance rejection controllers based on PID tuning rules. *Control Decis.* **2021**, *36*, 1592–1600.
38. Khalil, N.; Sarhan, A.; Alshewimy, M.A.M. An efficient color/grayscale image encryption scheme based on hybrid chaotic maps. *Opt. Laser Technol.* **2021**, *143*, 107326. [[CrossRef](#)]
39. Yildiz, B.S.; Pholdee, N.; Bureerat, S.; Yildiz, A.R.; Sait, S.M. Enhanced grasshopper optimization algorithm using elite opposition-based learning for solving real-world engineering problems. *Eng. Comput.* **2021**, *38*, 4207–4219. [[CrossRef](#)]
40. Mirjalili, S. SCA: A Sine Cosine Algorithm for solving optimization problems. *Knowl.-Based Syst.* **2016**, *96*, 120–133. [[CrossRef](#)]

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.