

Article

Enhancing Active Disturbance Rejection Control for a Vehicle Active Stabiliser Bar with an Improved Chicken Flock Optimisation Algorithm

Zhenglin Tang ¹, Qiang Zhao ^{1,*}, Duc Truong Pham ²  and Xuesong Zhang ¹¹ College of Mechanical and Electrical Engineering, Northeast Forestry University, Harbin 150040, China² Department of Mechanical Engineering, University of Birmingham, Birmingham B15 2TT, UK

* Correspondence: zhaoqiang@nefu.edu.cn

Abstract: An active stabiliser bar significantly enhances the anti-roll capabilities of vehicles. The control strategy is a crucial factor in enabling the active stabiliser bar to function effectively. This paper investigates an active disturbance rejection control (ADRC) strategy. Given the numerous parameters of the ADRC and their significant mutual influence, optimising these parameters is challenging. To address this, an improved chicken flock optimisation algorithm is proposed to optimise the ADRC parameters and enhance its performance. First, a three-degree-of-freedom dynamic model of the vehicle is established, and an active disturbance rejection control-based optimisation model utilising a chicken flock optimisation algorithm is constructed. To tackle the issues of getting stuck in local optima and low precision when dealing with complex problems in the traditional chicken flock optimisation (CFO) algorithm, several strategies, including improved Lévy flight, have been adopted. Subsequently, the twelve parameters of the ADRC are optimised using the improved chicken flock optimisation algorithm. Comprehensive testing on multiple benchmark functions demonstrates that the improved chicken flock optimisation (ICFO) algorithm is distinctly superior to other advanced algorithms in terms of solution quality and robustness. Simulation results show that the ICFO-ADRC controller is significantly superior. In four different complex road condition tests, the ICFO-ADRC controller shows an average performance improvement of 8% compared to the fuzzy PI-PD controller, an average improvement of 82% compared to the non-optimised ADRC controller, and an average improvement of 18% compared to the CFO-ADRC controller. Our findings confirm that this paper was able to provide new solutions for vehicle stability control whilst opening up new possibilities for the application of metaheuristic algorithms.

Keywords: vehicle roll stability; active stabiliser bar; active disturbance rejection control; improved chicken flock optimisation algorithm



Citation: Tang, Z.; Zhao, Q.; Pham, D.T.; Zhang, X. Enhancing Active Disturbance Rejection Control for a Vehicle Active Stabiliser Bar with an Improved Chicken Flock Optimisation Algorithm. *Processes* **2024**, *12*, 1979. <https://doi.org/10.3390/pr12091979>

Academic Editor: Olympia Roeva

Received: 23 July 2024

Revised: 5 September 2024

Accepted: 10 September 2024

Published: 13 September 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

The lateral stabiliser bar is an integral component of a vehicle's suspension system and is designed to mitigate excessive rolling of the vehicle body [1]. When a vehicle traverses uneven terrain or navigates a sharp turn, the suspension on either side experiences varying degrees of deformation due to the roll force acting on the vehicle body. Concurrently, the endpoints of the stabiliser bar exhibit a significant elevation difference relative to the ground, causing the stabiliser bar to undergo torsional deformation as a whole. Once deformed, the torsional rigidity of the stabiliser bar generates an anti-roll torque that suppresses the vehicle body's roll to a certain extent, thereby enhancing the vehicle's driving stability and ride comfort [2].

Traditional passive stabiliser bars can reduce the vehicle's roll angle to some extent, but their inability to adjust stiffness according to driving requirements limits their anti-roll capability [3]. The active stabiliser bar applies an active anti-roll torque to the stabiliser bar by adding a driving force generated by an actuating mechanism on the basis of the

passive stabiliser bar and uses a control algorithm to generate the required anti-roll torque in real time according to the anti-roll requirements of the system, thus greatly improving the anti-roll capability of the system [4]. Over the past two decades, the research and application of active rod design and control have been increasing and have attracted widespread attention [5]. Many control methods have been applied to the control of active stabiliser bars, such as fuzzy control [6–8], fuzzy PID control [9], H_∞ /LPV control [10,11], sliding mode control [12], LQR control [13,14], and active disturbance rejection control (ADRC) [15]. The input of a fuzzy controller for an active stabiliser bar generally includes the vehicle body roll angle, the unsprung mass displacement, the differences between the forces at the wheels, etc., and its control output is the input current or voltage into the active actuators [6]. The design of fuzzy rules requires the experience of operators and experts, which is difficult, especially for multiple inputs. Fuzzy PID control uses the fuzzy rule to regulate the coefficients of proportional, integral, and derivative terms, and its fuzzy rule can be designed according to the principle of PID control, thus allowing easier realisation [9]. The H_∞ /LPV controller in [10] used the vehicle forward velocity and the normalised load transfers as the dependent varying parameters, and used the H_∞ /LPV controller with weighting functions to realise on-line performance adaptation to rollover risk. Although the corresponding design theory and process are relatively complex, it is a further developed version of optimal control that considers robustness and parameter time-varying characteristics. Reference [12] used sliding mode control for the upper-level controller to calculate the desired active anti-roll torque, which fully utilises simplicity, robustness, and rapidness. However, sliding mode control has the weakness that it has some difficulties expressing the quadratic performance index, which is easily done by optimal control [14].

Among the various active stabiliser bar control algorithms, ADRC stands out for its unique advantages. It is a novel control algorithm proposed by Han based on traditional PID control that can reconcile the contradiction between response speed and overshooting in traditional PID control [15–17]. Compared with PID control, the advantages of ADRC are as follows [15–17]: (i) the transition process is arranged, and the differential is extracted in a reasonable manner, which reduces overshooting and improves system stability; (ii) a nonlinear feedback method is used to achieve a nonlinear combination approach, allowing for a rapid response to large errors and fine-tuning for small errors; and (iii) an extended state observer is designed to estimate and compensate for the total disturbance, thereby effectively suppressing it and enhancing the system's robustness against both internal uncertainties and external disturbances. The ADRC control strategy integrates estimation, compensation, and control into a unified framework, simplifying the overall control system design. Moreover, compared to other control algorithms, ADRC also has several notable advantages [15–17]. For instance, it offers (i) enhanced robustness in the presence of model uncertainties and external disturbances, which is a common challenge for many optimisation-based control algorithms, allowing ADRC to maintain stable performance even when the system model is imprecise or changing; and (ii) improved transient performance and stability, which are crucial for real-time control applications that require quick and stable responses, as ADRC can achieve faster settling times and reduced oscillations. These features make ADRC not only flexible but also highly reliable for complex control tasks. However, the practical application of ADRC is not without its challenges. The most critical issue is the multitude of parameters that need to be adjusted, which increases the complexity of ADRC design and debugging [18].

As previously mentioned, ADRC is an advanced and innovative variant of PID control, and traditional PID parameter tuning methods can provide some inspiration for ADRC parameter adjustment. Regarding the tuning of traditional PID controller parameters, whilst there are some classical methods, such as the Ziegler–Nichols method and the Cohen–Coon method [19], their effectiveness is limited, as these methods typically rely on linear system models and struggle to address the control requirements of nonlinear and complex systems. Hence, a significant trend in recent years has been the use of metaheuristic algorithms for parameter tuning [20]. Metaheuristic algorithms offer several advantages in

parameter optimisation: they can search for optimal solutions in large-scale search spaces without requiring precise mathematical models of the system; they can effectively handle nonlinear, multi-variable, and complex constraint optimisation problems; they possess good global search capabilities and local optimisation abilities; and they can find near-globally optimal solutions within reasonable computation times [21]. These characteristics enable metaheuristic algorithms to achieve efficient and precise tuning of PID controller parameters. For instance, Liu et al. proposed using a grey wolf optimiser (GWO) to optimise fuzzy PID controller parameters online for controlling a quarter-car semi-active suspension system [22]. Mohd considered five gain parameters of a fractional-order PID controller and proposed a variant of a marine predator algorithm to adjust these parameters [23].

Compared to traditional PID control, ADRC has more than ten interlinked parameters, making its parameter adjustment more challenging [24]. Influenced by PID parameter tuning methods, metaheuristic algorithms have also been widely applied to ADRC parameter optimisation in recent years, with their advantages equally applicable in the ADRC parameter tuning process. Ren et al. applied a GWO to optimise ADRC control parameters, achieving significant results in ship course control [24]. Kang et al. developed an improved hybrid algorithm based on a fish swarm algorithm and a particle swarm optimisation (PSO) algorithm, successfully applying it to ADRC parameter tuning [25]. Yu et al. [26] and Rivera et al. [27] separately utilised PSO algorithms to optimise ADRC parameters, both achieving good results. In terms of genetic algorithms (GA), Huang et al. [28] and Shui et al. [29] applied them to ADRC parameter optimisation, demonstrating the potential of GA in this field. Shen et al. [30] went further by combining a PSO with a GA for parameter tuning of attitude ADRC in quadrotor aircraft, showcasing the advantages of hybrid algorithms in complex control systems. In addition to these methods, the artificial bee colony algorithm [31] and other nature-inspired algorithms [32] have also been successfully applied to ADRC parameter tuning, further expanding the application scope of metaheuristic algorithms in ADRC parameter optimisation. These research outcomes not only demonstrate the effectiveness of metaheuristic algorithms in ADRC parameter tuning, but also provide new insights for addressing the more complex parameter structure of ADRC, laying the foundation for achieving high-performance control systems.

Despite the impressive performance of numerous metaheuristic algorithms in the field of ADRC parameter tuning, the recently introduced chicken flock optimisation (CFO) algorithm has not yet been applied to the parameter optimisation of ADRC (note that the original algorithm was called the “Chicken Swarm Optimisation (CSO)” algorithm; however, as the correct English term for a group of chickens is a flock, the term “Chicken Flock Optimisation (CFO)” algorithm will be used in this paper) [33]. The CFO algorithm simulates the hierarchical structure and foraging behaviour of a chicken flock, including roles such as the rooster, hen, and chick, and offers the advantages of fast convergence speed and high efficiency in reaching solutions. It is capable of simply and swiftly resolving a variety of numerical computation problems within the scientific research domain [34]. According to the “no free lunch” theorem in the field of optimisation, there is no single algorithm that can outperform all others across every problem. This theorem highlights the balance between the universality and particularity of optimisation algorithms, which encourages researchers to continue developing and improving new algorithms while expanding their application range [35]. By integrating the unique characteristics of CFO, such as its hierarchical structure and foraging behaviour, into the optimisation of ADRC parameters, it may be possible to achieve a more robust and efficient control system. Such a system would be better adapted to the dynamic and complex conditions encountered in driving scenarios.

Based on the aforementioned analysis and leveraging the unique advantages of the CFO algorithm while also addressing its shortcomings, such as susceptibility to local optima and low precision, this paper introduces an improved version of the CFO algorithm. This enhanced algorithm incorporates an efficient strategy for population initialisation, improved Lévy flight, and random differential mutation among other search operators, significantly enhancing the algorithm’s search efficiency and global optimisation capabil-

ities. Furthermore, this paper applies the ICFO algorithm to optimise the parameters of ADRC. This approach effectively addresses the challenge of optimising numerous coupled parameters that traditional empirical tuning methods struggle with. In comparison with other advanced algorithms, the ICFO algorithm has not only demonstrated superior performance in solving complex optimisation problems, but has also shown its potential and prospects in the parameter optimisation of ADRC. It offers a novel optimisation tool for the field of engineering control.

In summary, the main contributions of this paper are as follows:

1. A novel ICFO algorithm is introduced. The proposed algorithm incorporates an improved initialisation strategy, adaptive search mechanism, dynamic weight factor, and differential mutation strategy, significantly enhancing the search precision of CFO and reducing the likelihood of it getting trapped in local optima.
2. For the first time, the ICFO algorithm has been applied to the parameter optimisation of the ADRC, thereby expanding the application scope of the CFO algorithm within the field of engineering control.
3. Under various road conditions, implementation and testing were conducted. By comparison with other control methods, the applicability and effectiveness of the optimised ADRC have been validated.

The structure of this paper is organised as follows: Section 1 establishes a three-degree-of-freedom vehicle model with a stabiliser bar and subsequently transforms it into a state-space equation for control design. Section 2 describes the design of the active disturbance rejection controller for the active stabiliser bar system. Section 3 analyses the CFO algorithm, improves it, and outlines the process of designing the ADRC based on the ICFO algorithm. Section 4 verifies the proposed ADRC through simulation examples. Finally, the conclusions are presented, the limitations are discussed, and directions for future research are explored.

2. Active Disturbance Rejection Control of the Vehicle Active Lateral Stabiliser

2.1. Three-DOF Vehicle Dynamics Model

To study the influence of the active lateral stabiliser bar on the roll stability of the vehicle, a vehicle system dynamics model including lateral, yaw, and roll degrees of freedom needs to be established. The position of the vehicle body's centre of mass is taken as the coordinate origin. A coordinate system is established according to the right-hand rule, with the positive direction of the x-axis set as the forward direction of the vehicle, the positive direction of the y-axis set as the left side of the forward direction, and the z-axis set as the vertical upwards direction through the centre of mass. A schematic model of a vehicle with an active lateral stabiliser is then established, as shown in Figure 1.

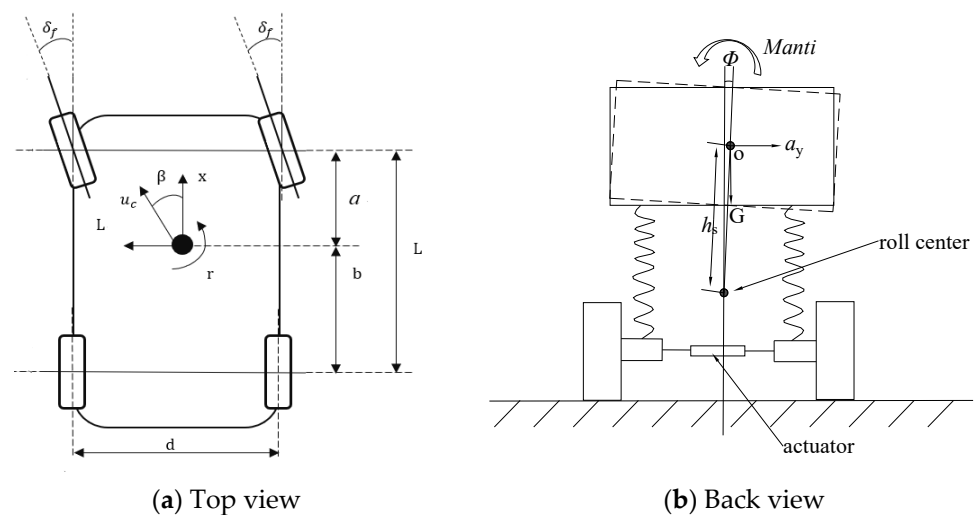


Figure 1. Diagram of a vehicle with an anti-roll stabiliser bar.

According to the Newton–Euler method, the dynamic equations of the vehicle with three degrees of freedom are established as follows:

- (1) Lateral movement of the vehicle:

$$M(\dot{u}_y + u_c r) + (am_f - bm_r)\dot{r} + m_b h_s \ddot{\phi} = F_{yf} + F_{yr} \quad (1)$$

where M is the total mass of the vehicle, u_y and u_c are the lateral speed and total speed of the vehicle respectively, r is the yaw rate of the body, $am_f - bm_r$ is the torque around the centre of mass of the vehicle, m_f and m_r are the masses of the front and rear axles, respectively, m_b is the sprung mass, a and b are the distances from the centre of mass of the vehicle to the front and rear axles, respectively, h_s is the difference between the sprung mass height and the roll centre height, ϕ is the roll angle of the body, and F_{yf} and F_{yr} are the lateral forces of the front tire and the rear tire, respectively, $F_{yf} = K_f(\beta + \frac{ar}{u_x} - \delta)$, and $F_{yr} = K_r(\beta - \frac{br}{u_c})$, where K_f and K_r are the lateral stiffness of the front and the rear tires, respectively, β is the side slip angle, u_x is the longitudinal speed of the vehicle, and δ is the steering wheel angle.

- (2) Yaw motion of the vehicle:

$$(am_f - bm_r)(\dot{u}_y + u_c r) + I_{zz}\dot{r} + I_{xz}\ddot{\phi} = aF_{yf} - bF_{yr} \quad (2)$$

where I_{zz} is the moment of inertia of the vehicle around the Z-axis and I_{xz} is the product of inertia of the vehicle around the XZ-axis.

- (3) Roll motion of the vehicle:

$$I_{xx}\ddot{\phi} + I_{xz}\dot{r} + D_\phi\dot{\phi} + (K_\phi - m_b g h_s)\phi + m_b h_s(\dot{u}_y + u_c r) = d_f F_{yf} + d_r F_{yr} \quad (3)$$

where I_{xx} is the moment of inertia of the complete vehicle around the X-axis, D_ϕ is the angular damping provided by the suspension when the vehicle body rolls, K_ϕ is the angular stiffness provided by the suspension when the vehicle body rolls, and d_f and d_r are the offsets of the front and rear tires caused by per unit value of vehicle body roll angle, respectively.

To represent the above motions of the vehicle using the state equation method, the state variables are considered:

$$x = \begin{bmatrix} u_y & r & \phi & \dot{\phi} \end{bmatrix}^T \quad (4)$$

Then, Equations (1)–(3) are arranged into the form of a state-space equation based on the aforementioned state variables to obtain the following state representation:

$$E_0 \dot{x} = A_0 x + B_0 \delta_f \quad (5)$$

where the specific elements in matrices E_0 , A_0 , and B_0 are as follows:

$$E_0 = \begin{bmatrix} M & am_f - bm_r & 0 & m_b h_s \\ am_f - bm_r & I_{zz} & 0 & I_{xz} \\ m_b h_s & I_{xz} & D_\phi & I_{xx} \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (6)$$

$$A_0 = \begin{bmatrix} \frac{K_f + K_r}{u_c} & \frac{aK_f - bK_r}{u_c} - Mu_c & 0 & 0 \\ \frac{aK_f - bK_r}{u_c} & \frac{a^2 K_f + b^2 K_r}{u_c} - (am_f - bm_r)u_c & 0 & 0 \\ \frac{d_f K_f + d_r K_r}{u_c} & \frac{ad_f K_f - bd_r K_r}{u_c} - m_b h_s u_c & m_b g h_s - K_\phi & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \quad (7)$$

$$B_0 = \begin{bmatrix} -K_f & -aK_f & -d_fK_f & 0 \end{bmatrix}^T \quad (8)$$

Then, Equation (5) is further transformed into the general form of the standard state equation, as represented in the following forms:

$$\dot{x} = Ax + B_f\delta_f \quad (9)$$

where A is the system matrix, B_f is the input matrix, and they are as follows:

$$A = E_0^{-1}A_0 \quad (10)$$

$$B_f = E_0^{-1}B_0 \quad (11)$$

Notably, the dynamic system described by the aforementioned state-space expression, in the case of a passive anti-roll bar, does not incorporate active anti-roll torque. After adding the active anti-roll bar to the vehicle dynamics model, the body roll motion equation accordingly includes the active anti-roll torque input, as described in the following:

$$I_{xx}\ddot{\phi} + I_{xz}\dot{r} + D_\phi\dot{\phi} + (K_\phi - m_bgh_s)\phi + m_bh_s(\dot{u}_y + u_cr) - M_{anti} = d_fF_{yf} + d_rF_{yr} \quad (12)$$

where M_{anti} denotes the active anti-roll torque input.

A schematic diagram of the active anti-roll torque generated by the active lateral stabiliser bar installed with a motor actuator or hydraulic motor actuator is shown in Figure 2. Under the active anti-roll torque provided by the actuator, the left and right parts of the lateral stabiliser bar balance the roll movement of the vehicle body through the action of torque, which reduces the roll angle of the vehicle body and improves the roll stability of the body.

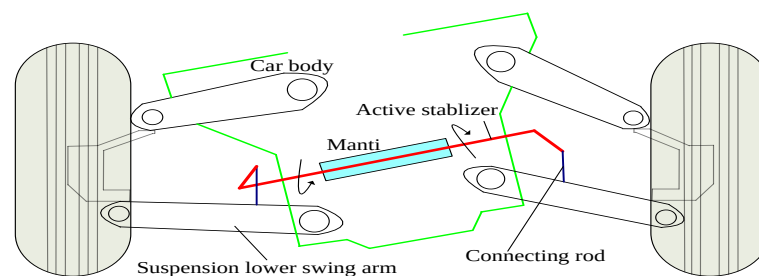


Figure 2. Diagram of the active lateral stabiliser bar.

After adding the active anti-roll torque term, the general form of the state equation in Equation (9) is further converted to:

$$\dot{x} = Ax + B_f\delta_f + B_uM_{anti} \quad (13)$$

where $B_u = [0 \ 0 \ -1 \ 0]^T$, and δ_f is the steering angle of the front wheel.

Equation (13) shows that the inputs of the vehicle body roll system are the front wheel steering angle and the anti-roll torque provided by the active lateral stabiliser bar. When $M_{anti} = 0$, it corresponds to the passive lateral stabiliser bar system, in which case the stabiliser bar system functions in a passive mode by providing only a torque caused by torsion deformation, as represented in Equation (9).

2.2. Active Disturbance Rejection Control of the Active Stabiliser Bar

According to Han's research [15], ADRC primarily consists of three components: the tracking differentiator (TD), the extended state observer (ESO), and the nonlinear state error feedback (NLSEF) control law. The TD is tasked with processing the input signal, while the ESO seeks to mitigate the impact of both internal and external disturbances on the target system. Moreover, NLSEF offers a tailored control strategy to navigate the

inherent trade-off between overshooting and response time encountered with traditional PID controls.

The implementation of ADRC within the active stabiliser bar is illustrated in Figure 3. In this figure, $v_0(t)$ denotes the input signal, which corresponds to the signal provided by the error feedback controller and signifies the target roll angle as discussed in this paper; $v_1(t)$, $v_2(t)$, $\dots$, and $v_n(t)$ indicate the differentiated signal, which is processed by the TD and is noted for its high signal-to-noise ratio; $z_1(t)$, $z_2(t)$, $\dots$, and $z_n(t)$ represent the state estimates derived from the observation and extension by the ESO; $y(t)$ is the system's output signal, which corresponds to the actual roll angle of the vehicle's body; and $d(t)$ denotes the external disturbance signals that the controlled system encounters.

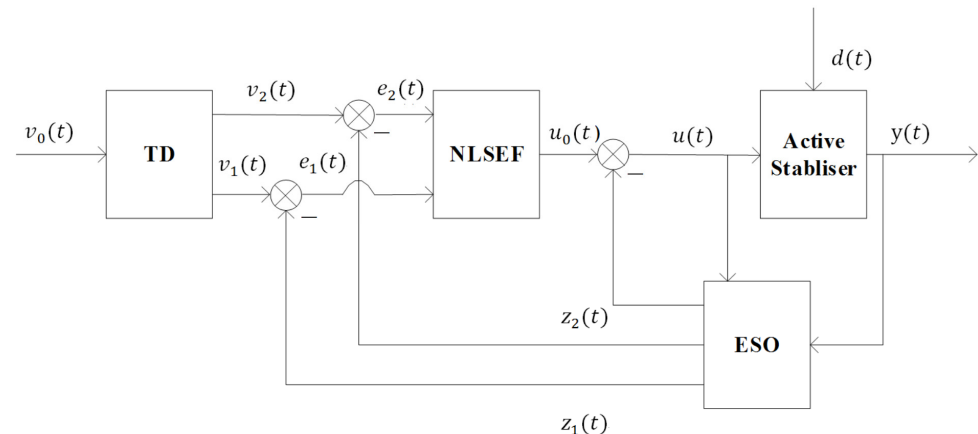


Figure 3. ADRC of the active stabiliser bar.

The function of the TD is to process the given input signal, not only to track the given reference signal but also to differentiate the subsequent signal. In addition, herein it is taken as:

$$\dot{v}_2 = -R \cdot \text{sat}(v_1 - v + |v_2|v_2/(2R), \delta_0) \quad (14)$$

where $\text{sat}(\cdot)$ is a conditional function as follows:

$$\text{sat}(x_1, x_2) = \begin{cases} \text{sign}(x_1) & \text{if } |x_1| > x_2 \\ x_1/x_2 & \text{else} \end{cases} \quad (15)$$

The ESO module is a pivotal component of ADRC. Its primary function is to estimate both the internal and external disturbances of the system once the state variables have been determined. Upon obtaining the total disturbance, the ESO module calculates the compensatory measure and conveys it to the NLSEF module. This compensatory measure, once processed by the NLSEF module, serves to neutralise the system's disturbances caused by internal and external interferences. In this paper, the equations selected for the ESO module are designed as:

$$\begin{cases} e = z_1 - y \\ \dot{z}_1 = z_2 - \beta_{01}\text{fal}(e, \alpha, \delta_1) \\ \dot{z}_2 = z_3 - \beta_{02}\text{fal}(e, \alpha, \delta_1) + b_0 u \\ \dot{z}_3 = -\beta_{03}\text{fal}(e, \alpha, \delta_1) \end{cases} \quad (16)$$

where $\text{fal}(\cdot)$ is determined as:

$$\text{fal}(x_1, x_2, x_3) = \begin{cases} |x_1|^{x_3} \text{sign}(x_1) & \text{if } |x_1| \geq x_3 \\ x_1/x_3^{1-x_2} & \text{else} \end{cases} \quad (17)$$

The primary function of NLSEF is to transform the entire system into a controllable and observable standard form of an integral chain by combining calculations of $e_1(t)$, $e_2(t)$, $\dots$, and $e_n(t)$ and its derivatives with the compensation derived from the ESO module for the system's internal and external disturbances. The design variables for the NLSEF are presented as follows:

$$\begin{cases} e_1 = v_1 - z_1 \\ e_2 = v_2 - z_2 \\ u_0 = \beta_1 \text{fal}(e_1, \alpha, \delta) + \beta_2 \text{fal}(e_2, \alpha, \delta) \\ u = u_0 - z_3 / b_0 \end{cases} \quad (18)$$

3. Controller Parameter Optimisation Using ICFO

3.1. Fundamental Concept of the CFO

The CFO algorithm is a novel metaheuristic swarm intelligence algorithm that simulates the foraging behaviour of chickens, structuring them into distinct groups. Each group comprises a rooster, several hens, and chicks [33,34]. Figure 4 provides a visual representation of this hierarchical structure. Within each group, the rooster, which possesses the strongest search capability, occupies a dominant position and exhibits the highest fitness value. The hens, with relatively weaker search abilities, closely follow the rooster in foraging, holding a fitness value that ranks just below the rooster's. Some hens also guide the chicks, who have the weakest search capabilities and confine their foraging to the vicinity of the hens, resulting in the lowest fitness value. This hierarchy enables the realisation of both global and local search functions.

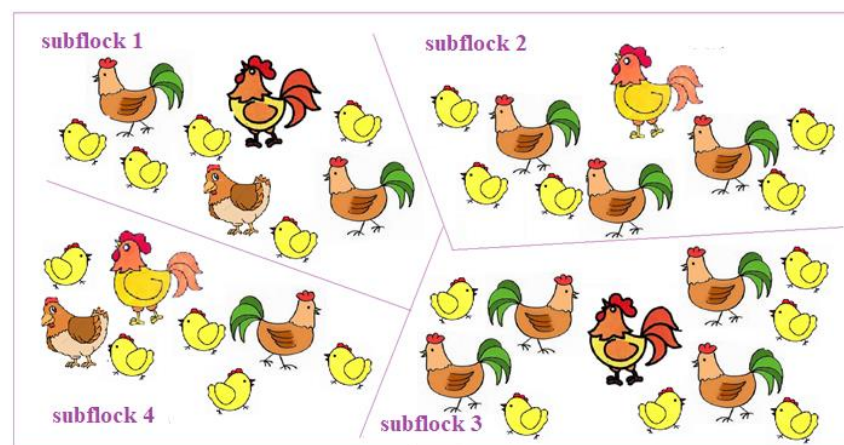


Figure 4. Diagram of the CFO algorithm.

Adhering to this stratified order, each member of the flock has an associated position and velocity update rule, detailed as follows [33]:

The position update rule of the rooster in each group is as follows:

$$\begin{cases} x_{ij}(t+1) = x_{ij}(t)(1 + \text{randn}(0, \sigma^2)) \\ \sigma^2 = \begin{cases} 1 & \text{if } f_i \leq f_k \\ \exp\left(\frac{f_k - f_i}{|f_i| + \epsilon}\right) & \text{else} \end{cases} \quad k = 1, 2, \dots, N_R \text{ \& } k \neq i \end{cases} \quad (19)$$

where $x_{ij}(t)$ represents the j th dimension value of rooster i when the iteration is t , $\text{randn}(0, \sigma^2)$ represents a random number subject to a Gaussian distribution with a mean of 0 and a variance of σ^2 , f_i represents the fitness value of rooster i , ϵ is a positive constant, with a small value, and N_R represents the number of roosters.

The foraging behaviour of hens within each group is predominantly influenced by the rooster. During foraging, these hens also engage in communication with the rooster

and with hens from other subgroups. The positional update formula for these hens is represented as follows:

$$\begin{cases} x_{ij}(t+1) = x_{ij}(t) + C_1 \cdot rand \cdot (x_{r1j}(t) - x_{ij}(t)) + C_2 \cdot rand \cdot (x_{r2j}(t) - x_{ij}(t)) \\ C_1 = \exp((f_i - f_{r1}) / (\text{abs}(f_i) + \varepsilon)) \\ C_2 = \exp(f_i - f_{r2}) \end{cases} \quad (20)$$

where *rand* is a random number satisfying a 0–1 uniform distribution, $x_{ij}(t)$ represents the *j*th dimension value of hen *i* when the iteration is *t*, x_{r1j} represents the *j*th dimension value of rooster r_1 of the subgroup of the hen when the iteration number is *t*, and $x_{r2j}(t)$ represents the *j*th dimension value of the rooster of the other subgroups or other hens r_2 when the iteration number is *t*. ε is a positive constant, with a small value.

The foraging behaviour of the chicks in each group is primarily influenced by the hens, with their positional update governed by the following formula:

$$x_{ij}(t+1) = x_{ij}(t) + F \cdot (x_{mj}(t) - x_{ij}(t)) \quad (21)$$

where *F* is a random number that obeys the uniform distribution of [0, 2] and $x_{mj}(t)$ is the *j*th dimension value of the individual hen *m* corresponding to chick *i* when the number of iterations is *t*.

The basic steps of the CFO algorithm are as follows:

Step 1: Establish the principal parameters, primarily flock size, iteration count, the frequency of population relationship updates, the dimension of individual positions, and the ratio of roosters, hens, and chicks within the flock.

Step 2: Based on fitness values, the first *RN* individuals are designated as roosters, the last *CN* individuals are designated as chicks, and the remainder are classified as hens. The flock is then divided into *RN* groups corresponding to the number of roosters. Hens are randomly assigned to these groups to establish partnerships between roosters and hens. *MN* hens are randomly selected to lead chicks, thereby determining the mother–child relationships.

Step 3: Evaluate whether the flock grouping and relationships require updating. If updates are needed, adjust the flock structure accordingly. If not, the positions of the roosters, hens, and chicks are updated individually, based on their specific strategies, while concurrently recalculating the fitness values for the new positions.

Step 4: Compare the fitness value of the new position with that of the original position. If the new position has a lower (better) fitness value, update the individual's location; otherwise, retain the original position.

Step 5: Determine whether stopping criteria have been met. If so, terminate the iteration and output the optimal solution. If not, return to step 3 and continue the iterative process in a loop for further exploration.

Following the described CFO process, we can observe that the CFO algorithm possesses a clear structure and is easy to implement, amongst other advantages. However, it still exhibits certain limitations and defects. When addressing high-dimensional complex problems, it tends to fall into local optima and suffers from low precision. The primary reasons for this are as follows:

Firstly, the CFO algorithm relies on randomness in the generation of the initial population, which may lead to insufficient diversity and consequently affect the algorithm's ability to search for the global optimal solution. The random generation of the initial population may cause the algorithm to concentrate excessively in certain areas of the solution space, neglecting other regions that may contain superior solutions.

Secondly, the CFO primarily depends on local changes of individuals for search, such as in its rooster search phase. This means that the algorithm may focus too much on the vicinity of the current solution during the iteration process, increasing the risk of falling into local optima.

Thirdly, it only uses simply generated random numbers to perturb the position of individuals. Although this method is straightforward and easy to implement, it may not effectively guide the algorithm out of the local optimal area. The simplicity of the random perturbation may lead to a lack of sufficient dynamics and adaptability in the search process, thereby affecting the quality of the solution.

Figure 5 illustrates the process of using the CFO to solve the function $f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$, which is a multimodal function prone to trapping optimisation algorithms in local optima. To fully observe the performance of the algorithm, the number of iterations was set to 1500. It can be seen that the CFO found a local minimum value of 3.11665 at the 683rd iteration. Considering the theoretical minimum value of the function is 0, this result still has a significant gap from the global optimum, highlighting the algorithm's tendency to fall into local optima.

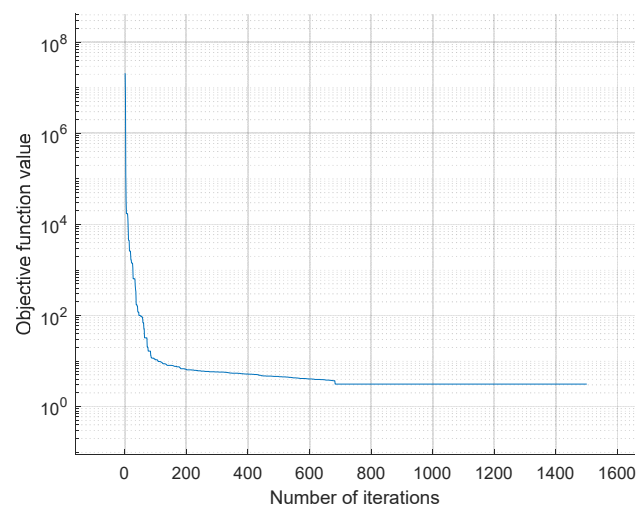


Figure 5. The process of using the CFO algorithm to solve a multimodal function.

3.2. Improved Chicken Flock Optimisation Algorithm

The parameter optimisation of ADRC presented in this paper is classified as a multimodal nonlinear optimisation problem. The objective function to be refined is characterised by the absence of an analytical formula, a multitude of inflection and extremum points, indeterminate local behaviour, and susceptibility to the influence of the initial solution. To ensure that the CFO algorithm can be effectively applied to this optimisation challenge, we have made a series of improvements. Next, we detail these improvements.

3.2.1. Initial Population Improvement through Tent Mapping

The quality of the initial population significantly impacts the iterative process of metaheuristic algorithms. In the standard CFO algorithm, the initial population is randomly generated, often resulting in a lack of diversity and representativeness, thereby limiting the algorithm's ability to solve complex problems. To address this issue, we introduce the concept of chaotic mapping, which is widely used in optimisation algorithms due to its unpredictability and sensitivity in dynamic systems [36,37].

We choose the simple and efficient tent map as a basis and make improvements to it. This improved tent map combines the complex dynamic characteristics of deterministic chaotic systems with random perturbations. By introducing a random factor, we add extra uncertainty while maintaining the inherent complexity of the chaotic system, which helps to generate a more diversified initial population. The proposed method not only enhances the diversity of the population but also strengthens the algorithm's exploration capabilities in high-dimensional spaces. By introducing a broader coverage of the solution space at the initial stage, the algorithm can more effectively avoid converging to local optima early on, laying a solid foundation for subsequent global optimisation.

The calculation process of the improved tent map is detailed in Equation (22):

$$x_{ij} = \begin{cases} 2p_{ij} + \text{rand} \times \frac{1}{N} & 0 \leq p_{ij} \leq \frac{1}{2} \\ 2(1 - p_{ij}) + \text{rand} \times \frac{1}{N} & \frac{1}{2} < p_{ij} \leq 1 \end{cases} \quad (22)$$

where p_{ij} is the j th dimension value of the i th individual as it is initially and randomly generated, x_{ij} is the j th dimension value of the i th individual after it has undergone the improved tent mapping, and rand represents a number that is randomly generated within the range $[0, 1]$, introducing an element of randomness to the mapping. N signifies the size of the initial population.

3.2.2. Adaptive Lévy Flight Strategy

In the standard CFO algorithm, the update strategy for roosters primarily relies on simple changes to the individual's current position, which limits the algorithm's exploration capabilities. To further enhance the search performance and encourage individuals within the population to explore different regions of the solution space, we introduce an adaptive Lévy flight strategy. Lévy flight is particularly advantageous due to its heavy-tailed step size distribution, which means that it can occasionally take very large steps. This property enables the algorithm to escape from local optima effectively and to explore the global solution space more efficiently. Additionally, Lévy flight is scale-free and can adapt to various scales of problems, making it a robust choice for global optimisation [38]. Figure 6 illustrates the trajectory of Lévy flight, showing how it navigates through the solution space with a combination of long-range exploration and local searches.

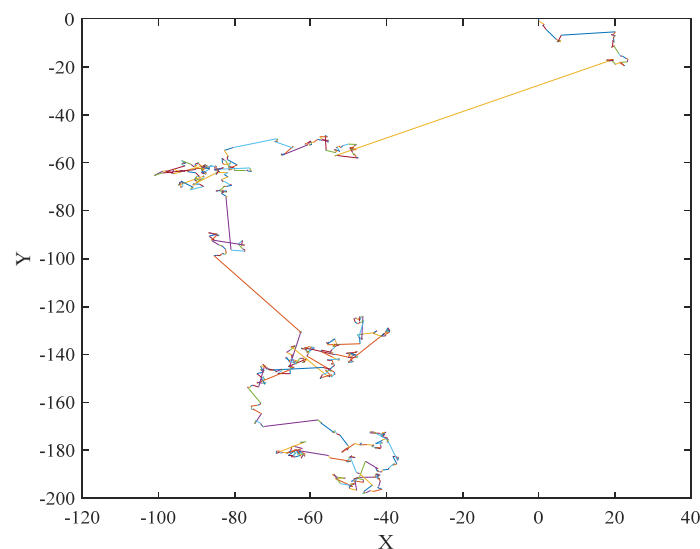


Figure 6. Lévy flight trajectory.

However, it should be noted that in the Lévy flight strategy, the step size scaling factor α is usually set to a fixed value. This approach may not be flexible enough to respond to the varying needs of different search stages [39]. To address this, we adapt the step size scaling factor α to adjust dynamically as follows:

$$\alpha = 0.5 \times \exp\left(-\frac{t}{t_{max}}\right) \quad (23)$$

where α is the Lévy flight step scaling factor, t is the current number of iterations, and t_{max} is the maximum number of iterations.

The improved adaptive Lévy flight strategy allocates a larger step size during the early stages of the algorithm, which helps the population explore the solution space widely and increases the chance of finding the global optimum. As the iterations continue, the step size is gradually reduced, focusing the search more narrowly and thus improving the accuracy and efficiency of the search in local areas. Following this, the rooster search behaviour in the ICFO algorithm is as follows:

$$\begin{cases} x_{ij}(t+1) = x_{ij}(t)(1 + randn(0, \sigma^2)), & \text{if } rand \leq p_c \\ x_{ij}(t+1) = x_{ij}(t) + a \times \text{Lévy}(\lambda) \times (x_{best,j}(t) - x_{ij}(t)), & \text{else} \end{cases} \quad (24)$$

where $x_{best,j}(t)$ is the j th dimensional value of the global optimal individual in the t th iteration, $x_{ij}(t)$ is the j th dimension value of the i th individual in the t th iteration, a is the scaling factor of the Lévy flight step length, as shown in Equation (23), $\text{Lévy}(\lambda)$ is the Lévy flight path, and the calculation method is shown in Equation (25).

$$\text{Lévy}(\lambda) = \frac{u}{|v|^{\frac{1}{\beta}}} \quad (25)$$

where β is usually taken between $[0, 2]$, and we set $\beta = 1.5$. $u \sim N(0, \sigma_u^2)$, $v \sim N(0, 1)$, and $\sigma_u = \left(\frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{2^{(\beta-1)/2}\Gamma(1+\beta)/2\beta} \right)^{1/\beta}$.

3.2.3. Dynamic Weight Factor

In the standard CFO algorithm, the local search function of the chicks is crucial for the algorithm to escape from local optima. However, the simulation of this behaviour in the standard CFO algorithm is only modulated by uniformly distributed random numbers. Although this method is uncomplicated, it also has certain limitations. For instance, it can lead to an uneven and incomplete search process, which impacts the overall performance of the algorithm.

To address this issue, a novel dynamic weighting strategy is introduced in the ICFO algorithm. This strategy integrates the periodic oscillation of the cosine function with the decay trend of an exponential decay term. The periodic oscillation ensures diversity in the search process, allowing the algorithm to switch between different search intensities, thus achieving a balance between local fine search and relatively large-scale exploration. Meanwhile, the overall decay trend ensures that the algorithm can gradually focus on the vicinity of the optimal solution as the iterations progress, enhancing the precision of the final solution.

The mathematical expression of the proposed dynamic weighting strategy is as follows:

$$w(t) = rand \cdot \left(\cos \left(2 \cdot \left(1 - \frac{10 \cdot t}{t_{max}} \right)^4 \right) - 1 \right) \quad (26)$$

where $w(t)$ is the dynamic weight factor at the t th iteration, $rand$ represents a number randomly generated in the $[0, 1]$ interval, $\cos$ is a cosine function, t is the current iteration number, and t_{max} is the maximum iteration number.

Figure 7 provides the variation curve for the weight w . The periodic characteristic of the cosine function is responsible for the weight's periodic oscillation throughout the iterative process. This oscillation is beneficial because it promotes a more agile exploration of the local area within the solution space.

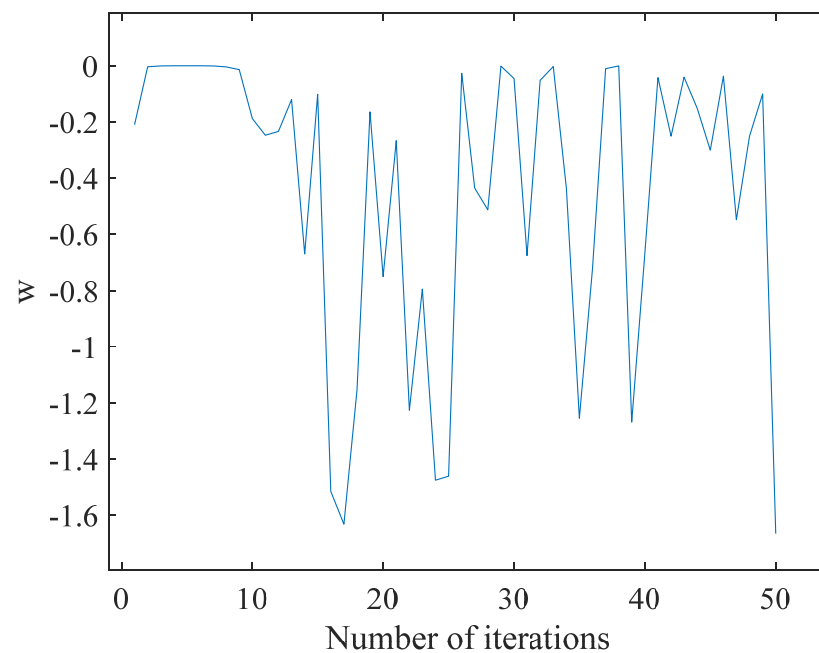


Figure 7. Dynamic weight factor iteration trajectory.

Finally, the chick search behaviour in ICFO is as follows:

$$x_{ij}(t+1) = w(t)x_{ij}(t) + F \cdot (x_{mj}(t) - x_{ij}(t)) \quad (27)$$

3.2.4. Random Individual Differential Mutation Strategy

After completing the iterative process of the ICFO algorithm, to further enhance its performance, we introduce the random individual differential mutation strategy. In each iteration, we randomly select K individuals for the differential mutation strategy. This mechanism, on one hand, maintains the tracking of the optimal solution, ensuring the convergence of the algorithm; on the other hand, it introduces randomness, increasing the diversity of the population and enhancing the algorithm's ability to escape from local optima. The expression is as follows:

$$x_{kj}(t+1) = rand \times (x_{best,j}(t) - x_{kj}(t)) + rand \times (x_{randj}(t) - x_{kj}(t)) \quad (28)$$

where $x_{kj}(t)$ is the value of the k th individual selected at the t th iteration in the j th dimension, $k \in \{1, 2, \dots, K\}$.

According to the defined mutation strategy, the ICFO algorithm can not only maintain the tracking of the optimal solution but also effectively explore the unknown regions within the solution space, thereby enhancing its global search capability. The introduction of this strategy aids in breaking away from local optimal solutions that the algorithm might otherwise become trapped in, promotes diversity within the population, and ultimately increases the likelihood of discovering the global optimal solution.

3.2.5. Overall Algorithm Framework

By incorporating the aforementioned improvement strategies with the steps of the standard COA, the pseudocode of ICOA (Algorithm 1) is as follows:

Algorithm 1: ICFO Main Loop.

```

01: Input: Problem parameters,  $N, T_{max}, P_c, K, G, RN, CN, MN$ 
02: Output: The best individual
02: Generate initial population using improved tent mapping
03: Obtain the global best individual
04:  $t = 1$ 
05: While  $t \leq T_{max}$ 
06:   If  $t = 1$  or  $mod(t, G) = 1$ 
07:     Divide the population into roosters, hens, and chicks groups
08:     Determine the relationship between chicks and hens within the group
09:   End if
10:   For  $i = 1:N$ 
11:     If  $i == \text{rooster}$ 
12:       If  $rand \leq P_c$ 
13:         Update its position based on Equation (24a)
14:       Else
15:         Update its position based on Equation (24b)
16:       End if
17:     End if
18:     If  $i == \text{hen}$ 
19:       Update its position based on Equation (20)
20:     End if
21:     If  $i == \text{chick}$ 
22:       Update its position based on Equation (27)
23:     End if
24:   End for
25:   Get new individuals
26:   Update individuals and global best individual
27:   Randomly select  $K$  individuals
28:   For  $k = 1:K$ 
29:     Update current individual based on Equation (28)
30:   End for
31:   Get new individuals
32:   Update individuals and global best individual
33:    $t = t + 1$ 
34: End while
35: Output the best individual

```

3.3. ADRC Parameter Tuning by the ICFO Algorithm

In the process of optimising ADRC with ICFO, we need to determine its 12 parameters. The TD module requires the design parameters R and δ_0 . The ESO module requires the design parameters $\beta_{01}, \beta_{02}, \beta_{03}, \alpha_1$, and δ_1 . The NLSEF module requires the design parameters δ, β_1, β_2 , and α_2 . For systems with disturbances, it is also necessary to determine the parameter b_0 . In addition, the Integrated Time and Absolute Error (ITAE) index of the vehicle roll angle is used as the objective function, as depicted in Equation (29):

$$\text{Objective function} = f(x) = \int_0^{t_e} t|e(t)|dt \quad (29)$$

where $e(t) = \phi_d - \phi$, ϕ_d is the desired roll angle.

Then, by incorporating the aforementioned ICFO algorithm procedure, the ADRC parameter optimisation process subsequent to the invocation of the stabiliser simulation model is illustrated in Figure 8.

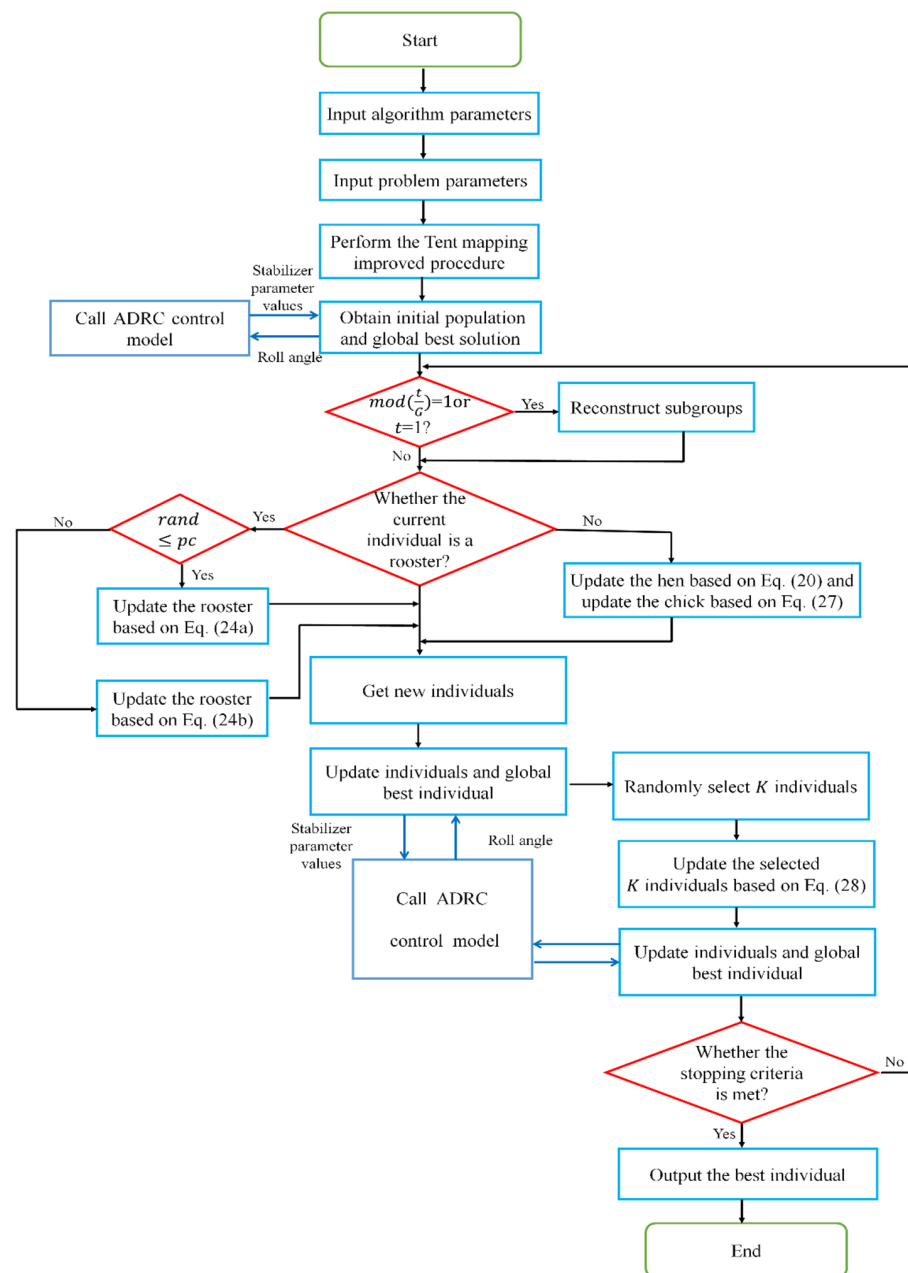


Figure 8. Flowchart of ICFO algorithm for stabiliser ADRC parameter tuning.

4. Experiment and Numerical Simulation

4.1. Algorithm Performance Analysis

In this section, we validate the performance of the ICFO algorithm by designing a series of assessment experiments. These experiments are based on six carefully selected CEC2005 benchmark test functions. These functions cover the diversity and complexity of real-world problems [40], enabling a comprehensive evaluation of the performance of ICFO across different aspects.

1. Benchmark test functions 1 and 2 are unimodal; they each contain only one global optimum. These functions test the algorithm's convergence rate and precision in locating the optimal solution.

2. Benchmark test functions 3 and 4 are multimodal and contain multiple local optima. These functions test the algorithm's ability to avoid local optima and continue searching for the global optimum, which is crucial for measuring the algorithm's robustness.
3. Benchmark test functions 5 and 6 are composites that integrate various mathematical properties to simulate more complex search scenarios. These functions test the algorithm's exploration capabilities and adaptability in complex and variable environments.

Table 1 details these test functions, including their mathematical formulations, the dimensions of the search space, and the locations of their optima.

Table 1. Overview of test functions.

Function	Dimension (n)	Value Range	Minimum Value
$f_1(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - \frac{2}{x_i})^2 + (x_i - 1)^2]$	30	$[-30, 30]^n$	0
$f_2(x) = \sum_{i=1}^n i x_i^4 + \text{random}[0, 1)$	30	$[-1.28, 1.28]^n$	0
$f_3(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos 2\pi x_i\right) + 20 + e$	30	$[-32, 32]^n$	0
$f_4(x) = \frac{\pi}{n} \left\{ 10 \sin^2(\pi y_i) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] \right.$ $\left. + (y_n - 1)^2 \right\} + \sum_{i=1}^n u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{1}{4}(x_i + 1)$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a, \\ 0, & -a \leq x_i \leq a, \\ k(-x_i - a)^m, & x_i < -a. \end{cases}$	30	$[-50, 50]^n$	0
$f_5(x) = \left[\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^n (x_i - a_{ij})^6} \right]^{-1}$	2	$[-65.536, 65.536]^n$	1
$f_6(x) = -\sum_{i=1}^{10} \left[(x - a_i)(x - a_i)^T + c_i \right]^{-1}$	4	$[0, 10]^n$	-10

Based on the aforementioned six benchmark test functions, we compare the ICFO algorithm with six efficient and advanced metaheuristic algorithms: CFO [33], sparrow search algorithm (SSA) [41], GWO [42], PSO algorithm [41,43], genetic algorithm (GA) [44,45], and evolutionary strategy (ES) [46]. The parameters for these algorithms are set in accordance with the outcomes of preliminary experiments and the suggestions found in these studies, as detailed in Table 2. The meanings of the symbols are referenced in the relevant literature that follows the algorithms.

Additionally, to ensure a fair comparison between different algorithms, the comparison should be based on the same number of fitness evaluations or an equivalent processing time. Therefore, in this study, the termination condition for all methods is a runtime limit of 10 s for each test function. This standard is primarily based on the work of Mernik et al. [47] and Draa [48]. Furthermore, to ensure statistical reliability, we conduct thirty independent runs for each algorithm and present the results in terms of the best, worst, and average values, and standard deviations.

All metaheuristic codes were run on MATLAB software R2020b, utilising a 64-bit 2.5 GHz Intel (R) Core (TM) i7 processor and 8 GB of RAM. Furthermore, the simulations detailed in Section 4.2 were conducted using a co-simulation between Simulink and CarSim 2019 within the MATLAB R2020b environment. The final results are shown in Table 3. In Figure 9, we also display the convergence curves when each algorithm achieves its best value.

Table 2. Parameter settings for each algorithm.

Algorithms	Parameter Settings
ICFO	Population size = 100, $RN = 10$, $MN = 30$, $CN = 60$, $G = 10$, $p_c = 0.4$, $K = 20$
SSA	Population size = 100, the number of the producers and SD accounts for 20% and 10%, respectively, and $ST = 0.8$.
GWO	Population size = 100, $\vec{a}$ value is linearly decreased from 2 to 0 and r_1, r_2 are random vectors in $[0, 1]$
PSO	Population size = 100, $w = 0.7298$, $c_1, c_2 = 1.4995$
GA	Population size = 100, Mutation rate = 0.2, Mutation percentage = 0.05, Crossover rate = 0.5, Crossover percentage = 0.75
CFO	Population size = 100, $RN = 20$, $MN = 30$, $CN = 50$, $G = 5$
ES	Population size = 100, $\lambda = 3$

Table 3. Comparison results of various algorithms (worst value, W; average value, A; best value, B; standard deviation, SD).

Function		ICFO	PSO	GWO	SSA
f_1	W	2.2942×10^{-21}	1.1964×10^{-13}	7.1791	1.7195×10^{-8}
	A	3.9859×10^{-22}	1.459×10^{-14}	5.7575	9.535×10^{-14}
	B	2.9085×10^{-26}	2.0974×10^{-21}	4.3721	7.1847×10^{-13}
	SD	6.9979×10^{-22}	3.7148×10^{-14}	0.80144	5.4078×10^{-9}
f_2	W	4.3498×10^{-6}	6.6448×10^{-5}	2.4575×10^{-5}	2.4898×10^{-5}
	A	1.9831×10^{-6}	2.7329×10^{-5}	1.0294×10^{-5}	1.0165×10^{-5}
	B	9.7921×10^{-7}	1.0329×10^{-5}	1.9978×10^{-6}	4.2416×10^{-6}
	SD	1.2433×10^{-6}	1.8349×10^{-5}	7.3439×10^{-6}	6.9366×10^{-6}
f_3	W	8.8818×10^{-16}	4.4409×10^{-15}	4.4409×10^{-15}	1.5099×10^{-14}
	A	8.8818×10^{-16}	3.0198×10^{-15}	4.0856×10^{-15}	5.862×10^{-15}
	B	8.8818×10^{-16}	8.8818×10^{-16}	8.8818×10^{-16}	8.8818×10^{-16}
	SD	0	1.8346×10^{-15}	1.1235×10^{-15}	4.1701×10^{-15}
f_4	W	4.8084×10^{-32}	4.7116×10^{-32}	0.0027216	4.9052×10^{-32}
	A	4.7504×10^{-32}	4.7116×10^{-32}	0.0020007	4.7528×10^{-32}
	B	4.7116×10^{-32}	4.7116×10^{-32}	0.001565	4.7116×10^{-32}
	SD	4.9991×10^{-34}	1.154×10^{-47}	0.00038726	6.6574×10^{-34}
f_5	W	2.9821	0.998	10.7632	12.6705
	A	1.1964	0.998	4.3244	6.8419
	B	0.998	0.998	0.998	0.998
	SD	0.62743	0	4.513	5.8001
f_6	W	-10.5364	-2.4273	-9.5916	-10.5364
	A	-10.5364	-9.1894	-9.9846	-10.5364
	B	-10.5364	-10.5364	-10.3977	-10.5364
	SD	1.6748×10^{-15}	2.9126	0.23307	2.1349×10^{-15}
Function		CFO	GA	ES	
f_1	W	4.2107×10^{-5}	3.9866	1.5137×10^{-13}	
	A	2.1598×10^{-11}	0.79732	3.0284×10^{-14}	
	B	4.895×10^{-22}	4.1564×10^{-19}	6.7278×10^{-20}	
	SD	4.8293×10^{-11}	1.7829	6.7691×10^{-14}	
f_2	W	2.2521×10^{-5}	3.7434×10^{-5}	2.5959×10^{-5}	
	A	1.0625×10^{-5}	2.6329×10^{-5}	1.9328×10^{-5}	
	B	2.0155×10^{-6}	1.9305×10^{-5}	1.1657×10^{-5}	
	SD	1.0597×10^{-5}	8.2253×10^{-6}	5.4669×10^{-6}	
f_3	W	4.4409×10^{-15}	1.5099×10^{-14}	4.4409×10^{-15}	
	A	3.7303×10^{-15}	9.4147×10^{-15}	4.4409×10^{-15}	
	B	8.8818×10^{-16}	7.9936×10^{-15}	4.4409×10^{-15}	
	SD	1.5888×10^{-15}	3.1776×10^{-15}	0	
f_4	W	4.8084×10^{-32}	5.0989×10^{-32}	4.9537×10^{-32}	
	A	4.7552×10^{-32}	4.8133×10^{-32}	4.823×10^{-32}	
	B	4.7116×10^{-32}	4.7116×10^{-32}	4.7116×10^{-32}	
	SD	5.0199×10^{-34}	1.645×10^{-33}	1.1556×10^{-33}	
f_5	W	2.9821	2.9821	2.9821	
	A	2.0326	0	2.1885	
	B	0.998	2.9821	0.998	
	SD	1.2867	0	1.0867	
f_6	W	-10.5364	-9.7008	-10.1194	
	A	-10.5364	-10.0701	-10.2984	
	B	-10.5364	-10.2887	-10.4772	
	SD	1.7764×10^{-15}	0.2328	0.16526	

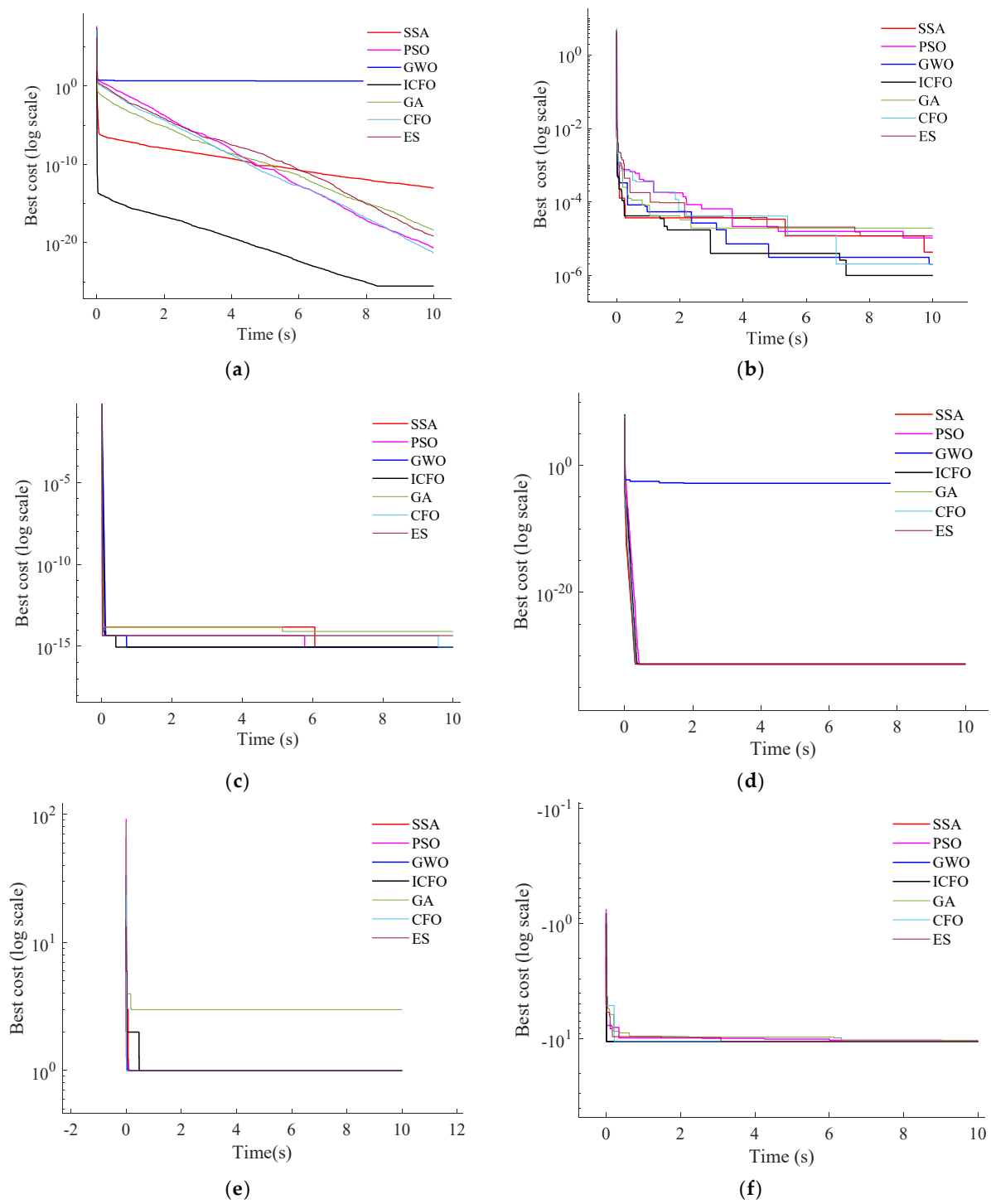


Figure 9. Convergence curves of best performance for each algorithm based on f_1 (a), f_2 (b), f_3 (c), f_4 (d), f_5 (e), f_6 (f).

Based on the results in Table 3 and Figure 9, it can be observed that the ICFO algorithm performs exceptionally well in this benchmark test. Whether on the unimodal functions f_1 and f_2 or the more complex multimodal functions such as f_3 and the composite functions f_5 and f_6 that incorporate various mathematical properties, ICFO demonstrates advantages in quickly locating optimal solutions, high stability, and precision. In the unimodal function tests, ICFO not only converges rapidly, but also provides stable and reliable results with the smallest standard deviation, demonstrating its superior accuracy. When faced with the multimodal function f_3 , ICFO effectively avoids becoming trapped in local optima

while maintaining the smallest standard deviation, showing excellent global search capabilities and stability. On the composite function f_6 , ICFO once again demonstrates its adaptability and efficiency in complex search scenarios, achieving the smallest values of all the statistical indicators, including the smallest standard deviation. Although f_4 and f_5 are slightly inferior to PSO, the ability of ICFO to find the optimal solution is still reasonable. Additionally, it needs to be pointed out that ICFO outperformed CFO across all test functions, demonstrating the efficacy of the enhanced strategies within ICFO. Considering the performance across all test functions, the speed, stability, accuracy, and robustness of the ICFO algorithm make it a promising algorithm for a variety of optimisation problems, especially in scenarios where rapid, stable, and precise identification of the global optimum is needed, where the advantages of ICFO are particularly prominent.

4.2. Significance Analysis

To validate the reliability of the ICFO algorithm's performance, this section employs statistical analysis methods to compare the performance of the ICFO algorithm with other algorithms. We have chosen the Wilcoxon signed-rank test as our analytical tool and, following recommendations in the literature, set the significance level p -value to 0.05 [49]. The test results are presented in Table 4.

Table 4. Wilcoxon signed-rank test results for algorithm performance comparison.

f_1	SSA	PSO	GWO	CFO	GA	ES
ICFO	0.8657×10^{-5}	0.00195	0.00183	0.00267	0.0053	0.00248
f_2	SSA	PSO	GWO	CFO	GA	ES
ICFO	0.00231	0.04380	1.2345×10^{-5}	0.00189	0.02987	0.00231
f_3	SSA	PSO	GWO	CFO	GA	ES
ICFO	0.00912	0.02998	0.01543	0.00156	0.02387	0.03360
f_4	SSA	PSO	GWO	CFO	GA	ES
ICFO	0.09755	0.12358	0.01789	0.00203	0.00175	1.7263×10^{-5}
f_5	SSA	PSO	GWO	CFO	GA	ES
ICFO	0.01123	0.04562	0.02012	0.01452	0.04478	0.01123
f_6	SSA	PSO	GWO	CFO	GA	ES
ICFO	0.01045	0.03234	0.01987	0.0172	0.00587	0.00662

The results in Table 4 reveal that in most test cases, the p -value is below the 0.05 threshold, indicating significant differences in performance between the ICFO algorithm and other algorithms. It is particularly noteworthy that for the test function f_4 there is no statistical difference between the ICFO algorithm and the best-performing PSO and SSA algorithms, further confirming the effectiveness of the ICFO algorithm.

4.3. Optimising the ADRC Parameters for Active Stabiliser Bars with the ICFO Algorithm

In this section, we employ a light-duty bus as a case study to design and validate the proposed control and optimisation methodology. The light-duty bus is chosen due to its higher centre of gravity, which, when combined with high-speed manoeuvres or significant inclines, results in substantial roll angles and an increased propensity for rollover incidents. Given that these vehicles are intended for passenger transport, the consequences of a rollover can be particularly severe in terms of injuries and fatalities. The specific parameters of the light bus are detailed in Table 5.

In addition, following pre-experimental tuning, the ICFO parameters are set as follows: $T_{max} = 200$, $N = 100$, $RN = 20$, $MN = 30$, $CN = 50$, $G = 10$, $p_c = 0.4$, and $K = 10$. Subsequently, upon running the program, the optimal ADRC parameters for the active stabiliser bar are obtained, as listed in Table 6.

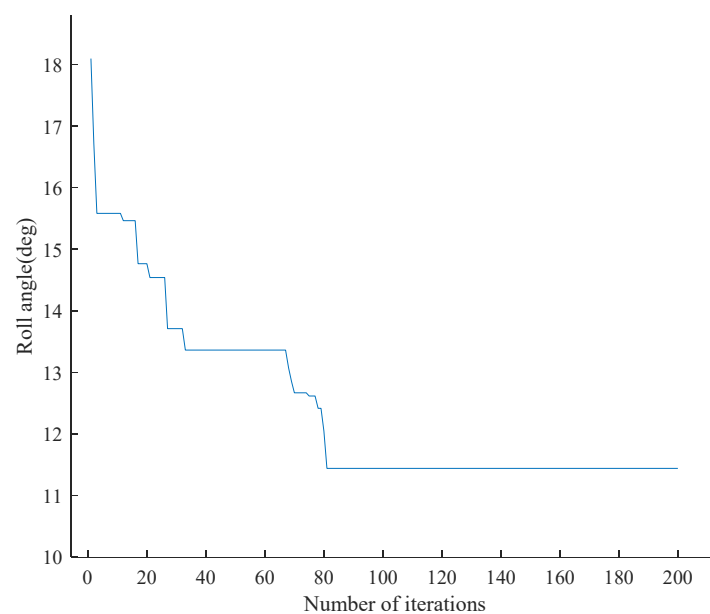
Table 5. Model parameters.

Variables	Parameter Name and Unit	Parameter Value	Variables	Parameter Name and Unit	Parameter Value
M	Vehicle mass/kg	2700	I_{xx}	Moment of inertia of the body about the axle/kg·m ²	6000
g	Gravitational acceleration/m/s ²	9.8	I_{zz}	Moment of inertia of the body about the axle z/kg·m ²	4500
m_f	Front unsprung mass/kg	125.0	I_{xz}	product of inertia of the vehicle around the XZ-axis /kg·m ²	790
m_r	Rear unsprung mass/kg	147.1	u_c	Vehicle speed/m/s	60
m_b	Sprung mass/kg	2427.9	K_ϕ	Body roll stiffness/N·m/rad	131,380
a	Centre of mass to front axle/m	1.6	D_ϕ	Body roll damping/N·m·s/rad	4200
b	Centre of mass to rear axle distance/m	1.7	K_T	Stabiliser bar torsional stiffness/N·m/rad	206,611
h_s	Height of centre of roll from centre of mass/m	0.5	d_f	Front tire deflection per body roll angle/m/rad	0.00018
k_f	Front tire cornering stiffness/m/rad	59,780	d_r	Rear tyre deflection per body roll angle/m/rad	−0.00011
k_r	Rear tire cornering stiffness/m/rad	101,920			

Table 6. Optimal ADRC parameter values determined by the ICFO algorithm.

Parameter	Value
R	10.40
δ_0	0.009
β_{01}	80.00
β_{02}	77.43
β_{03}	64.06
α_1	0.61
δ_1	0.0008
δ	0.0803
β_1	118.19
β_2	8.63
α_2	1.5
b_0	0.8

The convergence curve of the ICFO algorithm is depicted in Figure 10. Figure 10 shows that after 200 rounds of iteration, the roll angle decreases from 18 degrees to 11.5 degrees, with a relative improvement of 36.11%, which shows a remarkable parameter-tuning result.

**Figure 10.** Roll angle convergence curve.

4.4. Performance Analysis of ICFO-ADRC Control

4.4.1. Comparison with Other Controllers

To demonstrate the effectiveness of the ICFO-ADRC control, we compare its performance with that of a passive stabiliser bar, fuzzy PI-PD control, non-optimised ADRC control, and CFO-ADRC control across various driving conditions, including hyperbola, serpentine, drift, and obstacle avoidance scenarios. The CFO parameters are set as follows: $T_{max} = 200$, $N = 100$, $RN = 10$, $MN = 30$, $CN = 60$, $G = 10$. The results of the simulations are presented in Figures 11–14.

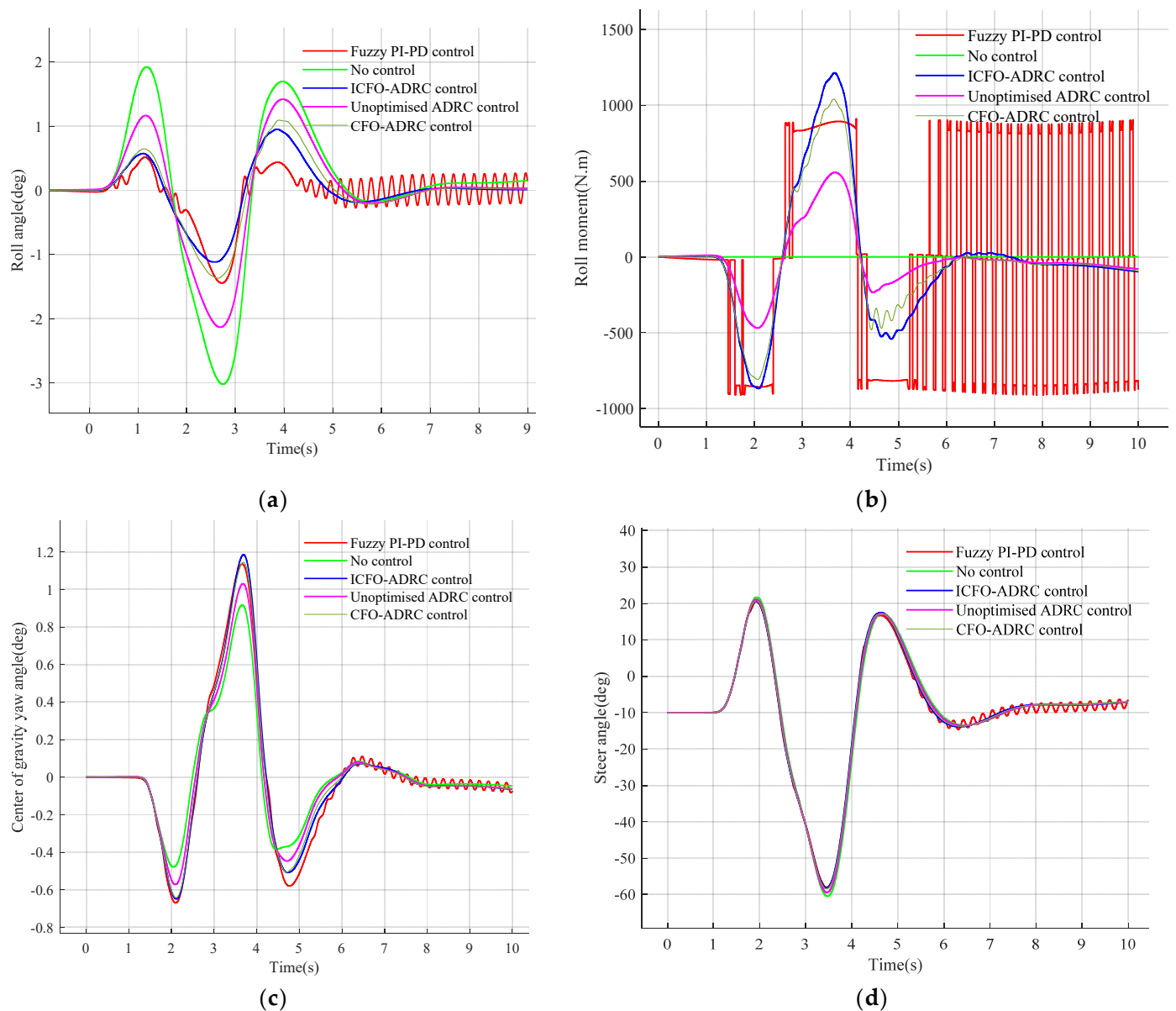


Figure 11. Simulation results of different controllers under hyperbola driving conditions: roll angle (a), anti-roll torque (b), centre of gravity yaw angle (c), steer angle (d).

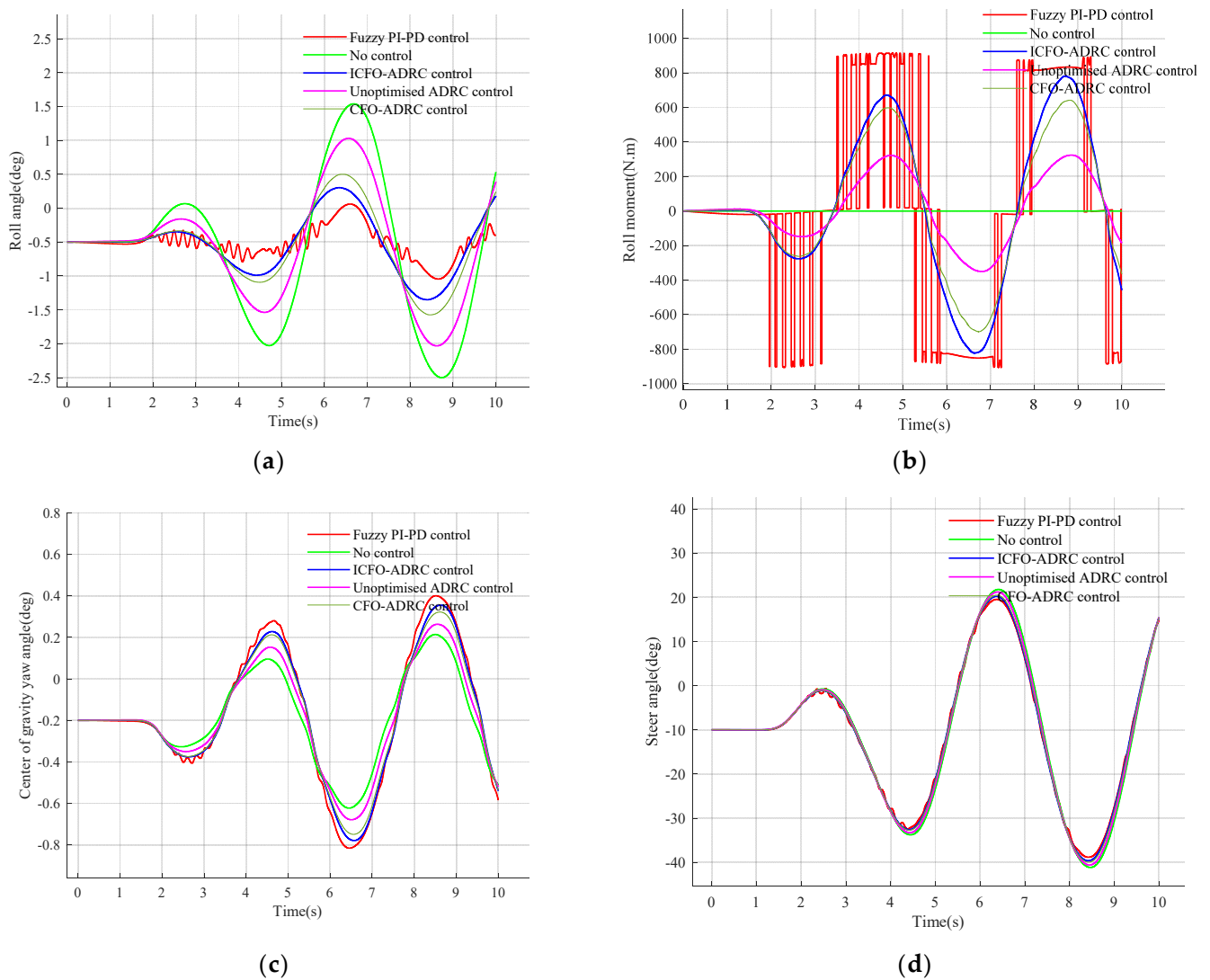


Figure 12. Simulation results of different controllers under serpentine driving conditions: roll angle (a), anti-roll torque (b), centre of gravity yaw angle (c), steer angle (d).

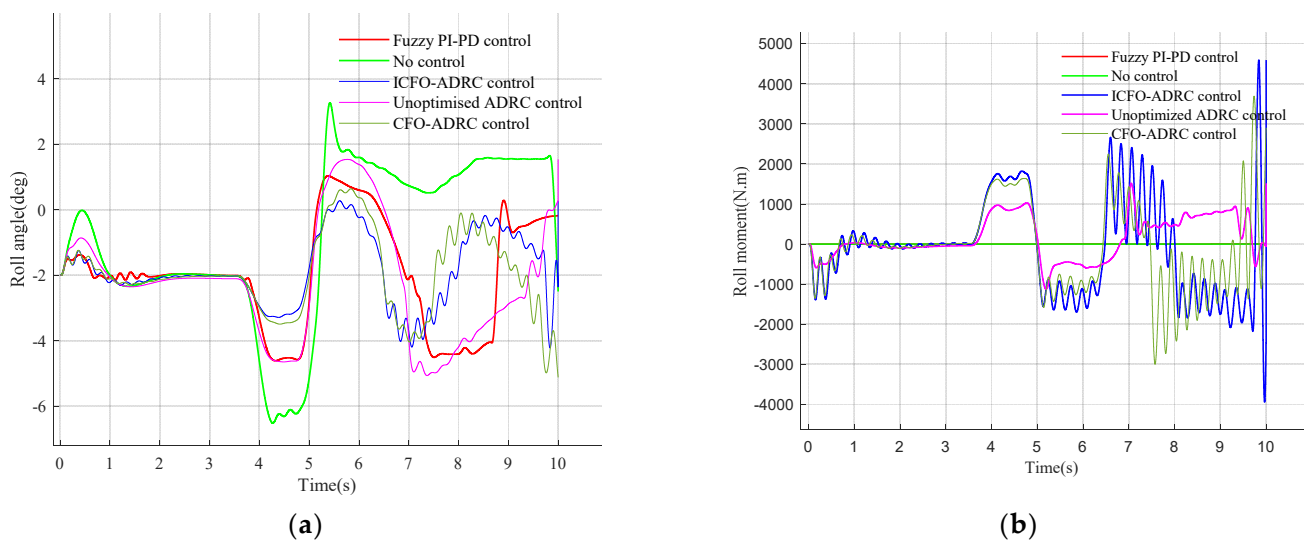
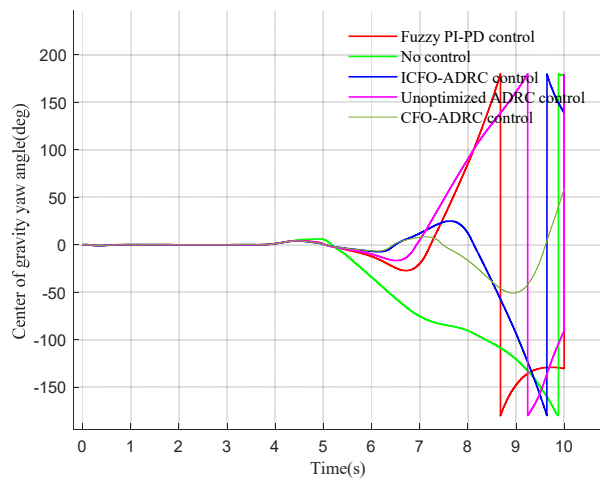
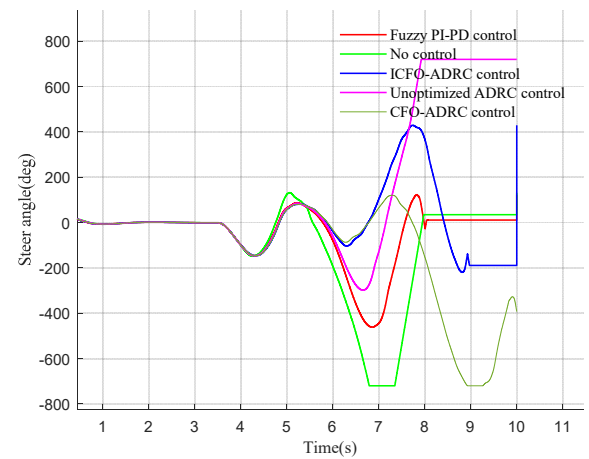


Figure 13. Cont.

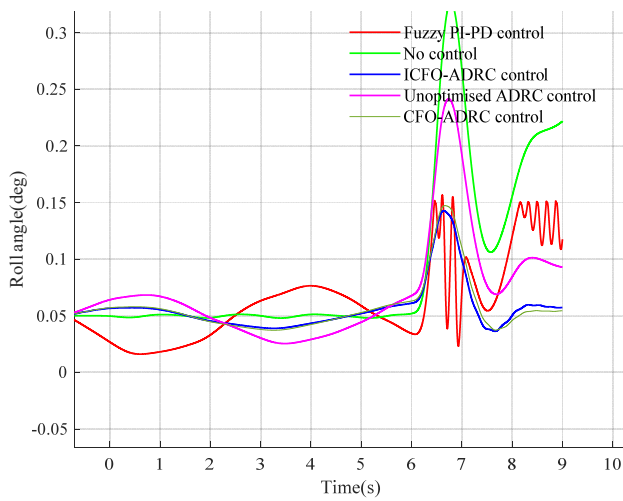


(c)

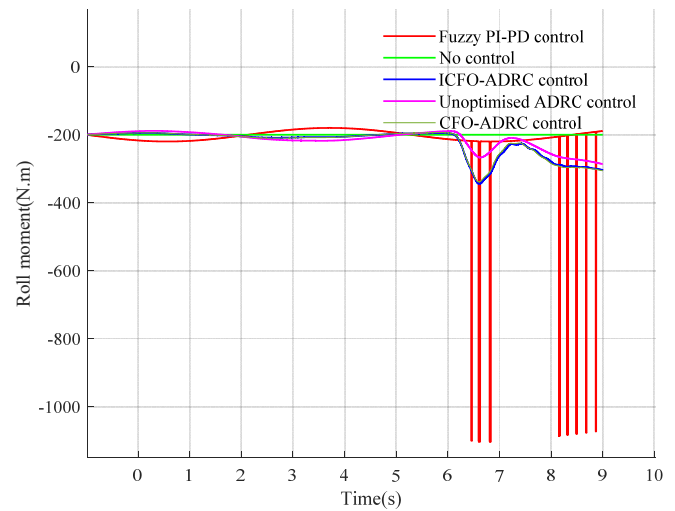


(d)

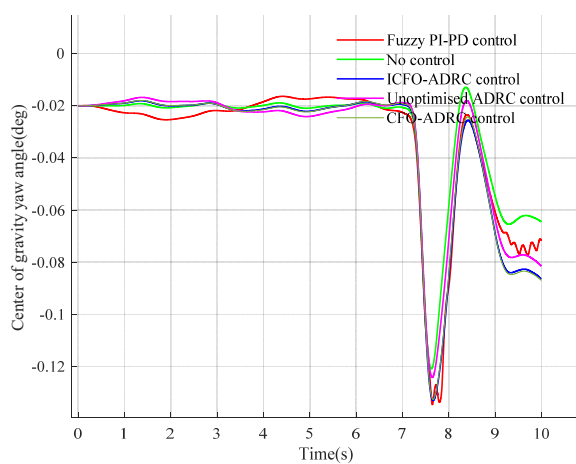
Figure 13. Simulation results of different controllers under drift driving conditions: roll angle (a), anti-roll torque (b), centre of gravity yaw angle (c), steer angle (d).



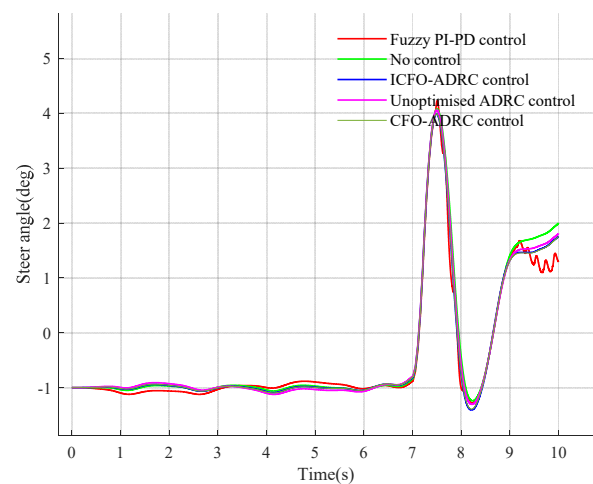
(a)



(b)



(c)



(d)

Figure 14. Simulation results of different controllers under obstacle avoidance driving conditions: roll angle (a), anti-roll torque (b), centre of gravity yaw angle (c), steer angle (d).

The simulation results from the aforementioned four driving conditions were aggregated, yielding the results presented in Table 7. The term “improvement in roll angle” refers to the degree of improvement compared to the passive stabiliser.

Table 7. Comparison of control effect values.

Control Mode	Driving Condition	Roll Angle (deg)	Anti-Roll Torque (N·m)	Center of Gravity Yaw Angle (deg)	Steer Angle (deg)	Improvement in Roll Angle (%)
Passive stabiliser	Hyperbola	3.01	0	0.92	50.46	-
	Serpentine	2.04	0	0.42	31.78	-
	Drift	5.27	0	179.99	720	-
	Obstacle avoidance	0.28	0	0.10	5.08	-
Fuzzy PI-PD control	Hyperbola	1.45	911	1.14	48.00	51.83
	Serpentine	0.56	915	0.62	29.52	72.55
	Drift	3.03	945	180	461	42.50
	Obstacle avoidance	0.11	904	0.11	5.23	60.71
Unoptimised ADRC control	Hyperbola	2.13	557	1.03	49.42	29.24
	Serpentine	1.53	350	0.49	31.20	25.00
	Drift	3.53	1516	179.96	720	33.02
	Obstacle avoidance	0.19	86	0.10	5.06	47.37
ICFO-ADRC control	Hyperbola	1.12	1214	1.18	48.19	62.79
	Serpentine	0.85	822	0.58	30.22	58.33
	Drift	2.27	4589	179.99	429.12	56.93
	Obstacle avoidance	0.09	144	0.113	4.99	67.86
CFO-ADRC control	Hyperbola	1.36	1041	1.14	48.43	54.82
	Serpentine	1.07	698	0.55	30.50	47.55
	Drift	3.11	3688	58.94	720	40.99
	Obstacle avoidance	0.10	138	0.11	4.99	64.29

In summary, the active roll stabiliser integrated with the optimised ADRC controller demonstrates excellent lateral stability, with performance improvements that range from 56.93% to 67.86%. In the three complex road conditions of hyperbolic driving, drifting, and obstacle avoidance, the system achieves the best improvement in roll angle. In four different road condition tests, the ICFO-ADRC controller shows an average performance increase of 8% compared to the fuzzy PI-PD controller, an average improvement of 82% compared to the non-optimised ADRC controller, and an average increase of 18% compared to the CFO-ADRC controller. This optimised system effectively reduces the roll angle of the vehicle during driving, ensuring the stability and safety of the vehicle.

4.4.2. Analysis of ICFO Effectiveness in Optimising ADRC Parameters

To validate the effectiveness of using ICFO for ADRC parameter optimisation, we compare its optimisation results with those obtained using CFO, based on Equation (29). To comprehensively assess the stability and reliability of the algorithm, the experiment was set to run for 7200 s. Other parameter settings remained the same as in the previous sections. Considering the randomness of metaheuristic algorithms, each optimisation algorithm was run independently ten times. The average objective function value for ICFO optimising ADRC parameters over ten runs was 13.8, whilst for CFO it was 15.1, demonstrating the performance advantage of using ICFO to optimise ADRC parameters. Furthermore, Figure 15 provides a more detailed statistical analysis through a box plot, the length of which further corroborates the robustness of ICFO-optimised ADRC. Finally, to verify the statistical significance of these performance improvements, we employed the Wilcoxon signed-rank test. The test results show a significance level of 0.002, further confirming the effectiveness and superiority of using ICFO in optimising ADRC parameters.

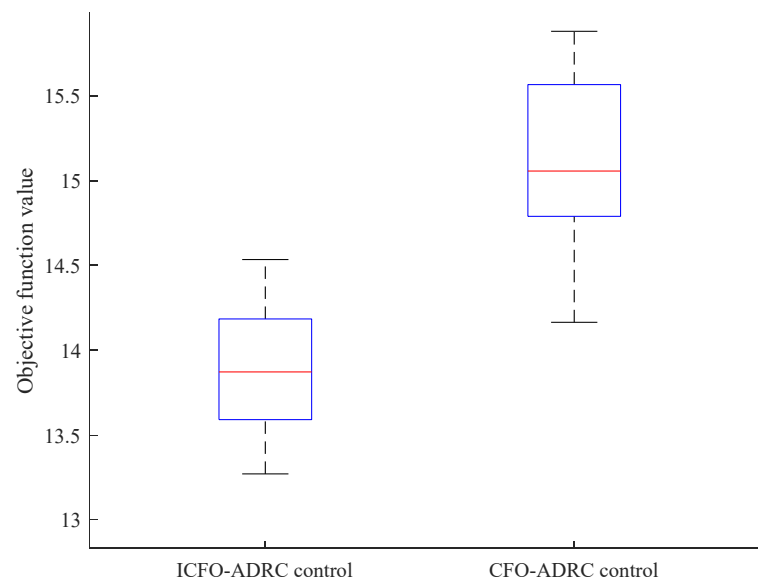


Figure 15. Box plot comparison of objective function values for ICFO and CFO in ADRC parameter optimisation.

5. Conclusions and Future Work

This paper focuses on active anti-roll control technology for lateral stability bars in passenger cars and describes detailed research and development. An ADRC controller is designed to manage the trade-off between the roll system's response time and the overshooting of the active lateral stability bar. An improved chicken flock optimisation algorithm is utilised to optimise the parameters within the ADRC's TD, ESO, and NLSEF modules, enhancing the system's robustness in controlling vehicle roll.

Using a whole vehicle model in CarSim and a controller model in Simulink, a joint simulation was performed to test the control effects under various conditions, such as double-lane-change and fishhook manoeuvres, for the passive stabiliser bar, fuzzy PI-PD controller, and both the non-optimised and optimised ADRC controllers. The results demonstrate that both controllers can suppress excessive roll angles during roll states, with the optimised ADRC controller exhibiting superior robustness, achieving a 56.93–67.86% improvement in control effects across the four conditions compared to the passive lateral stability bar model.

While this paper advances the field of active anti-roll control technology for lateral stability bars, there are some limitations. This study only considered the anti-roll control of the active stabiliser bar and did not account for the anti-roll role of the suspension. Future research could further explore the combined anti-roll control of the active stabiliser bar and active suspension, considering the coupling effects between them to design a composite ADRC for optimal anti-roll performance [34]. Furthermore, this study used an improved chicken flock optimisation algorithm to optimise the ADRC parameters and enhance its performance. In future research, other metaheuristic algorithms based on large neighbourhood search [50] and evolutionary methods [51] could be utilised to compare with our algorithm.

Author Contributions: Conceptualization, Q.Z.; methodology, Z.T., Q.Z. and X.Z.; supervision, D.T.P.; validation, Z.T., Q.Z. and X.Z.; writing—original draft, Z.T., Q.Z. and X.Z.; writing—review and editing, D.T.P. and X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to extend their appreciation to Northeast Forestry University for funding this work through the Central Universities Basic Scientific Research Business Innovation Project for Doctoral Students (Type A) (2572019AB25), Northeast Forestry University, Harbin, China.

Data Availability Statement: The original contributions presented in the study are included in the article; further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no competing interests.

References

1. Nguyen, D.N.; Dang, N.D.; Tran, T.T.H.; Hoang, T.B.; Nguyen, T.A. Effect of the passive stabilizer bar on the vehicle's stability. *Model. Simul. Eng.* **2022**, *2022*, 5523012. [\[CrossRef\]](#)
2. Gavhare, B.; Manivannan, P.V. Development of Hybrid Control Strategy for Dual Channel Electromechanical Stabilizer Bar for Roll Stabilization of Vehicle. In *Conference of Innovative Product Design and Intelligent Manufacturing System*; Springer Nature: Singapore, 2022; pp. 311–322.
3. Nguyen, T.A.; Hoang, T.B. Review on the stabilizer bar equipped with the vehicle. *J. Mech. Eng. Res. Dev.* **2021**, *44*, 156–161.
4. Cronje, P.H.; Els, P.S. Improving off-road vehicle handling using an active anti-roll bar. *J. Terramech.* **2010**, *47*, 179–189. [\[CrossRef\]](#)
5. Strassberger, M.; Guldner, J. BMW's dynamic drive: An active stabilizer bar system. *IEEE Control Syst. Mag.* **2004**, *24*, 28–29.
6. Nguyen, T.A. Establishing a novel adaptive fuzzy control algorithm for an active stabilizer bar with complex automotive dynamics model. *Ain Shams Eng. J.* **2024**, *15*, 102334. [\[CrossRef\]](#)
7. Nguyen, T.A. Proposing a novel fuzzy control algorithm for automotive active stabilizer bars based on views of roll stability and ride comfort. *Proc. Inst. Mech. Eng. Part K J. Multi-Body Dyn.* **2023**, *237*, 585–600. [\[CrossRef\]](#)
8. Nguyen, T.A. Improving the stability of the passenger vehicle by using an active stabilizer bar controlled by the fuzzy method. *Complexity* **2021**, *2021*, 6569298. [\[CrossRef\]](#)
9. Khalil, M.M.; Atia, M.R.; Mabrouk, M.H. Improving vehicle rollover resistance using fuzzy PID controller of active anti-roll bar system. *SAE Int. J. Passeng. Cars-Mech. Syst.* **2018**, *12*, 35–50. [\[CrossRef\]](#)
10. Sename, O.; Dugard, L.; Gaspar, P. H ∞ /LPV controller design for an active anti-roll bar system of heavy vehicles using parameter dependent weighting functions. *Heliyon* **2019**, *5*, e01827.
11. Sename, O.; Dugard, L.; Pham, T.P. H ∞ /LPV control for an active anti-roll bar system to improve the roll stability of heavy vehicles. In *Proceedings of the 10th National Conference on Mechanics and the 8th National Congress of the Vietnam Association for Mechanics*, Hanoi, Vietnam, 27 December 2017.
12. Kong, Z.; Pi, D.; Wang, X.; Wang, H.; Chen, S. Design and Evaluation of a Hierarchical Control Algorithm for an Electric Active Stabilizer Bar System. *Stroj. Vestn. J. Mech. Eng.* **2016**, *62*, 565–576. [\[CrossRef\]](#)
13. Vu, V.T.; Sename, O.; Bui, D.T. Improving Vehicle Roll Stability by LQR Active Anti-roll Bar Control. In *Advances in Engineering Research and Application: Proceedings of the International Conference, ICERA 2018*; Springer International Publishing: Berlin/Heidelberg, Germany, 2019; pp. 350–356.
14. Anafteh, B.; Moosavi, H.; Danesh, M. Active optimal roll control of railway vehicles in curved tracks using an electrically actuated anti-roll bar system. *Int. J. Control Autom. Syst.* **2023**, *21*, 1127–1142. [\[CrossRef\]](#)
15. Han, J. From PID to active disturbance rejection control. *IEEE Trans. Ind. Electron.* **2009**, *56*, 900–906. [\[CrossRef\]](#)
16. Gu, N.; Wang, D.; Peng, Z.; Wang, J.; Han, Q.L. Disturbance observers and extended state observers for marine vehicles: A survey. *Control Eng. Pract.* **2022**, *123*, 105158. [\[CrossRef\]](#)
17. Bilal, H.; Yin, B.; Aslam, M.S.; Anjum, Z.; Rohra, A.; Wang, Y. A practical study of active disturbance rejection control for rotary flexible joint robot manipulator. *Soft Comput.* **2023**, *27*, 4987–5001. [\[CrossRef\]](#)
18. 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.* **2018**, *34*, 2841–2855. [\[CrossRef\]](#)
19. Isdaryani, F.; Feriyonika, F.; Ferdiansyah, R. Comparison of Ziegler-Nichols and Cohen Coon tuning method for magnetic levitation control system. *J. Phys. Conf. Ser.* **2020**, *1450*, 012033. [\[CrossRef\]](#)
20. Hu, C.; Wu, F.; Zou, H. New PID parameter tuning based on improved dung beetle optimization algorithm. *Can. J. Chem. Eng.* **2024**. [\[CrossRef\]](#)
21. Gandomi, A.H.; Yang, X.S.; Talatahari, S.; Alavi, A.H. Metaheuristic algorithms in modeling and optimization. *Metaheuristic Appl. Struct. Infrastruct.* **2013**, *1*, 1–24.
22. Liu, Y.; As'array, A.; Ahmed, H.; Hairuddin, A.A.; Hassan, M.K.; Zakaria, M.Z.; Yang, S. Online optimal tuning of fuzzy PID controller using grey wolf optimizer for quarter car semi-active suspension system. *Adv. Mech. Eng.* **2024**, *16*, 16878132231219620. [\[CrossRef\]](#)
23. Mohd Tumari, M.Z.; Ahmad, M.A.; Suid, M.H.; Hao, M.R. An Improved Marine Predators Algorithm-Tuned Fractional-Order PID Controller for Automatic Voltage Regulator System. *Fractal Fract.* **2023**, *7*, 561. [\[CrossRef\]](#)
24. 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. [\[CrossRef\]](#)
25. 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. [\[CrossRef\]](#)
26. Yu, W.B. *A PSO Algorithm for the Parameter Tuning of ADRC and Its Application Research*; Shanghai University of Engineering Science: Shanghai, China, 2016.

27. Rivera, M.M.; Guerrero-Mendez, C.; Lopez-Betancur, D.; Saucedo-Anaya, T. Dynamical Sphere Regrouping Particle Swarm Optimization: A Proposed Algorithm for Dealing with PSO Premature Convergence in Large-Scale Global Optimization. *Mathematics* **2023**, *11*, 4339. [\[CrossRef\]](#)
28. Huang, Z.; Chen, Z.; Zheng, Y.; Sun, M.; Sun, Q. Optimal design of load frequency active disturbance rejection control via double-chains quantum genetic algorithm. *Neural Comput. Appl.* **2021**, *33*, 3325–3345. [\[CrossRef\]](#)
29. Shui, J.J.; Peng, D.G. Application research on improved genetic algorithm and active disturbance rejection control on quadcopters. *Meas. Control.* **2024**, *2024*, 00202940241240408. [\[CrossRef\]](#)
30. Shen, S.; Xu, J. Attitude active disturbance rejection control of the quadrotor and its parameter tuning. *Int. J. Aerosp. Eng.* **2020**, *2020*, 8876177. [\[CrossRef\]](#)
31. Chao, K.H.; Li, J.Y. Global maximum power point tracking of photovoltaic module arrays based on improved artificial bee colony algorithm. *Electronics* **2022**, *11*, 1572. [\[CrossRef\]](#)
32. Pietrenko-Dabrowska, A.; Koziel, S. Model management for low-computational-budget simulation-based optimization of antenna structures using nature-inspired algorithms. *Appl. Soft Comput.* **2024**, *154*, 111356. [\[CrossRef\]](#)
33. Meng, X.; Liu, Y.; Gao, X.; Zhang, H. A new bio-inspired algorithm: Chicken swarm optimization. In *Advances in Swarm Intelligence, Proceedings of the Advances in Swarm Intelligence: 5th International Conference, ICSI 2014, Hefei, China, 17–20 October 2014; Proceedings, Part I 5*; Springer International Publishing: Berlin/Heidelberg, Germany; pp. 86–94.
34. Deb, S.; Gao, X.Z.; Tammi, K.; Kalita, K.; Mahanta, P. Recent Studies on Chicken Swarm Optimization algorithm: A review (2014–2018). *Artif. Intell. Rev.* **2020**, *53*, 1737–1765. [\[CrossRef\]](#)
35. Wolpert, D.H.; Macready, W.G. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1997**, *1*, 67–82. [\[CrossRef\]](#)
36. Li, Y.; Han, M.; Guo, Q. Modified whale optimization algorithm based on tent chaotic mapping and its application in structural optimization. *KSCE J. Civ. Eng.* **2020**, *24*, 3703–3713. [\[CrossRef\]](#)
37. Chen, L.; Song, N.; Ma, Y. Harris hawks optimization based on global cross-variation and tent mapping. *J. Supercomput.* **2023**, *79*, 5576–5614. [\[CrossRef\]](#) [\[PubMed\]](#)
38. Campeau, W.; Simons, A.M.; Stevens, B. The evolutionary maintenance of Lévy flight foraging. *PLoS Comput. Biol.* **2022**, *18*, e1009490. [\[CrossRef\]](#) [\[PubMed\]](#)
39. Peng, L.; Zhang, D. An adaptive Lévy flight firefly algorithm for multilevel image thresholding based on Rényi entropy. *J. Supercomput.* **2022**, *78*, 6875–6896. [\[CrossRef\]](#)
40. Li, Y.F.; Zhan, Z.H.; Lin, Y.; Zhang, J. Comparisons study of APSO OLPSO and CLPSO on CEC2005 and CEC2014 test suits. In *Proceedings of the 2015 IEEE Congress on Evolutionary Computation (CEC)*, Sendai, Japan, 25–28 May 2015; pp. 3179–3185.
41. Xue, J.; Shen, B. A novel swarm intelligence optimization approach: Sparrow search algorithm. *Syst. Sci. Control Eng.* **2020**, *8*, 22–34. [\[CrossRef\]](#)
42. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [\[CrossRef\]](#)
43. Kennedy, J.; Eberhart, R. Particle swarm optimization. In *Proceedings of the ICNN'95-International Conference on Neural Networks*, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
44. Fathollahi-Fard, A.M.; Hajiaghahi-Keshteli, M.; Tavakkoli-Moghaddam, R. Red deer algorithm (RDA): A new nature-inspired meta-heuristic. *Soft Comput.* **2020**, *24*, 14637–14665. [\[CrossRef\]](#)
45. Holland, J.H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*; MIT Press: Cambridge, MA, USA, 1992.
46. Yao, X.; Liu, Y. Fast evolution strategies. In *International Conference on Evolutionary Programming*; Springer: Berlin/Heidelberg, Germany, 1997; pp. 149–161.
47. Mernik, M.; Liu, S.H.; Karaboga, D.; Črepinšek, M. On clarifying misconceptions when comparing variants of the artificial bee colony algorithm by offering a new implementation. *Inf. Sci.* **2015**, *291*, 115–127. [\[CrossRef\]](#)
48. Draa, A. On the performances of the flower pollination algorithm—Qualitative and quantitative analyses. *Appl. Soft Comput.* **2015**, *34*, 349–371. [\[CrossRef\]](#)
49. Li, Y.; Yu, X.; Liu, J. Enhanced butterfly optimization algorithm for large-scale optimization problems. *J. Bionic Eng.* **2022**, *19*, 554–570. [\[CrossRef\]](#)
50. Fathollahi-Fard, A.M.; Wu, P.; Tian, G.; Yu, D.; Zhang, T.; Yang, J.; Wong, K.Y. An efficient multi-objective adaptive large neighborhood search algorithm for solving a disassembly line balancing model considering idle rate, smoothness, labor cost, and energy consumption. *Expert Syst. Appl.* **2024**, *250*, 123908. [\[CrossRef\]](#)
51. Zhang, X.; Tian, G.; Fathollahi-Fard, A.M.; Pham, D.T.; Li, Z.; Pu, Y.; Zhang, T. A chance-constraint programming approach for a disassembly line balancing problem under uncertainty. *J. Manuf. Syst.* **2024**, *74*, 346–366. [\[CrossRef\]](#)

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.