mathematics

MDPI

Article

Attitude Control of UAVs with Search Optimization and Disturbance Rejection Strategies

Wensheng Li [1,2], Fanke Yang [3,*], Liqiang Zhong [1,2], Hao Wu [1,2], Xiangyuan Jiang [3,*] and Andrei V. Chukalin [4]

1 China Southern Power Grid Technology Co., Ltd., Guangzhou 510080, China; liwensheng@gddky.csg.cn (W.L.); zhongliqiang@gddky.csg.cn (L.Z.); wuhao@gddky.csg.cn (H.W.)
2 Guangdong Engineering Technology Research Center of Special Robots for Special Industries, Guangzhou 510080, China
3 Institute of Marine Science and Technology, Shandong University, Qingdao 266237, China
4 Laboratory of Interdisciplinary Problems in Energy Production, Ulyanovsk State Technical University, Ulyanovsk 432027, Russia; chukalin.andrej@mail.ru
* Correspondence: 202136981@mail.sdu.edu.cn (F.Y.); xyjiang@sdu.edu.cn (X.J.)

Abstract: This study aims to achieve rapid and stable control of quadrotor unmanned aerial vehicles' (UAVs) attitude by using an Active Disturbance Rejection Control (ADRC) controller. Addressing the challenge of numerous and complex ADRC parameters, optimization algorithms are employed for parameter tuning. This paper draws on the group mechanism of the Ant Colony Optimization (ACO) algorithm and innovatively introduces population search into the Beetle Antennae Search (BAS) algorithm. The refined algorithm is then applied to tune the ADRC parameters, reducing complexity and human intervention while enhancing intelligence and efficiency. The advanced optimization algorithm exhibits an exceptional global optimization capacity, convergence speed, and stability. Ultimately, flight simulation and experimental results suggest that the optimized ADRC controller demonstrates superior control and antidisturbance capabilities.

Keywords: UAV; ADRC; optimization algorithm; parameter tuning; attitude control

MSC: 93C85; 68T40; 93D21

Citation: Li, W.; Yang, F.; Zhong, L.; Wu, H.; Jiang, X.; Chukalin, A.V. Attitude Control of UAVs with Search Optimization and Disturbance Rejection Strategies. *Mathematics* **2023**, *11*, 3794. https://doi.org/10.3390/math11173794

Academic Editor: Davide Astolfi

Received: 30 June 2023
Revised: 10 August 2023
Accepted: 28 August 2023
Published: 4 September 2023
Corrected: 20 June 2024

1. Introduction

Unmanned aerial vehicles (UAVs) encompass a range of aircrafts that operate without an onboard pilot, communicating through wireless remote control devices and controlled by flight controllers or ground control stations that issue flight tasks [1,2]. Among various UAV configurations, this study focuses on quadcopters, which are characterized by their simple construction, small size, flexible flight, fewer flight-condition restrictions, and vertical takeoff and landing capabilities [3].

In recent years, the continuous advancement of computer technology, artificial intelligence, sensor technology, and inherent benefits of quadcopters have led to their widespread use in various fields, including daily photography, map surveying, intelligence reconnaissance, search and rescue missions, and military defense [4–6]. Quadcopters are underactuated systems with six degrees of freedom and four control inputs [7]. Their nonlinear, strongly coupled, and multivariable nature makes flight control complex, with minor issues potentially causing mission failures and serious problems leading to life and property-safety hazards.

Effective control algorithms can swiftly and accurately adjust quadcopter flight attitude, significantly reducing in-flight risks and broadening the range of executable tasks. To capitalize on quadcopter benefits, this paper investigates quadcopter flight control algorithms, introduces a novel antidisturbance control method, and verifies the results through modeling analysis and simulation experiments.

The first flight control algorithm, based on the principle of negative feedback control, is the proportional–integral–derivative (PID) control algorithm [8]. Simple and with excellent control performance, PID control has been widely adopted in distributed control systems, programmable logic controllers, and most single-chip control systems [9,10]. Numerous scholars have developed more complex control algorithms based on PID control. Zhang [11] introduced a cascade PID control by adding an extra level of PID control, reducing interference in both inner and outer loops during UAV flight. Andrade et al. [12] combined cascade PID control with fuzzy control, proposing cascade fuzzy PID control that not only reduces interference but also enhances UAV adaptability. Feng et al. [13] incorporated a variable domain into cascade fuzzy PID control, optimizing fuzzy rules and improving their applicability and accuracy under varying interferences.

As the drone industry rapidly evolves, the demand for increased control accuracy and speed exposes the limitations of PID algorithms, such as their simplicity and lack of adaptability. Scholars have extensively researched and applied various advanced algorithms to drone control, including model predictive control, sliding mode control, adaptive control, and H_∞ control [14,15].

Professor Han addressed the shortcomings of PID control in nonlinear system control by integrating a Tracking Differentiator (TD), Extended State Observer (ESO), and Nonlinear State Error Feedback Control Law (NLSEF) to develop the Active Disturbance Rejection Control (ADRC) algorithm [16]. ADRC systems embody the concept of total disturbance, emphasizing disturbance rejection as the central control issue and modifying the linear combination of PID control laws into a nonlinear combination. By utilizing tracking differentiators to generate tracking and differential signals for input, the impact of input signal jumps on the system is minimized, reducing overshoot [17]. The nonlinear extended state observer, the core component of ADRC, distinguishes it from other control systems through its unique feature—Active Disturbance Rejection. This is achieved by obtaining specific values of total disturbance via the observer and compensating in advance to transform the controlled object into a standard form, thereby eliminating the disturbance impact [18].

Controller parameters are critical factors in realizing controller functions and determining system control quality. Consequently, the complexity of controller-parameter tuning significantly influences the widespread adoption of controllers in engineering applications. Although the ADRC algorithm offers exceptional disturbance rejection performance and adaptability, it involves numerous parameters. A comprehensive second-order ADRC system may contain over a dozen internal parameters requiring tuning, debugging, and optimization, typically necessitating extensive experience and experimental verification. To address this challenge, adaptive control techniques and machine learning methods have been introduced to ADRC parameter adjustment in recent years, such as fuzzy logic control, neural network control, reinforcement learning, and offline parameter tuning algorithms based on intelligent optimization algorithms [19–21].

Numerous approaches, research, and applications of parameter tuning for bioinspired artificial intelligence algorithms have been extensively explored. Huang [22] transformed the PI controller into a linear ADRC system, simplifying its parameters for servo system control applications and achieving remarkable results. Gao [23] proposed an optimization method for parameter tuning in ADRC systems with large time delays based on the Internal Model Control principle. The simulation results demonstrated the effectiveness of this method, providing guidance for parameter tuning. Sun [24] established a reduced-order linear ADRC system by incorporating model information into the ADRC controller and used optimization algorithms for parameter tuning, significantly simplifying the process. Cai [25] proposed a nonlinear ADRC controller-parameter tuning scheme based on the Grey Wolf Optimization algorithm, showing improvements in robustness and efficiency when compared to other optimization algorithms under various degrees of random disturbance. Chiumeo [26] proposed a parameter conversion formula between the PID controller and linear ADRC controller, determining the bandwidth value range by restricting the ADRC bandwidth through the parameter formula. Zheng [27] and colleagues conducted extensive

simulations, deriving an empirical formula for fitting ADRC controller parameters by summarizing and analyzing experimental data, greatly simplifying the parameter-tuning process for a specific type of ADRC system. Huang [28] designed an ADRC controller for torque ripple suppression in Permanent Magnet Synchronous Motors, using Ant Colony Optimization for self-learning iterative parameter optimization. The final simulation results showed that the optimized parameters provided strong robustness and excellent torque ripple suppression. Based on the time-scale concept proposed by Jingqing Han, Plenz introduced a calculation format for the second-order controlled system's time scale [29]. By establishing the relationship between the time scale and the controlled system, the scholar derived rules for controller-parameter conversion, achieving parameter tuning for ADRC systems.

In 2017, Jiang and Li proposed the Beetle Antennae Search (BAS) algorithm [30], a novel bioinspired search algorithm inspired by the foraging behavior of beetles. The BAS algorithm does not require knowledge of the specific form or gradient information of the target function to be optimized. Instead, it only needs to obtain information about a single beetle through "antennae". The BAS algorithm employs diverse search strategies, an elite retention mechanism, and adaptive adjustment of the exploration range, among other optimization strategies, effectively improving the search efficiency and accuracy.

The Ant Colony Optimization (ACO) algorithm, proposed by Dorigo et al. in 1991 at the first European Conference on Artificial Intelligence (ECAL) [31], is a heuristic optimization algorithm developed based on the foraging behavior of ants in nature. This algorithm emulates the behavioral patterns of ants while searching for food to address optimization problems. In ACO, a colony of ants moves randomly within the search space, exchanging information through pheromone secretion [32]. Ants select paths based on pheromone concentrations, which they continuously update throughout their journey. Following multiple iterations, paths with high pheromone concentrations are more likely to be chosen, ultimately leading to the optimal solution.

The BAS and ACO algorithms have the advantages of being easy to implement, having a wide range of applications, and having a high search efficiency. They have been applied in various optimization problems, such as machine learning, image processing, fault diagnosis, and neural network training, achieving favorable results [33,34].

This study innovatively incorporates the ACO group search mechanism into BAS to enhance the algorithm's global optimization capability, convergence speed, and stability. The improved algorithm is then combined with ADRC for controller-parameter tuning. This approach enhances the ADRC algorithm's intelligence and efficiency, reduces the complexity of parameter setting and the degree of human intervention, and improves the performance and robustness of the control system. The effectiveness of the improved algorithm is validated through simulation and flight experiments in comparison with cascaded PID control.

The structure of this paper is organized as follows. Section 2 introduces the basic algorithm and model. Section 3 describes the experimental platform utilized. Section 4 presents the improved algorithm and its performance results. Section 5 conducts simulations and experiments on unmanned aerial vehicle attitude control and analyzes the results. Finally, conclusions and summaries are provided.

2. Algorithm Introduction

2.1. Active Disturbance Rejection Controller

The application of second-order ADRC is very extensive, as most industrial processes are second-order systems or can be transformed into systems consisting of several coupled second-order models. The ADRC algorithm has low model dependency on the controlled object, an excellent disturbance rejection capability, good adaptability and robustness, and a natural decoupling property. After decoupling, the attitude control system of a quadrotor unmanned aerial vehicle can be represented in second-order nonlinear form [35], as shown in (1):

$$\ddot{y} = f(y, \dot{y}, w(t), t) + bu, \tag{1}$$

where y represents the output of the controlled object, u signifies the output of the controller, $w(t)$ denotes the external disturbance, $f(y, \dot{y}, w(t), t)$ determines the total disturbance of the combined external and internal disturbances, and b is the compensation coefficient.

By selecting the state variables $x_1 = y$ and $x_2 = \dot{y}$, Equation (1) can be transformed into a state equation:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f(x_1, x_2, w(t), t) + bu \\ y = x_1 \end{cases} \tag{2}$$

The core of ADRC lies in how to estimate $f(y, \dot{y}, w(t), t)$ in real time and eliminate it, transforming (1) into a linear integrator series in standard form. The corresponding ADRC with a disturbance-tracking compensation capability is as follows:

(1) Establishing a second-order TD:

$$\begin{cases} \dot{v}_1 = v_2 \\ v_2 = fhan(v_1 - v, v, r, h) \end{cases}, \tag{3}$$

where $fhan(\cdot)$ refers to the estimation function in the nonlinear observer used to estimate the disturbance of the system.

(2) Establishing a third-order nonlinear ESO

Let the total disturbance be a new state of the controlled system, and expand (2) to (4):

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 + bu \\ \dot{x}_3 = f'(x_1, x_2, w(t), t) \\ y = x_1 \end{cases}. \tag{4}$$

where x_3 means the state variable that represents the disturbance expanded from the original object. Establishing an ESO for (4):

$$\begin{cases} \varepsilon_1 = z_1 - y \\ \dot{z}_1 = z_2 - \beta_1 \varepsilon_1 \\ \dot{z}_2 = z_3 - \beta_2 fal(\varepsilon_1, 0.5, \delta) + bu \\ \dot{z}_3 = -\beta_3 fal(\varepsilon_1, 0.25, \delta) \end{cases}, \tag{5}$$

where β_1, β_2, and β_3 are a set of parameters. The $fal(\cdot)$ refers to the estimation function in the linear observer used to estimate the system state.

(3) Designing a NLSEF:

$$\begin{cases} e_1 = v_1 - z_1 \\ e_2 = v_2 - z_2 \\ u_0 = fhan(e_1, ce_2, r_1, h_1) \end{cases}, \tag{6}$$

where c, r_1, and h_1 are a set of parameters.

(4) Compensation control variable:

$$u = \frac{u_0 - z_3}{b_0}, \tag{7}$$

where b_0 is a manually selected parameter for the controlled object. As a control variable, u is divided into two parts, where $-z_3/b_0$ is the component that compensates for disturbances and u_0/b_0 is the component that uses nonlinear feedback to control the integrator in series.

2.2. Beetle Antennae Search Algorithm

The BAS algorithm is an effective metaheuristic algorithm, inspired by the foraging behavior of beetles. The simplified model of the beetle is divided into its left and right

antennae, and there is an equal distance between each antenna and the beetle's centroid. The search process of the BAS algorithm encompasses two phases: search behavior and detection-update behavior. In each iteration, the beetle moves in a random direction, and the implementation of this algorithm requires vector normalization of the search direction as shown in (8):

$$\overrightarrow{b} = \frac{rand(j,1)}{||rand(j,1)||}, \tag{8}$$

where $rand(\cdot)$ signifies a random function and j denotes the spatial dimensions.

Initialize the bidirectional search of the beetle using (9):

$$x_r = x_t + d_t \overrightarrow{b}; x_l = x_t - d_t \overrightarrow{b}, \tag{9}$$

where x_r and x_l signify the locations of the two antennae, x_t is the coordinate of the beetles, t denotes the number of iterations, and d_t indicates the distance from the antennae to the centroid of the beetles.

The position of the beetle is updated according to the rule in (10):

$$x_t = x_{t-1} - \delta_t * \overrightarrow{b} * sign(f(x_r) - f(x_l)), \tag{10}$$

where δ_t is the step size of the search and $f(x)$ is used to calculate the fitness value. The $sign(\cdot)$ determines the beetle's search direction. If the fitness value in the right direction is higher, the function returns one, prompting the beetle to continue in that direction. Conversely, if the fitness value in the opposite direction is greater, the beetle moves accordingly. The step size for each movement is determined by δ_t. In traditional algorithms, d_t and δ_t are usually reduced linearly with an iteration number, with a reduction factor of 0.95.

2.3. Ant Colony Optimization Algorithm

In nature, ants cooperate with each other and interact with their environment while foraging for food. This collaboration relies on pheromones, substances that facilitate indirect communication within the ant colony, enabling the discovery of the shortest path between the nest and food source.

During their search for food, ants tend to choose paths randomly, but they can perceive pheromone concentrations on the ground and gravitate toward areas with higher concentrations. As shorter paths entail a reduced round-trip time, more ants traverse these paths per unit of time, resulting in faster pheromone accumulation compared to longer paths. Consequently, when ants arrive at a junction, they can detect the information left by their predecessors and are inclined to select shorter paths. This positive feedback mechanism leads to an increasing number of ants traveling the shortest path between the nest and food. As pheromones on other paths evaporate over time, all ants eventually follow the optimal path. The ACO algorithm process is as follows.

When ants choose a path, use (11) to calculate the probability of each path selection:

$$p_{ij}(t) = \frac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_{k \in J_i} [\tau_{ik}(t)]^{\alpha} \cdot [\eta_{ik}]^{\beta}}, \tag{11}$$

where $p_{ij}(t)$ represents the probability of an ant moving from node i to node j at time t; $\tau_{ij}(t)$ represents the pheromone concentration on path (i, j) at time t; α and β represent the weights of the pheromone concentration and heuristic factors, respectively; η_{ij} represents the visibility coefficient from node i to node j; and J_i represents the set of neighbor nodes of node i.

After the ants choose a path, use (12) to update the pheromone concentration on the path:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t), \tag{12}$$

where $\tau_{ij}(t+1)$ represents the pheromone concentration on path (i,j) at time $t+1$, ρ represents the pheromone evaporation coefficient, m represents the number of ants in the current iteration, and $\Delta\tau_{ij}^{k}(t)$ represents the pheromone increment of the kth ant on path (i,j).

Update the heuristic factor value during the iteration process by using (13):

$$\eta_{ij} = \frac{1}{d_{ij}}, \tag{13}$$

where d_{ij} represents the distance from node i to node j.

3. Design of ADRC Parameter Tuning

This study investigates the application of second-order ADRC in the pitch and roll control of quadcopter drones, as described in Section 2.1, which requires the tuning of over a dozen parameters. For the parameters that are difficult to configure, such as $(\beta_1, \beta_2, \beta_3, c, r_1, h_1)$, we utilize optimization algorithms for parameter tuning. The block diagram is illustrated in Figure 1.

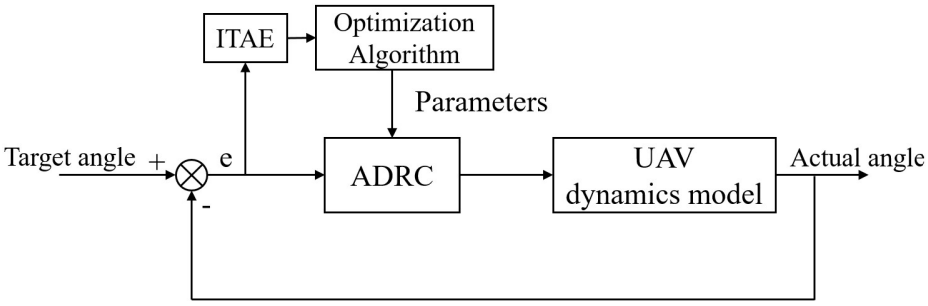
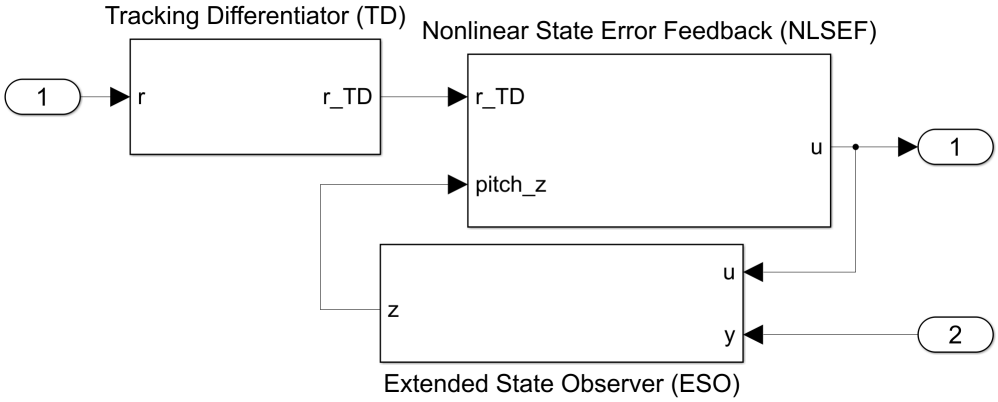


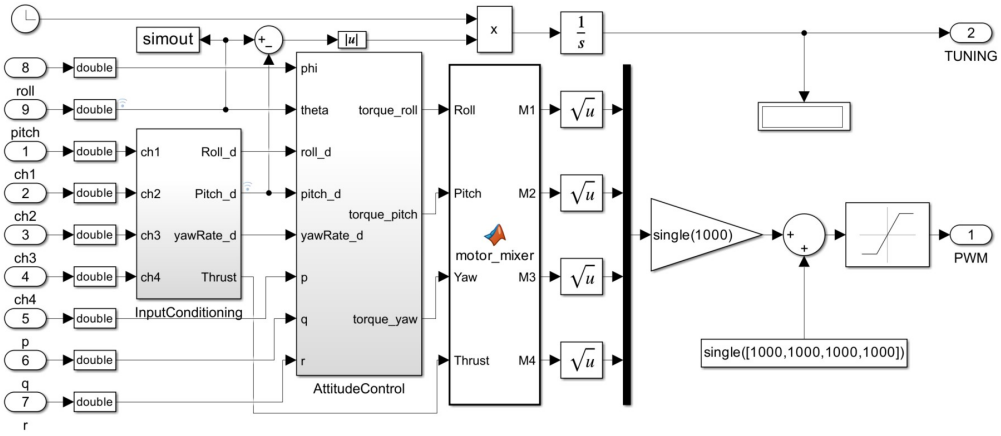
Figure 1. Block diagram of the ADRC parameters' tuning.

To tune the parameters of ADRC and test its effectiveness in UAV attitude control, this study adapts Rflysim [36] to establish a simulation platform in MATLAB/Simulink. The improved platform enables parameter tuning and attitude control simulation and allows the controller to be compiled and downloaded to the Pixhawk hardware in version 2.4.8 and PX4 software in version 1.7.1 environment for flight verification. Figure 2 shows the ADRC controller and parameter tuning for the pitch channel in Simulink. The modified experimental platform is versatile and suitable for the parameter tuning of various controllers.



(**a**) Structure of ADRC in Simulink.

Figure 2. *Cont.*



(**b**) Parameters tuning of pitch channel.

Figure 2. Experimental platform in MATLAB/Simulink.

Subsequent sections evaluate the controller's effectiveness under the current parameters by analyzing the cumulative absolute error magnitude resulting from a step input. Hence, this study utilizes the Integral of Time-weighted Absolute Error (ITAE) as the fitness function for the optimization algorithm, as expressed:

$$ITAE = \int_0^\infty t|e(t)|dt, \tag{14}$$

where t represents the simulation time and $e(t)$ means the error. A lower ITAE value signifies proficient controller performance within the prevailing parameters, whereas a higher value implies diminished performance.

4. Algorithm Improvement

In this section, this study initially implements the group search mechanism in BAS and subsequently assess its performance through ADRC parameter tuning.

4.1. Group Mechanism

The ACO algorithm is a heuristic group-optimization method simulating the optimal path search by ants, whereas the BAS algorithm is a single-agent search algorithm mimicking the foraging behavior of beetles. To upgrade BAS from a single-agent search to a group-optimization algorithm, this study integrates the ACO group search mechanism. Following initialization and group search, we adopt the pheromone update method from ACO, such as employing the inverse distance of the global optimal solution as the pheromone strength and updating the pheromone matrix based on a specific rule (e.g., the evaporation factor). Next, using the updated pheromone matrix, we regenerate each beetle's position, taking inspiration from the path-selection method used by ants in ACO, such as applying the roulette wheel selection method to choose new positions [37].

This research refers to the improved BAS algorithm based on ACO as BAC, and the following Algorithm 1 is an overview of the improved algorithm process.

Algorithm 1 BAC Algorithm Flow

Initialize the number of beetles N, the number of iterations T, the evaporation factor ρ, and the pheromone intensity factor α.
Randomly generate initial positions of N beetles: $X_i(0), i = 1, 2, \ldots, N$.
Initialize the global best solution G^* and the global pheromone matrix P.
for $t = 1, 2, \ldots, T$ **do**
    for $i = 1, 2, \ldots, N$ **do**
        Update position $X_i(t)$ according to BAS algorithm:

$$X_i = X_{i-1} - \delta_t * \overrightarrow{b} * sign(f(X_{i-1}^a) - f(X_{i-1}^b)),$$

where X_{i-1}^a and X_{i-1}^b are the two endpoints of the beetle's antennae and δ_t is the step size of the search.
        Calculate the fitness value $f(X_i(t))$ of the new position.
        Update the local best solution L_i^* for beetle i.
    end for
    Update the global best solution G^*.
    Update the pheromone matrix P:

$$P_{ij} = (1 - \rho)P_{ij} + \alpha \cdot \frac{1}{d(G^*)},$$

where $d(G^*)$ is the fitness value of the global best solution.
    Generate new positions for each beetle i based on the pheromone matrix P, using the roulette wheel selection method, and calculate the selection probability:

$$p_i = \frac{P_{ij}}{\sum_{k=1}^{N} P_{kj}}.$$

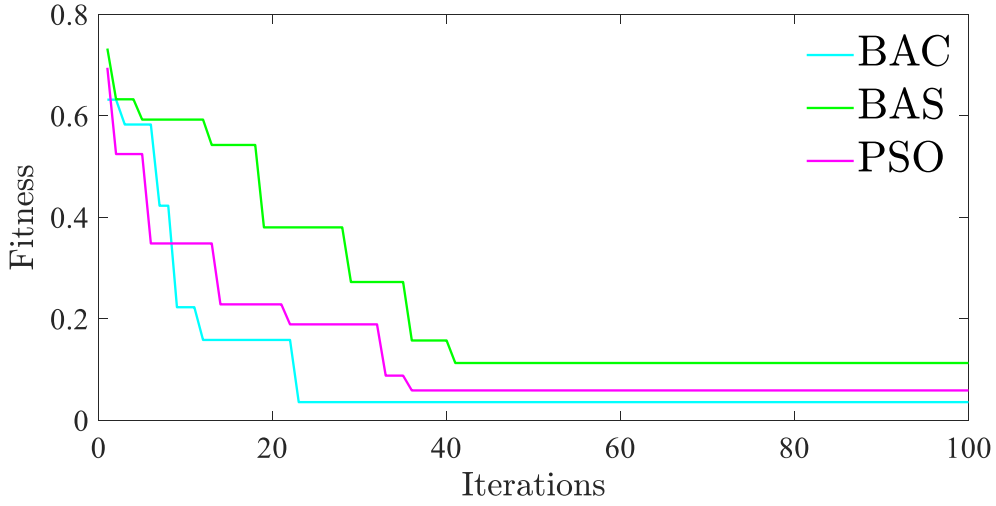
end for
Output the global best solution G^*.

4.2. Algorithm Performance Testing

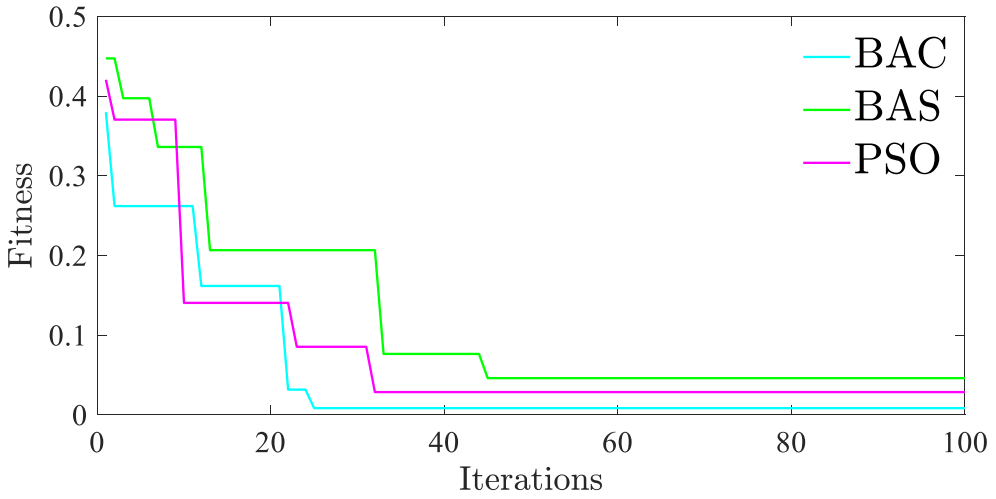
The improved algorithm will be experimentally compared with the standard BAS and Particle Swarm Optimization (PSO) algorithms. We tune the ADRC parameters to test and compare the algorithm performance by using the platform built in Section 3. The initial parameter setting for ADRC are: $r = 17.0$, $h = 2.1$, $\beta_1 = 50.0$, $\beta_2 = 1500.0$, $\beta_3 = 2500.0$, $\delta = 0.01$, $b_0 = 47.4$, $c = 5.0$, $r_1 = 200.0$, and $h_1 = 20.0$. Following the separation principle, we first tune the ESO parameters $(\beta_1, \beta_2, \beta_3)$ followed by the NLSEF parameters (c, r_1, h_1). As the approach and steps are similar, this study will only present and analyze the tuning of the ESO parameters for pitch angle.

Several algorithms are using the same settings. The population size is 30, and the maximum number of iterations is 100. As each adjustment involves three parameters, the dimension is three. The upper and lower limits for the parameters to be optimized are set to (1000, 3000, 4000) and (1, 1, 1), respectively.

The optimal individual fitness curve for tuning the ESO and NLSEF parameters is shown in Figure 3. From Figure 3, BAC reaches a stable fitness value after around 20–30 generations, converging faster than BAS and PSO. Additionally, BAC exhibits significantly lower fitness values compared to the other two algorithms, indicating a higher search accuracy.



(**a**) ESO fitness.



(**b**) NLSEF fitness.

Figure 3. ESO/NLSEF optimal individual fitness curve.

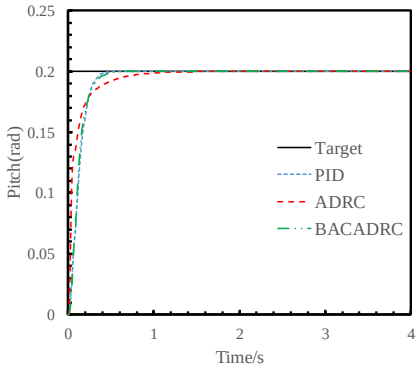
5. Experiment and Analysis

To verify the control effectiveness of ADRC with tuned parameters, this study first conducts a simulation of attitude control and then proceeds with experimental validations.

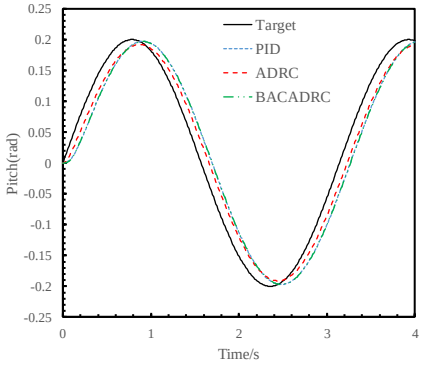
5.1. Simulation and Results Analysis

Based on Section 4.2, this section outlines the ADRC parameters for BAC tuning, which are: $\beta_1 = 2.4$, $\beta_2 = 1243.6$, $\beta_3 = 3570.5$, $c = 4.2$, $r_1 = 294.3$, and $h_1 = 16.1$. The ADRC algorithm utilizing optimized parameters shall be referred to as BACADRC. Cascaded PID and basic ADRC algorithms are selected for a performance comparison with BACADRC. The cascade PID parameters are set as follows: $Kp_{Angle} = 6.5$; $Kp_{AngleRate} = 0.1$; $Ki_{AngleRate} = 0.02$; and $Kd_{AngleRate} = 0.001$. Section 4.2 presents the parameter settings for ADRC.

Based on the parameters established in the controller model, this study performed a drone attitude simulation experiment. Figure 4 illustrates the dynamic response curves of various controllers for the drone's pitch angle channel under a step input of $r(t) = 0.2$ and a time-varying input of $r(t) = 0.2\sin(2t)$ without an internal model and external environmental interference.



(**a**) Step input $r(t) = 0.2$.

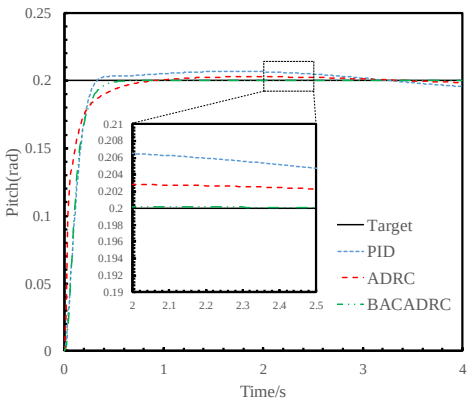


(**b**) Time-varying input $r(t) = 0.2sin(2t)$.

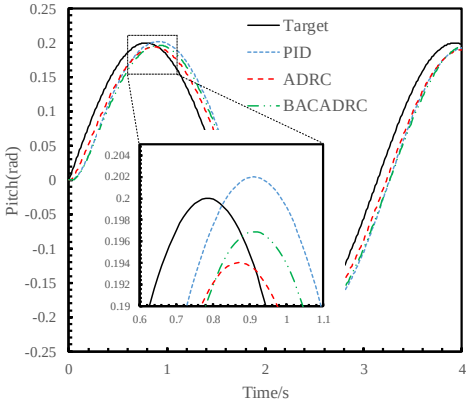
Figure 4. Dynamic response of controllers without interference.

Figure 5 illustrates the dynamic response curves of various controllers for the pitch angle channel under interference caused by a damaged drone model, with a step input of $r(t) = 0.2$ and a time-varying input of $r(t) = 0.2\sin(2t)$. The interference parameter, denoted as $\theta(t)$, is presented in (15):

$$\theta(t) = \begin{bmatrix} 10\sin(t) & 5\sin(4t) \end{bmatrix}^{\mathrm{T}} \tag{15}$$



(**a**) Step input $r(t) = 0.2$.



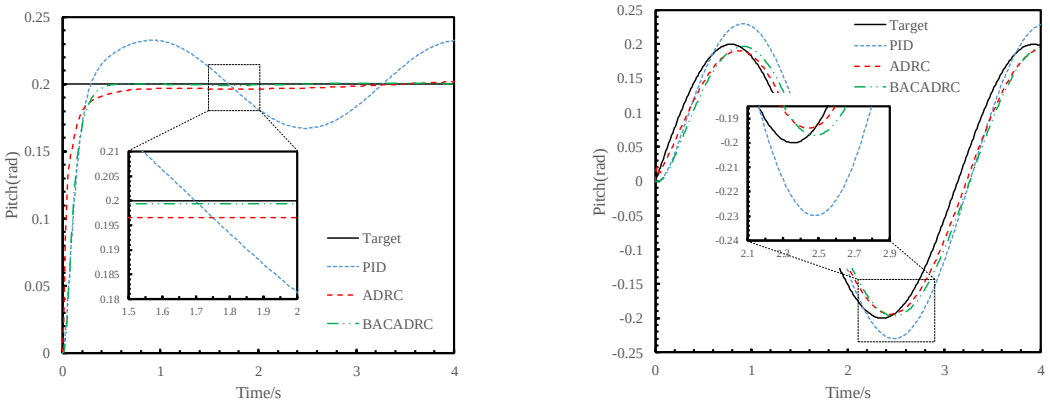
(**b**) Time-varying input $r(t) = 0.2sin(2t)$.

Figure 5. Dynamic response of controllers with internal model interference $\theta(t)$.

Additionally, Figure 6 illustrates the dynamic response curves of various controllers for the pitch angle channel affected by an external disturbance denoted as '$d(t)$', as depicted in (16):

$$d(t) = 10sin(2t) \tag{16}$$

To assess the performance of distinct controllers quantitatively, we conducted numerical analyses of the experimental data derived from Figures 4–6, yielding Tables 1 and 2. A direct numerical comparison was utilized to examine and appraise the performance of BACADRC, PID, and ADRC under various conditions.



(**a**) Step input $r(t) = 0.2$. (**b**) Time-varying input $r(t) = 0.2sin(2t)$.

Figure 6. Dynamic response of controllers with external environmental interference $d(t)$.

Table 1. Numerical performance of different controllers with an input $r(t) = 0.2$.

	Entry	PID	ADRC	BACADRC
No Interference	Rise Time/s	0.19	0.14	0.18
	Steady-State Error/rad	0	0.0001	0.00016
	Deviation ratio/%	0	0.05	0.08
$\theta(t)$ Interference	Maximum Deviation/rad	0.0068	0.0028	0.00021
	Deviation ratio/%	3.4	1.4	0.105
$d(t)$ Interference	Maximum Deviation/rad	0.033	0.0035	0.001
	Deviation ratio/%	16.5	1.75	0.5

Table 2. Numerical performance of different controllers with an input $r(t) = 0.2sin(2t)$.

	Entry	PID	ADRC	BACADRC
No Interference	Peak Deviation/rad	0.00285	0.0089	0.0031
	Deviation ratio/%	1.427	4.45	1.55
	Lag time/s	0.125	0.085	0.125
$\theta(t)$ Interference	Peak Deviation/rad	0.002	0.0066	0.0031
	Deviation ratio/%	1	3.3	1.55
	Lag time/s	0.125	0.085	0.125
$d(t)$ Interference	Peak Deviation/rad	0.03	0.0071	0.0031
	Deviation ratio/%	15	3.546	1.55
	Lag time/s	0.125	0.075	0.125

Upon analyzing Figures 4–6 and Tables 1 and 2, it becomes evident that the control performance of PID, ADRC, and BACADRC in the attitude system of quadrotor UAVs exhibits noticeable differences. These disparities are associated with disturbances within the UAV model, external environmental disruptions, and target signal characteristics.

In the absence of disturbances, a comparison of the response curves in Figure 4 and the rise and lag time values in Tables 1 and 2 reveals that the performance difference between PID and the other two controllers is negligible. Consequently, BACADRC can achieve the rapid response objective in actual UAV control.

Assuming an internal model interference '$\theta(t)$' is present in the pitch channel, a comparison of the deviation ratio values in Figure 5 and Tables 1 and 2 shows that ADRC and BACADRC outperform PID in terms of control performance. This suggests that the disturbance detected by ESO provides feedforward compensation, mitigating interference from the model. A comparison between ADRC and BACADRC indicates that BACADRC's deviation ratio is lower than ADRC's, regardless of whether the input target value is

time varying. Thus, when the UAV model changes, the BACADRC controller, featuring optimized algorithm-tuned parameters, surpasses the conventional ADRC. Examining BACADRC at $r(t) = 0.2sin(2t)$, the deviation ratio is greater under the influence of $\theta(t)$ but remains equal to the ratio without $\theta(t)$ at 1.55%. It can be inferred that the deviation stems from the controller's lag characteristics rather than $\theta(t)$'s influence. This finding implies that BACADRC exhibits comparable performance in suppressing $\theta(t)$'s influence under time-varying input and step input conditions, demonstrating strong robustness.

Assuming that the UAV is subject to an external environmental disturbance '$d(t)$', a comparison of the deviation ratio values in Figure 6 and Tables 1 and 2 indicates that BACADRC's anti-interference performance surpasses that of ADRC and PID, aligning with the scenario affected by the internal disturbance $\theta(t)$. The deviation ratio of BACADRC under time-varying input remains approximately 1.55%, suggesting that BACADRC effectively suppresses external disturbance $d(t)$ irrespective of the presence of time-varying factors in the input, thus showcasing excellent anti-interference capabilities.

In the subsequent step, this study tests the control effectiveness under the influence of external wind. We used the Dryden turbulence model [38] to simulate the wind field. When the drone maintains hovering, the antidisturbance control curves of the controllers for pitch angle are shown in Figure 7.

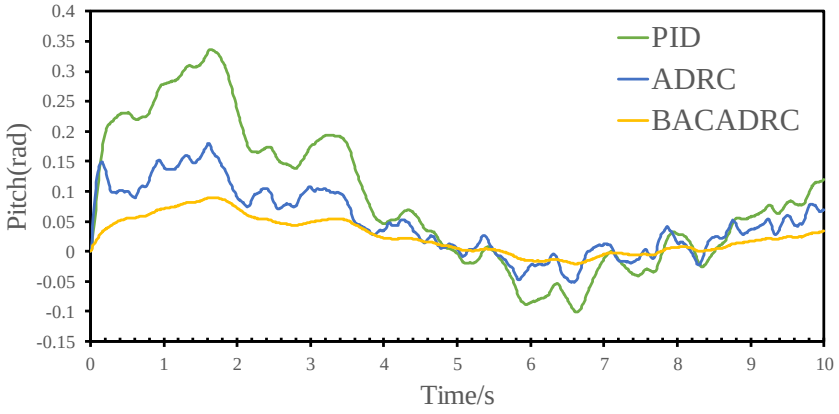


Figure 7. Pitch angle under wind field interference.

According to the simulation experiment data, the numerical analysis in Table 3 is established. It is evident that BACADRC performs the best in suppressing wind disturbances, with the angle error curve being closest to the zero baseline and having the smallest standard deviation.

Table 3. Error statistics of different controllers under wind field interference.

Controllers	Maximum Angle (rad)	Average Value	Standard Deviation
PID	0.3366	0.0800	0.1164
ADRC	0.1799	0.0488	0.0556
BACADRC	0.0895	0.0255	0.0299

Overall, the use of BACADRC can effectively suppress the effects of damaged rotors and wind fields, significantly improving the anti-interference performance of quadcopter UAVs during flight.

5.2. Experiment and Results Analysis

In Section 5.1, this paper tested the excellent control and anti-interference capabilities of BACADRC through simulations. Then, we discretized ADRC, converted it into C++ code using MATLAB, replaced the PID in the PX4 firmware, compiled it, and downloaded it to Pixhawk4 for experimental verification. The equipment is shown in Figure 8a. First,

we conducted a stationary flight experiment, shown in Figure 8b, and then an outdoor actual flight experiment, shown in Figure 8c.

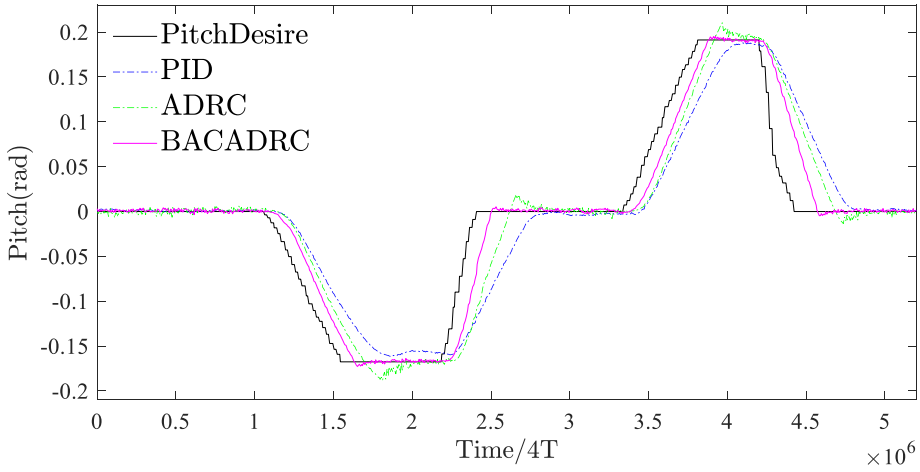
The results of the stationary and actual flight are shown in Figure 9. The two flight modes demonstrate that the drone using BACADRC can track the attitude angle input of the remote control faster and more stably. During actual flight, compared with stationary flight indoors, environmental and internal interference increase, so the performance gap of the controller becomes larger. It can be more clearly seen that BACADRC has better control and a better anti-interference ability.

(**a**) Experimental equipment.

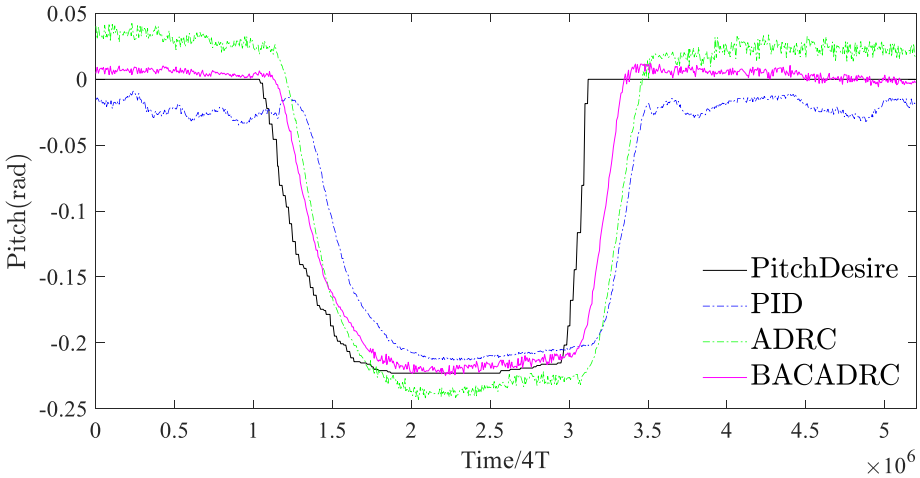
(**b**) Stationary flight.

(**c**) Actual flight.

Figure 8. Attitude control experiment.



(**a**) Stationary flight data.



(**b**) Actual flight data.

Figure 9. Attitude control experimental data.

6. Conclusions

This study introduces a method for rapid and stable quadrotor UAV attitude control by using second-order ADRC and optimization algorithms. Leveraging the population search mechanism of ACO, we incorporate it into BAS. The enhanced BAS algorithm demonstrates robust randomness, an accelerated convergence rate, and high search accuracy. We validate our approach on the modified Rflysim platform for ADRC parameter tuning and attitude control, where BACADRC exhibits exceptional performance in terms of rapid response, stability, and antidisturbance capability.

Our findings are consistent with previous research on quadrotor UAV control methods and extend existing knowledge by integrating an advanced optimization algorithm for ADRC parameter tuning. The proposed method presents a promising alternative to traditional PID control techniques, offering effective and robust quadrotor UAV attitude control.

The implications of this research are far-reaching as the method has potential applicability to various UAV systems and other intricate control tasks. In light of previous studies, our results support the feasibility of utilizing optimization algorithms alongside ADRC to enhance control performance in quadrotor UAVs.

Regarding future research directions, we aim to assess the performance of our proposed method in diverse real-world situations and challenging conditions, such as high wind speeds and rapidly evolving environments. Furthermore, we intend to examine the integration of machine learning techniques to augment the optimization process and the adaptability of the control algorithm, enabling quadrotor UAVs to effectively manage complex tasks and improve their overall performance across a broad array of applications.

Author Contributions: Investigation, W.L., F.Y., L.Z., H.W. and A.V.C.; Supervision, F.Y. and X.J.; Writing—original draft, W.L. and F.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Program of the Department of Natural Resources of Guangdong Province under Grant No. GDNRC[2022]26, the Shandong Province Key R&D Project under Grant No. 2021CXGC011304, the Project of the Natural Science Foundation of Shandong Province under Grant No. ZR2020MF066, and the Youth Innovation Group Project of Shandong University under Grant No. 2020QNQT016.

Data Availability Statement: Data sharing is not applicable.

Acknowledgments: We would like to thank those who have touched our scientific paths.

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

References

1. Telli, K.; Kraa, O.; Himeur, Y.; Ouamane, A.; Boumehraz, M.; Atalla, S.; Mansoor, W. A Comprehensive Review of Recent Research Trends on Unmanned Aerial Vehicles (UAVs). *Systems* **2023**, *11*, 400. [CrossRef]
2. Li, Z.; Zhang, Y.; Wu, H.; Suzuki, S.; Namiki, A.; Wang, W. Design and Application of a UAV Autonomous Inspection System for High-Voltage Power Transmission Lines. *Remote Sens.* **2023**, *15*, 865. [CrossRef]
3. Darvishpoor, S.; Roshanian, J.; Raissi, A.; Hassanalian, M. Configurations, flight mechanisms, and applications of unmanned aerial systems: A review. *Prog. Aerosp. Sci.* **2020**, *121*, 100694. [CrossRef]
4. Ge, W.; Li, X.; Jing, L.; Han, J.; Wang, F. Monitoring canopy-scale autumn leaf phenology at fine-scale using unmanned aerial vehicle (UAV) photography. *Agric. For. Meteorol.* **2023**, *332*, 109372. [CrossRef]
5. Meng, Y.; Xu, J.; He, J.; Tao, S.; Gupta, D.; Moreira, C.; Tiwari, P.; Guo, C. A cluster UAV inspired honeycomb defense system to confront military IoT: A dynamic game approach. *Soft Comput.* **2023**, *27*, 1033–1043. [CrossRef]
6. Bartulović, V.; Trzun, Z.; Hoić, M. Use of Unmanned Aerial Vehicles in Support of Artillery Operations. *Strateg. Znan. Čas. Hrvat. Vojn. Učil. Dr. Franjo Tuđman* **2023**, *7*, 71–92.
7. Viswanathan, S.P.; Sanyal, A.K.; Samiei, E. Integrated guidance and feedback control of underactuated robotics system in SE (3). *J. Intell. Robot. Syst.* **2018**, *89*, 251–263. [CrossRef]
8. Lopez-Sanchez, I.; Moreno-Valenzuela, J. PID control of quadrotor UAVs: A survey. *Annu. Rev. Control* **2023**, *56*, 100900. [CrossRef]
9. El-Shafei, M.A.; El-Hawwary, M.I.; Emara, H.M. Implementation of fractional-order PID controller in an industrial distributed control system. In Proceedings of the 2017 14th International Multi-Conference on Systems, Signals & Devices (SSD), Marrakech, Morocco, 28–31 March 2017; pp. 713–718.

10. Alphonsus, E.R.; Abdullah, M.O. A review on the applications of programmable logic controllers (PLCs). *Renew. Sustain. Energy Rev.* **2016**, *60*, 1185–1205. [CrossRef]
11. Zhang, Y.; Wang, S.; Ji, G. A comprehensive survey on particle swarm optimization algorithm and its applications. *Math. Probl. Eng.* **2015**, *2015*, 931256. [CrossRef]
12. Andrade, F.A.; Guedes, I.P.; Carvalho, G.F.; Zachi, A.R.; Haddad, D.B.; Almeida, L.F.; de Melo, A.G.; Pinto, M.F. Unmanned aerial vehicles motion control with fuzzy tuning of cascaded-pid gains. *Machines* **2021**, *10*, 12. [CrossRef]
13. Feng, Y.; Wu, M.; Chen, X.; Chen, L.; Du, S. A fuzzy PID controller with nonlinear compensation term for mold level of continuous casting process. *Inf. Sci.* **2020**, *539*, 487–503. [CrossRef]
14. Gavilan, F.; Vazquez, R.; Camacho, E.F. An iterative model predictive control algorithm for UAV guidance. *IEEE Trans. Aerosp. Electron. Syst.* **2015**, *51*, 2406–2419. [CrossRef]
15. Azar, A.T.; Serrano, F.E.; Koubaa, A.; Kamal, N.A. Backstepping h-infinity control of unmanned aerial vehicles with time varying disturbances. In Proceedings of the 2020 First International Conference of Smart Systems and Emerging Technologies (SMARTTECH), Riyadh, Saudi Arabia, 3–5 November 2020; pp. 243–248.
16. Han, J. From PID to active disturbance rejection control. *IEEE Trans. Ind. Electron.* **2009**, *56*, 900–906. [CrossRef]
17. Meng, X.; Yu, H.; Zhang, J.; Xu, T.; Wu, H. Liquid level control of four-tank system based on active disturbance rejection technology. *Measurement* **2021**, *175*, 109146. [CrossRef]
18. Liu, S.; Li, D.; Tian, B.; Xue, W.; Sun, L.; Zhu M. A hybrid receding horizon optimization and active disturbance rejection control of boiler superheated steam temperature. In *Process Safety and Environmental Protection*; Elsevier: Amsterdam, The Netherlands, 2023.
19. Dong, Q.; Li, Q. Current control of BLDCM based on fuzzy adaptive ADRC. In Proceedings of the 2009 Ninth International Conference on Hybrid Intelligent Systems, Shenyang, China, 12–14 August 2009; Volume 3, pp. 355–358.
20. Qiao, H.; Meng, H.; Ke, W.; Gao, Q.; Wang, S. Adaptive control of missile attitude based on BP-ADRC. *Aircr. Eng. Aerosp. Technol.* **2020**, *92*, 1475–1481. [CrossRef]
21. Su, Z.Q.; Zhou, M.; Han, F.F.; Zhu, Y.W.; Song, D.L.; Guo, T.T. Attitude control of underwater glider combined reinforcement learning with active disturbance rejection control. *J. Mar. Sci. Technol.* **2019**, *24*, 686–704. [CrossRef]
22. Huang, Y.; Xue, W. Active disturbance rejection control: Methodology and theoretical analysis. *ISA Trans.* **2014**, *53*, 963–976. [CrossRef]
23. Gao, B.; Zheng, L.; Shen, W.; Zhang, W. A Summary of Parameter Tuning of Active Disturbance Rejection Controller. *Recent Adv. Electr. Electron. Eng. Former. Recent Patents Electr. Electron. Eng.* **2023**, *16*, 180–196. [CrossRef]
24. Sun, L.; Jin, Y.; You, F. Active disturbance rejection temperature control of open-cathode proton exchange membrane fuel cell. *Appl. Energy* **2020**, *261*, 114381. [CrossRef]
25. Cai, Z.; Lou, J.; Zhao, J.; Wu, K.; Liu, N.; Wang, Y.X. Quadrotor trajectory tracking and obstacle avoidance by chaotic grey wolf optimization-based active disturbance rejection control. *Mech. Syst. Signal Process.* **2019**, *128*, 636–654. [CrossRef]
26. Chiumeo, R.; Raggini, D.; Veroni, A.; Clerici, A. Comparative Analysis of PI and ADRC Control through CHIL Real Time Simulations of a DC-DC DAB into a Multi-Terminal MVDC/LVDC Distribution Network. *Energies* **2022**, *15*, 7631. [CrossRef]
27. Zheng, Y.; Tao, J.; Sun, Q.; Sun, H.; Sun, M.; Chen, Z. An intelligent course keeping active disturbance rejection controller based on double deep Q-network for towing system of unpowered cylindrical drilling platform. *Int. J. Robust Nonlinear Control* **2021**, *31*, 8463–8480. [CrossRef]
28. Huang, T.; Kang, Y.; Du, S.; Zhang, Q.; Luo, Z.; Tang, Q.; Yang, K. A survey of modeling and control in ball screw feed-drive system. *Int. J. Adv. Manuf. Technol.* **2022**, *121*, 2923–2946. [CrossRef]
29. Plenz, D.; Ribeiro, T.L.; Miller, S.R.; Kells, P.A.; Vakili, A.; Capek, E.L. Self-organized criticality in the brain. *Front. Phys.* **2021**, *9*, 639389. [CrossRef]
30. Jiang, X.; Li, S. BAS: Beetle antennae search algorithm for optimization problems. *arXiv* **2017**, arXiv:1710.10724.
31. Dorigo, M. Ant algorithms solve difficult optimization problems. In Proceedings of the Advances in Artificial Life: 6th European Conference, ECAL 2001 Prague, Czech Republic, 10–14 September 2001; Proceedings; Springer: Berlin/Heidelberg, Germany, 2001; pp. 11–22.
32. Deepa, O.; Senthilkumar, A. Swarm intelligence from natural to artificial systems: Ant colony optimization. *Netw. Graph-Hoc* **2016**, *8*, 9–17.
33. Khan, A.T.; Cao, X.; Li, S.; Hu, B.; Katsikis, V.N. Quantum beetle antennae search: A novel technique for the constrained portfolio optimization problem. *Sci. China Inf. Sci.* **2021**, *64*, 152204. [CrossRef]
34. Shi, H.; Li, W. Artificial neural networks with ant colony optimization for assessing performance of residential buildings. In Proceedings of the 2009 International Conference on Future BioMedical Information Engineering (FBIE), Sanya, China, 13–14 December 2009; pp. 379–382.
35. Mofid, O.; Mobayen, S. Adaptive sliding mode control for finite-time stability of quad-rotor UAVs with parametric uncertainties. *ISA Trans.* **2018**, *72*, 1–14. [CrossRef]

36. Dai, X.; Ke, C.; Quan, Q.; Cai, K.Y. RFlySim: Automatic test platform for UAV autopilot systems with FPGA-based hardware-in-the-loop simulations. *Aerosp. Sci. Technol.* **2021**, *114*, 106727. [CrossRef]
37. Yu, F.; Fu, X.; Li, H.; Dong, G. Improved roulette wheel selection-based genetic algorithm for TSP. In Proceedings of the 2016 International Conference on Network and Information Systems for Computers (ICNISC), Wuhan, China, 15–17 April 2016; pp. 151–154.
38. Gage, S. Creating a unified graphical wind turbulence model from multiple specifications. In Proceedings of the AIAA Modeling and Simulation Technologies Conference and Exhibit, Austin, TX, USA, 11–14 August 2003; p. 5529.