2021 IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC) | 978-1-7281-8535-4/20/$31.00 ©2021 IEEE | DOI: 10.1109/IMCEC51613.2021.9482042

Optimization of ADRC Parameters Based on Particle Swarm Optimization Algorithm

Zhiliang Chen[1] ,Yunfei Li[1] ,Yongli Zhang[1*]
1. Tianjin University of Technology and Education, Tianjin, China
17863205558@163.com, 2816669024@qq.com, zylzhang@126.com
Corresponding Author: Yongli Zhang Email: zylzhang@126.com

***Abstract*—Active disturbance rejection control has multiple advantages such as faster system response, higher accuracy, no excessive requirements on the target model, and suitable for complex and non-linear systems. However, it has the problem of many parameters and difficulty in tuning. A particle swarm optimization algorithm is proposed to optimize the parameters β_1 and β_2 in the nonlinear state error feedback part of the ADRC. This paper compares the simulation effects of PID control, ADRC control and PSO-ADRC control. The simulation results show that ADRC control has better response speed and control accuracy than PID control. Comparing the control results of ADRC before and after optimization, it is found that the optimized controller achieves better control effects.**

Keywords—PID control; ADRC control; Particle Swarm Optimization; Parameter tuning

I. INTRODUCTION

Active disturbance rejection control is a new type of control technology proposed on the basis of traditional PID control. After long-term exploration by the research team, it has been highly recognized in the field of control. Compared with PID, the control quality and control accuracy of the ADRC have been improved, especially when the controlled system is in a harsh environment, its control accuracy can still meet the high standard control requirements. Therefore, when there is an uncertainty disturbance in the control system, the PID controller cannot effectively control it, and the ADRC technology can effectively make up for this shortcoming.

However, the ADRC controller has many control parameters. For a complex Multiple Input Multiple Output (MIMO) system, manual adjustment of ADRC parameters is neither practical nor convenient. Therefore, in order to solve the problem of ADRC control parameter tuning for complex systems, most of the current methods use swarm intelligence algorithms. In literature [1], He Zhihui et al. proposed a Fuzzy-ADRC strategy for the attitude control requirements of plant protection UAV, which can effectively suppress external interference and has strong robustness[1]; In literature [2], Fu Wenqiang et al. proposed a method of optimizing ADRC using BP neural network. The observation accuracy and robustness of the controller have been improved to a certain extent[2]; In the literature [3], Cai Gaipin et al. analyzed the problems of premature and easy to fall into local extremes in the optimization process of PSO algorithm with decreasing inertia weight, and proposed a dynamic inertia weight method to improve the PSO algorithm. And optimize the ADRC parameters, the control result has the advantages of high tracking accuracy and strong anti-interference ability[3].

From theoretical analysis to experimental verification to final practical application, ADRC has a set of mature theories of its own, and has achieved superior results in all major industrial science and technology fields. It not only broadens the development path for my country's intelligent control field, but also provides direction guidance for future intelligent control research.

II. ADRC CONTROLLER DESIGN

The ADRC is composed of tracking differentiator (TD), nonlinear state error feedback (NLSEF) and extended state observer (ESO). The system structure diagram is shown in Fig. 1. The principle of ADRC control strategy can be briefly described as follows: the reference signal v obtains the tracking signal v_1 and the derivative signal v_2 through TD. The ESO estimates the state variable and the unknown disturbance according to the feedback variable y of the controlled system, and calculates it to obtain the error variable. The initial control value can be obtained by nonlinearly weighting the error variable through NLSEF. The external disturbance estimated by ESO is multiplied by the coefficient gain to obtain the disturbance compensation value and the initial control value to obtain the final system control law u . This theory can well solve the actual control problem[4-6].

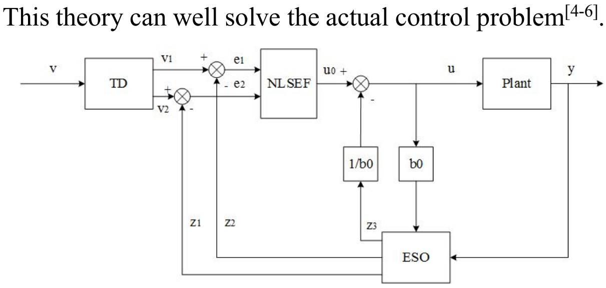


Fig. 1. Structure diagram of ADRC.

(1) Tracking differentiator

The tracking differentiator (TD) is used to transition the set reference signal into a differential signal and a tracking signal. The tracking signal is to ensure that it can fully track the pre-given reference value, and does not affect the subsequent state during the transition process. Its control law has the following equations:

$$\begin{cases} \text{fh} = \text{fhan}\left[x_1(k) - v(k), x_2(k), r, h_0\right] \\ x_1(k+1) = x_1(k) + hx_2(k) \\ x_2(k+1) = x_2(k) + h\text{fh} \end{cases} \tag{1}$$

(2) Expanded state observer

Extended State Observer (ESO) is the core part of ADRC, which observes the total disturbance generated in the system. The observer obtains the state variables from the controlled object, and estimates the feedback state variables of the system in real time through a set of calculation formulas. At the same time, one of the most important functions is to obtain the total unknown external disturbance that acts on the controlled system. Its control law has the following equations:

$$\begin{cases} e(k) = z_1(k) - y(k) \\ z_1(k+1) = z_1(k) + h\left[z_2(k) - \beta_{01}e(k)\right] \\ z_2(k+1) = z_2(k) + h\left[z_3(k) - \beta_{02}\text{fal}\left(e, \frac{1}{2}, \delta\right) + bu\right] \\ z_3(k+1) = z_3(k) - h\beta_{03}\text{fal}\left(e, \frac{1}{4}, \delta\right) \end{cases} \tag{2}$$

(3) Nonlinear State Error Feedback

Nonlinear State Error Feedback (NLSEF) is used to generate control variables. According to all the error variables of the previously acquired system and various nonlinear weights, it is estimated that the external disturbance is compensated to obtain the final control law. Its control law has the following equations:

$$\begin{cases} e_1 = v_1 - z_1 \\ e_2 = v_2 - z_2 \\ u_0 = \beta_1\text{fal}(e_1, a_1, \delta) + \beta_2\text{fal}(e_2, a_2, \delta) \\ u = u_k - \dfrac{z_3}{b_0} \end{cases} \tag{3}$$

In the above formula, the function has the following meaning: $r, \text{h}, \beta_{01}, \beta_{02}, \beta_{03}, \beta_1, \beta_2, \alpha_1, \alpha_2, \delta, \text{b}_0$ are all parameters in the controller. Among them, r is determined by the speed of the transition process and the endurance of the system; h is the sampling step size; $\beta_{01}, \beta_{02}, \beta_{03}$ are the observer parameters, and their values are related to the observer bandwidth; β_1, β_2 are the gain parameters; $\alpha_1, \alpha_2, \delta$ are the parameter factors of the function fal[7]. It can be seen that there are many parameter variables in the ADRC system, which are complicated and difficult to operate in the parameter adjustment process, resulting in poor ADRC adjustment effects. Therefore, this paper optimizes the parameters in ADRC by introducing the particle swarm algorithm.

The ADRC structure diagram built by Simulink is shown in Fig. 2:

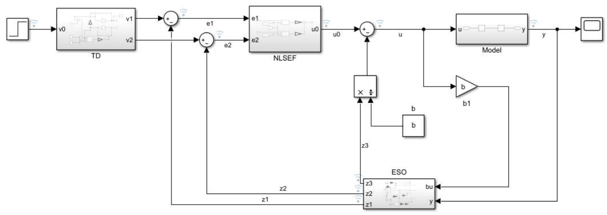


Fig.2. ADRC simulation structure diagram.

III. PARTICLE SWARM ALGORITHM OPTIMIZES THE ACTIVE DISTURBANCE REJECTION CONTROLLER

A. *Particle swarm algorithm*

The particle swarm algorithm is based on imitating the bird population as a prototype, transforming the foraging process into a problem of finding the optimal solution to the problem, and plays an important role in solving a series of system optimization problems. This algorithm is a parallel group intelligent optimization algorithm. Compared with genetic algorithm, it avoids population crossover and mutation operations. The optimization process is simple, the number of parameters is small, and it has the advantages of fast convergence speed and strong global search ability. So it is widely used in complex and high-dimensional problems such as function optimization[8].

First of all, the operator needs to set a certain number of particles and populations, and give each particle certain speed and position information. The fitness function is an important indicator that indicates the state of particle motion. It can be set according to the characteristics of the controller or the response error of the system, and use it as the termination condition of the particle optimization process. The change of particle position and velocity in the motion space is not only related to inertia weight w, learning factors c_1 and c_2, random constants r_1 and r_2, but also closely related to two extreme parameters. One is the individual extreme value Pbest of the particle, which represents the current optimal position of the particle, and the other is the global extreme value Gbest of the population, which represents the current best position in the population. The two extreme parameters indicate the direction for the movement of the particle swarm. The formula (4) and formula (5) are the update formulas for the velocity and position of the particles:

$$V_{id}^{k+1} = wV_{id}^{k} + c_1r_1\left(P_{id}^{k} - X_{id}^{k}\right) + c_2r_2\left(P_{gd}^{k} - X_{id}^{k}\right) \tag{4}$$

$$X_{id}^{k+1} = X_{id}^{k} + V_{id}^{k+1} \tag{5}$$

In the formulas above, $d = 1,2,...,D$; $i = 1,2,...,n$; k is the current number of iterations; $V_i = (V_{i1}, V_{i2}, ...V_{iD})^T$ and $X_i = (X_{i1}, X_{i2}, ...X_{iD})^T$ respectively represent the speed and position of the i-th particle in the D-dimensional space; The individual extreme value is $P_i = (P_{i1}, P_{i2}, ...P_{iD})^T$; The global extreme value is $P_g = (P_{g1}, P_{g2}, ...P_{gD})^T$.

The optimization process of particle swarm algorithm is shown in Fig. 3.

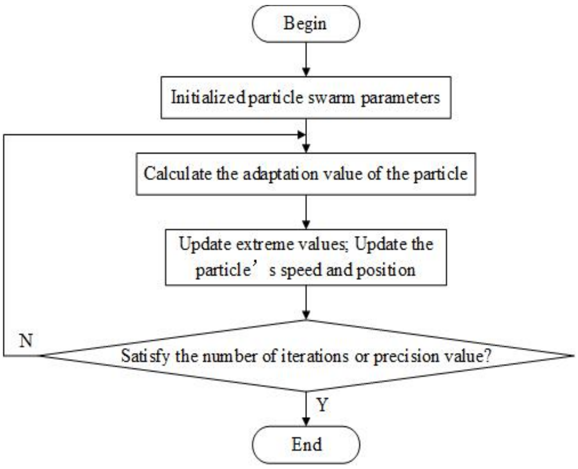


Fig. 3. Optimization flow chart of PSO.

B. *Tuning of ADRC Parameters*

When using the algorithm to optimize the controller, a suitable fitness function should first be determined to evaluate the performance of the controller, which directly affects the effectiveness of the ADRC parameters tuning. In this paper, the integral of the absolute value of the time error of the performance index J is used as the objective function of the particle swarm algorithm[9], and the output error of the control system is selected as the integral term.

$$J = \int_0^\infty t \left| e(t) dt \right| \tag{6}$$

Normally, the smaller the value of the performance index function, the better the parameter optimization effect, that is, the problem of ADRC parameter tuning is transformed into the problem of finding the minimum value of the function.

This paper uses particle swarm optimization to optimize the parameters β_1 and β_2 in NLSEF. The optimization process is as follows:

(1) Initialize the particle swarm parameters;

(2) Set the speed, position and fitness function of the particles;

(3) Assign the particles to the parameters β_1 , β_2 in turn, and call the model to run;

(4) The particle constantly adjusts its position according to the update formula of speed and position;

(5) Determine the individual extreme value and the global extreme value of the particle through comparison;

(6) Until the iteration result reaches the artificially set precision value, the program stops running.

IV. Algorithm Simulation and Result Analysis

In establishing a system model in Simulink, it is necessary to associate the model with performance indicators, select ITAE as the performance indicator function, and establish a simulation model of performance indicators based on the relationship of variables within the indicators, as shown in Fig. 4.

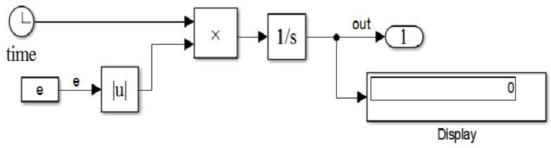


Fig. 4. Simulation model of performance index function.

In this paper, the second-order system is selected as the controlled object to verify the optimization effect of the active disturbance rejection controller using the particle swarm algorithm. The second-order transfer function is the formula (7).

$$G(s) = \frac{1}{s^2} \tag{7}$$

Based on the MATLAB/SIMULINK simulation platform, for the second-order control system, the PSO-ADRC design method is used to optimize the ADRC parameters. In the optimization process, set the number of particle populations to 40, the number of iterations to 200, and the learning factor to be 2. The change curve of the fitness function value is shown in Fig. 5. From the simulation diagram, it can be seen that when the number of iterations reaches about 120, the control accuracy of the system can be stabilized at about 0.5.

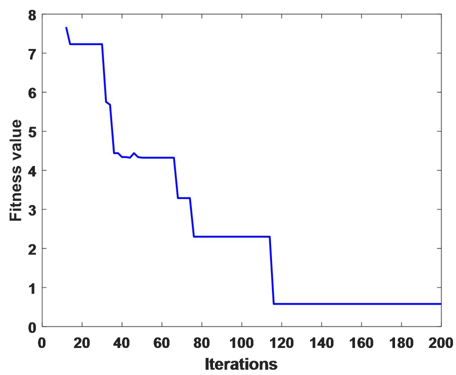
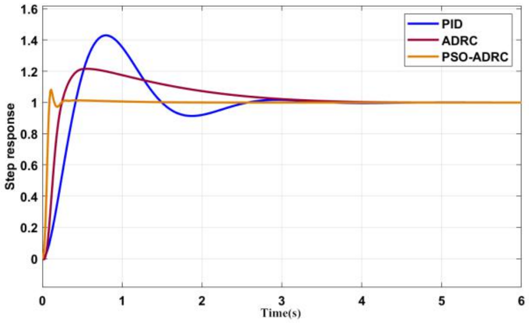


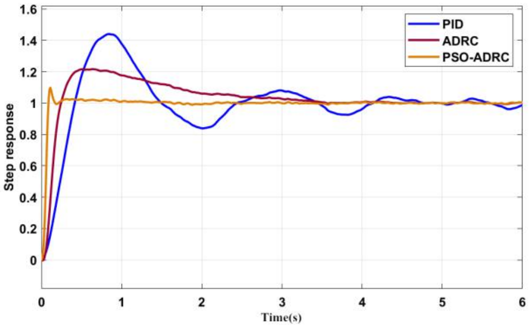
Fig. 5. Fitness function curve.

The simulation control output curve of the step signal of PID control and ADRC control is shown in Fig. 6. The figure (a) is the output curve in an ideal environment, and the figure (b) is the output curve after the system introduces white noise. The parameter values

$\beta_1 = 196, \beta_2 = 98.5$ optimized by the particle swarm algorithm. It can be seen from the simulation image that the effect of ADRC control is obviously better than PID control. Moreover, the effect of ADRC control optimized by PSO has also been significantly improved. The system can reach a stable state in a shorter time, and the response speed is also significantly improved.



(a)



(b)

Fig. 6. System response curve.

V. CONCLUSION

In the actual engineering application, ADRC has the problems of numerous parameters and difficulty in tuning parameters, and it is difficult to achieve accurate and effective control. This paper introduces the particle swarm algorithm for the problem of ADRC tuning difficult, and by using the intelligent optimization characteristics of the algorithm to determine more suitable parameter values, to a certain extent, avoiding the consumption of manpower. The PID control, ADRC control and PSO-ADRC control are simulated and compared, and the control effects of the three are visually analyzed. The experimental results show that the controller optimized by PSO has faster response to the system and better tracking effect. Therefore, to a certain extent, the particle swarm algorithm effectively solves the problem of the difficulty in parameter tuning of ADRC, which has certain research significance.

ACKNOWLEDGMENT

This work was supported by the National Key Research and Development Program of China (2017YFB0403904).

REFERENCES

[1] He Zhihui, Gao Wanlin, He Xiongkui, Wang Minjuan, Song Yue. Tandem plant protection drone attitude control simulation based on auto-disturbance rejection fuzzy parameter optimization [J]. Jiangsu University (Natural Science Edition), 2021, 42 (02): 198-206.

[2] Fu Wenqiang, Zhao Dongbiao, Zhao Shichao. Optimized Active Disturbance Rejection PMSM High-precision Speed Control Based on BP Neural Network [J]. Micro Motor, 2020, 48 (12): 50-54.

[3] Cai Gaipin, Zhou Xiaoyun, Liu Xin. Improved PSO algorithm optimization of electro-hydraulic position servo system auto disturbance rejection tracking control [J/OL]. Mechanical Science and Technology: 1-8 [2021-04-02].

[4] Nie Jialun, Optimal Design of Auto Disturbance Rejection Controller for Linear Synchronous Motor [D]. Shenyang University of Technology, 2020.

[5] Wang Feng, Zhu Huabing, Yu Dongcai, Gao Song. Smart Car Lateral Based on Auto Disturbance Rejection Technology Control [J]. Agricultural Equipment and Vehicle Engineering, 2021, 59 (02): 48-53.

[6] You Xiaoying, Research on Maximum Power Point Tracking of Photovoltaic Power Generation System Based on LADRC under Complex Illumination [D]. Hebei University, 2020.

[7] Tian Huangtian, Xie Yuan, Shi Lingli, Optimization of Variable Pulping ADRC Parameters Based on Improved Difference Algorithm [J]. Computer Simulation, 2020, 3(03): 78-82.

[8] Shi Lu, Lin Qi, ADRC of rope-drawn parallel robot based on particle swarm optimization algorithm [A]. Chinese Society of Automation. Proceedings of China Automation Conference 2020 (CAC2020) [C]. Chinese Society of Automation: China Automation Learn, 2020.

[9] Shi Chenxi, Research on ADRC and Controller Parameter Tuning Method [D]. Jiangnan University, 2008.