A Parameter Optimization Method of ADRC by Adaptive Multi-population Genetic Algorithm

Shimin Yu, Jianjian Wang, Qingyong Zhang
School of Automation , Wuhan University of Technology, Wuhan, 430070, China
E-mail: yushimin123@whut.edu.cn

***Abstract*—In order to solve the problem that the parameters of the auto disturbance rejection controller are hard to be set, this article has presented an improved adaptive multi-population genetic algorithm for parameter tuning of ADRC. Multiple population co-evolution was used to replace the traditional single population evolution, and an adaptive parameter model is applied to the multi-population genetic algorithm. Considering the dynamic performance and control requirements of the controller, an evaluation function is established. The course control of unmanned ship is simulated as an example. The simulation results show that the proposed optimization method in this article can improve the response speed and control accuracy of the ADRC obviously, and it has certain effectiveness and practicability.**

Keywords—ADRC; Optimized genetic algorithm; Parameter tuning; Adaptive adjustment

I. INTRODUCTION

Active disturbance rejection control [1] is a robust control method [2] that does not depend on accurate system models, and is a powerful means to solve the control problem of nonlinear uncertain systems [3]. The state observer and the total disturbance are estimated in real time by the extended state observer, and the corresponding compensation is given. The nonlinear uncertain object is fed back into a linear integral tandem object. At the same time, the special nonlinear effect is used to accelerate the convergence speed and improve the control system [4]. Dynamic performance. At present, the auto disturbance rejection controller has been successfully applied in many fields such as motor control, robot control, and aircraft control [5].

Although the ADRC [6] has strong adaptability and robustness, due to its complicated structure and many parameters, it is difficult to adjust and optimize its parameters, so that the control effect is difficult to achieve a more ideal state. Some scholars [7] have proposed to apply the genetic algorithm to the parameter tuning of the ADRC. Although this can optimize the parameters of the ADRC to a certain extent, it is difficult to get the desiring result as is trapped in the local optimal solution frequently and the search efficiency will be reduced as the population grows larger.

Aiming at the above problems of using genetic algorithm [8] to adjust parameters, a multi-population [9] adaptive genetic algorithm is proposed in this article to replace the traditional genetic algorithm.

In order to solve the problem that the traditional genetic algorithm is easy to fall into the local optimal solution and the model adaptability [10] is poor, multiple populations are introduced simultaneously for optimal search, and different populations produce different parameters to achieve different search purposes. The populations are co-evolved [11] by immigration, and the optimal individuals of each population evolution are retained by artificial selection operators [12] as the basis for the convergence [13] of the algorithm.

In order to settle the problems of traditional genetic algorithm in solving combinatorial optimization problems, such as slow convergence rate and is easy to fall into local optimal solution, an adaptive mechanism [14] is proposed to adjust genetic algorithm to improve genetic algorithm. According to the specific evolutionary state in the evolution process, the control parameters will be adjusted automatically. Thus it can improve the convergence speed and global search ability [15] of the algorithm.

There are some researches on automatic disturbance controllers at home and abroad, but they are all simple design. Without in-depth optimization, the response effect is not very good. In order to further improve its performance, it is necessary to seek a smarter and more effective one. Algorithm to improve it. In this paper, the improved genetic algorithm is applied to the parameter tuning of ADRC to improve its performance. Firstly, the improvement process of genetic algorithm is introduced, and then the genetic algorithm is applied to the optimization design of the auto disturbance rejection controller.

II. ADAPTIVE GENETIC ALGORITHM

The first step to improvement is to introduce an adaptive genetic algorithm (AGA). Traditional genetic algorithms use fixed parameters in the process of population evolution, but the system is often sensitive to the setting of parameters, especially the setting of crossover probability and mutation probability [16]. The value of the crossover probability determines the degree of gene exchange of the population, and also directly affects the diversity of the offspring. The value of the mutation probability determines how many individuals in the population will mutate. Genetic algorithms will degenerate to random search if the value too high, and it is easy to fall into the local optimal solution if too small. For different optimization problems, it has to repeatedly determine the crossover probability and the mutation probability, it is difficult to find the most appropriate value for different problems even so.

Adaptive Genetic Algorithm [17] (AGA) is an optimization algorithm whose crossover probability and mutation probability can automatically change with fitness. When the population tends to be the local optimal solution, the crossover probability and the mutation probability will increase, and when the group fitness is more dispersed, the crossover probability and the mutation probability will

*THE CORRESPONDING AUTHOR IS WANG JIANJIAN.
E-MAIL: 604502198@QQ.COM.
THIS PAPER IS SUPPORTED BY WHUT NATIONAL TRAINING PROGRAM OF INNOVATION AND ENTREPRENEURSHIP FOR UNDERGRADUATES IN 2018(ID: 2018 1049711014)

decrease. The adaptive genetic algorithm ensures the convergence of the genetic algorithm while maintaining the diversity of the population. It also can improves the convergence speed and convergence accuracy.

The adaptive adjustment model of crossover probability and mutation probability is as follows:

$$P_c = \begin{cases} \dfrac{k_1 * (f_{\max} - f)}{f_{\max} - f_{avg}}, & (f \geq f_{avg}) \\ \mathrm{k}_2, & (f < f_{avg}) \end{cases} \tag{1}$$

$$P_m = \begin{cases} \dfrac{k_3 * (f_{\max} - f^{'})}{f_{\max} - f_{avg}}, & (f^{'} \geq f_{avg}) \\ k_4, & (f^{'} < f_{avg}) \end{cases} \tag{2}$$

Where k_1 and k_2 are the mutation probability constants to be set, k_3 and k_4 are the mutation probability constants to be set; P_c and P_m are the adaptively adjusted crossover probability and variation probability; $f_{\max}$ is the maximum fitness in the population; f_{avg} is the population average Fitness; f is the greater fitness of the two individuals to cross; $f^{'}$ is the fitness value of the individual to be mutated.

The steps of the adaptive genetic algorithm are as follows:

Step1: Randomly generate an initial population [18].

Step2: Calculate the fitness value of each individual and judge whether it meets the optimization criteria. If it matches, output the optimal solution and the corresponding best individual, then end the algorithm, otherwise execute Step3.

Step3: According to the fitness value, the individual with high fitness is selected. The probability of being selected is high, and it is likely to be inherited.

Step4: Calculate and determine the current crossover probability according to formula (1) and generate new individuals by crossover.

Step5: Calculate and determine the current mutation probability according to formula (2), and generate new individuals through mutation.

Step6: Generate a new generation of populations from crossovers and mutations, then return to Step2.

TABLE I. THE PROCEDURE OF CROSSOVER AND MUTATION PROCESS

Procedure: Crossover and Mutation
IF f • favg
Pc = k1*(fmax - f)/(fmax - favg)
ELSE Pc == k2
END IF
Generate a random number r1
IF r1 • Pc
Generating new individuals by crossing
IF f' • favg
Pm = k3*(fmax - f')/(fmax - favg)
ELSE Pm == k4
END IF
Generate a random number r2
IF r2 • Pm
Randomly generate a mutation location
Locus mutation
ELSE Inherit father's genes
END IF
END IF

The adaptive genetic algorithm is simulated by a typical multi-extreme function (3), and compared with traditional genetic algorithms.

$$f(x) = x^2 - 10\cos(2\pi x) + 10.0, \quad 0 \leq x \leq 4 \tag{3}$$

The simulation results are as follows:

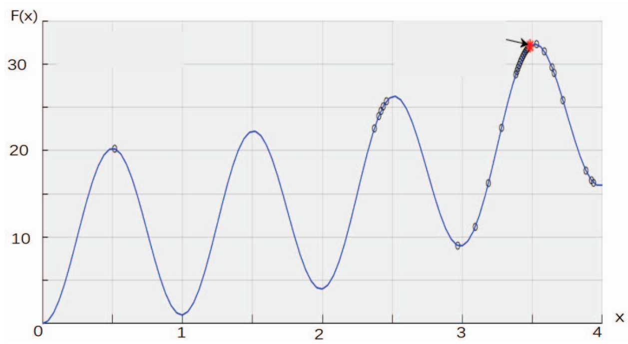

Fig. 1. Traditional genetic algorithm optimization procedure

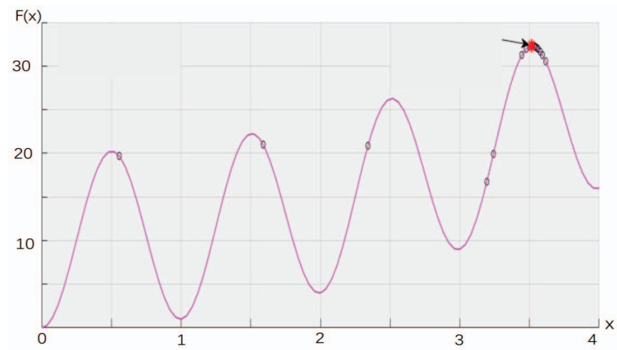

Fig. 2. Adaptive genetic algorithm optimization procedure

In Figures 1 and 2, the curve represents the function to be optimized, the small circle represents the optimization process, and the red solid point represents the final optimization result. The optimal solution is the maximum value of the function over the interval [0 4]. Deriving the original function, you get:

$$f^{'}(x) = 2x + 20\pi\sin(2\pi x), \quad 0 \leq x \leq 4 \tag{4}$$

The optimal solution is x=3.515 and f(x)=32.311 by calculation in matlab.

It can be seen from Fig. 1 and Fig. 2 that the adaptive genetic algorithm is obviously more accurate than the traditional genetic algorithm. In addition, from the perspective of the optimization process, the evolutionary algebra needed for the adaptive genetic algorithm to achieve the optimal solution is obviously less than the traditional legacy algorithm. Therefore, the adaptive genetic algorithm will play an important role in parameter optimization of the ADRC

III. MULTI-POPULATION GENETIC ALGORITHM

The second step of improvement is to replace the single-population genetic algorithm (SGA) with a multi-population genetic algorithm (MPGA). Selection operations in genetic algorithms are based on fitness values of individuals in the current population. When the fitness of an individual in the group is much higher than other individuals, the individual will be selected multiple times, so the next generation population will soon be controlled by the individual, the entire group will lose competition, so that the group can not further evolve. The proper setting of Pc and Pm involves the balance between the global search and the local search ability of the genetic algorithm, and the final result of the evolutionary search is quite sensitive to the values of Pc and Pm. The size of the population also has a great influence on the optimization performance of the genetic algorithm.

Therefore, premature convergence is a phenomenon that cannot be ignored in genetic algorithms, which often leads to the inability to search for global optimal solutions. In view of the above problems, a multi-population genetic algorithm is proposed to replace the conventional single-population genetic algorithm.

Multiple populations are introduced simultaneously for optimal search, and different populations can be assigned different parameters to achieve different search purposes. Through the immigration operation between the various populations, the co-evolution between the populations is realized, and the acquisition of the optimal solution is the comprehensive result of the co-evolution of multiple populations. Then through the manual selection operation, the optimal solution in each evolutionary generation of each population is saved and used as the basis for the convergence of the judgment algorithm.

The selection operation of the population is roulette selection [19], the cross operation is a single point crossing, and the mutation operation is a site variation [20].

In the genetic algorithm, the crossover operator determines the global search ability of the genetic algorithm. The mutation operator determines the local search ability of the genetic algorithm, but the value of Pc and Pm is innumerable. For different choices, the optimization result is also different. Multi-population genetic algorithm (MPGA) makes up for this deficiency of simple genetic algorithm (SGA) by co-evolving with populations without control parameters, taking into account the global search and local search of the algorithm. The different groups exchange information periodically through the immigration operator. The specific exchange method is to replace the worst individual in the target population with the optimal individual of the source population. The elite population is selected from other populations by artificial selection operators. The elite populations are not selected, crossed and mutated to ensure that the optimal individuals produced in each population will not be destroyed and lost during the evolution process.

The procedure of immigration operation and manual selection operation is as follows:

TABLE II. THE PROCEDURE OF IMMIGRANT OPERATOR

Procedure: Immigrant operator
MP = length(Chrom)
FOR i = 1 to MP
[MaxO,maxI] = max(ObjV{i})
next_i = i + 1
IF next_i > MP
next_i = mod(next_i,MP)
END IF
[MinO,minI] = min(ObjV{next_i})
chrom{next_i} (minI, :) = Chrom{i}(maxI, :)
ObjV{next_i}(minI) = ObjV{i}(maxI)
END FOR

TABLE III. THE PROCEDURE OF ARTIFICIAL SELECTION OPERATOR

Procedure: Artificial selection operator
MP = length(Chrom)
FOR i = 1 to MP
[MaxO,maxI] = max(ObjV{i})
IF MaxO > MaxObjV(i)
MaxObjV = MaxO
MaxChrom(i, :) = Chrom{i}(maxI, :)
END IF
END FOR

Through a multipolar binary function:

$$\max f(x,y) = 21.5 + x\sin(4\pi x) + y\sin(20\pi y) \qquad (5)$$

to find the best value to verify the performance of the multi-population genetic algorithm. Where x is -3.0 to 12.1, y is 4.1 to 5.8.

There are multiple extreme points in the function within the specified interval, which is easy to fall into the local optimal solution for general genetic algorithms.

The simulation results are as follows:

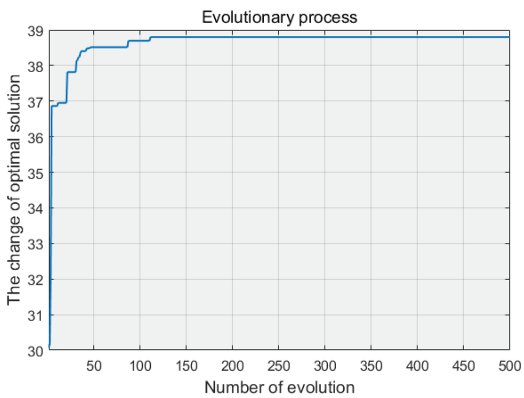

Fig. 3. Evolutionary with SGA

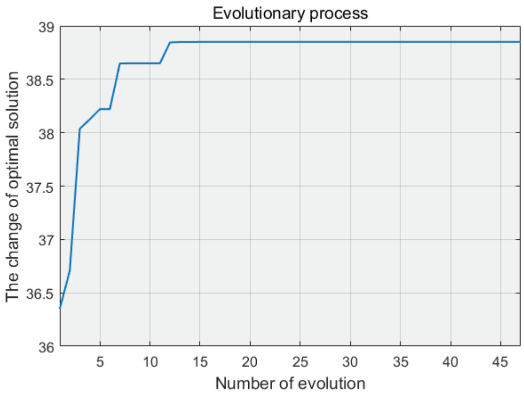

Fig. 4. Evolutionary with MPGA

The optimal value obtained by SGA is 38.7454, and the optimal solution obtained by MPGA is 38.8503. Obviously, the final solution obtained by multi-group genetic algorithm optimization is more ideal. In addition, it can be seen that the SGA algorithm finds the optimal solution when the population evolves to more than one hundred generations, and the MPGA finds the optimal solution when the population evolves to only twelve generations, and the operating efficiency is ten times that of the SGA.

In summary, it can be seen that the MPGA has a good advantage in both optimization effect and optimization efficiency. Applying the MPGA to the optimization design of the ADRC will inevitably lead to better results.

IV. Optimal Design of ADRC

A. Implementation of Multi-population Adaptive Genetic Algorithm

Although the MPGA has been significantly improved in the optimization effect, we can found that the control parameters of each population are different in the MPGA also. Otherwise, when there are only two populations, the control parameters are taken into two ways, and the optimization effect will be closer to the optimization effect of the traditional genetic algorithm. Therefore, in essence, the MPGA does not completely solve the problem that there are countless kinds of control parameters leading to countless kinds of results.

Therefore, this paper combines AGA with MPGA to solve the problem by replacing the fixed control parameters, so that control parameters can change within a certain range.

The improved model is as follows:

$$P_c = \begin{cases} \dfrac{i * k_1 * (f_{\max} - f)}{N * (f_{\max} - f_{avg})}, & (f \geq f_{avg}) \\ \dfrac{i * k_2}{N}, & (f < f_{avg}) \end{cases} \tag{6}$$

$$P_m = \begin{cases} \dfrac{i * k_3 * (f_{\max} - f')}{N * (f_{\max} - f_{avg})}, & (f' \geq f_{avg}) \\ \dfrac{i * k_4}{N}, & (f' < f_{avg}) \end{cases} \tag{7}$$

Where N is the number of populations; i is the ith population, i = 1, 2...N.

In this way, the crossover probability and the mutation probability are divided into a number of non-repetitive intervals according to the number of populations. This ensures the adequacy and effectiveness of the control parameters.

B. Implementation of basic auto disturbance rejection controller

The auto disturbance rejection controller [21] consists of three parts: the tracking differentiator, the extended state observer, and the nonlinear state error feedback. The tracking differentiator TD can track the input signal quickly and without overshoot, and give a better differential signal; the extended state observer ESO is used to estimate the state and expansion state of the system in real time, and feedback the nonlinear uncertain object into Linear integral series object; TD output and ESO output error get system state variable error, adopting appropriate nonlinear combination, realize nonlinear state error feedback control law, plus disturbance compensation, and finally form controlled object The amount of control. The block diagram of the n-order ADRC is as follows:

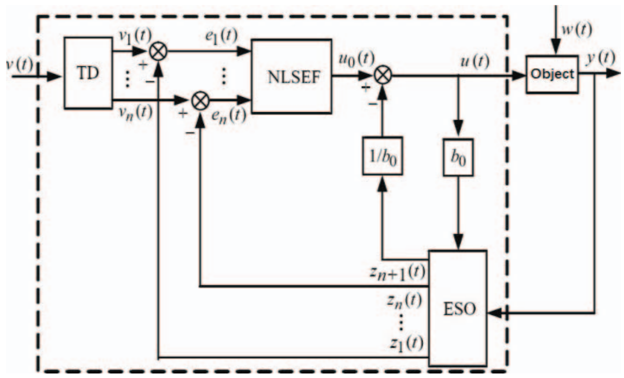

Fig. 5. Structure of ADRC

C. Optimal design of auto disturbance rejection controller

The auto disturbance rejection controller needs more tuning parameters, but from the structural point of view, each part can independently implement the corresponding function, so the problem can be simplified to respectively adjust the parameters of each part. In order to facilitate the implementation of the algorithm, some parameters are often set according to the empirical value. Taking the parameters of the second-order ADRC controller as an example, Usually taken in ESO alph1=1, alph2=0.5, alph3=0.25, In nonlinear feedback alph01=1or0.75, alph02=1or1.5, Nonlinear feedback parameter betal1 、betal2 、betal01 and b because it is not convenient to set by experience, it is adjusted by genetic algorithm.

The implementation steps are as follows:

Step1: Encoding of each parameter. Floating-point encoding is adopted.

Step2: Selection of the initial population. Set a set of parameters using the empirical setting method. The tracking differential parameter r is set according to the response speed of the object, and the parameters of the extended state observer are adjusted by the dynamic parameter setting method. The coefficient of the nonlinear error state feedback is controlled by the PD controller. The control effect of this group of parameters does not need to be too good, as long as the following formula can be satisfied [22].

$$u < U_{\max}, \quad u > U_{\min} and \left|e\right| < E \tag{8}$$

Where: $U_{\max}$ and $U_{\min}$ are the upper and lower limits of the penalty of the control quantity, E is the range allowed by the error. A wide range of individuals satisfying the above formula are searched around the set of parameters until the number of individuals searched is equal to the size of the population in the genetic algorithm. This improves the efficiency of ADRC parameter tuning.

Step3: Individual choice. The individuals are sorted according to the method of roulette after sorting the fitness values, and the selected individuals are used as parents to generate the next generation of individuals.

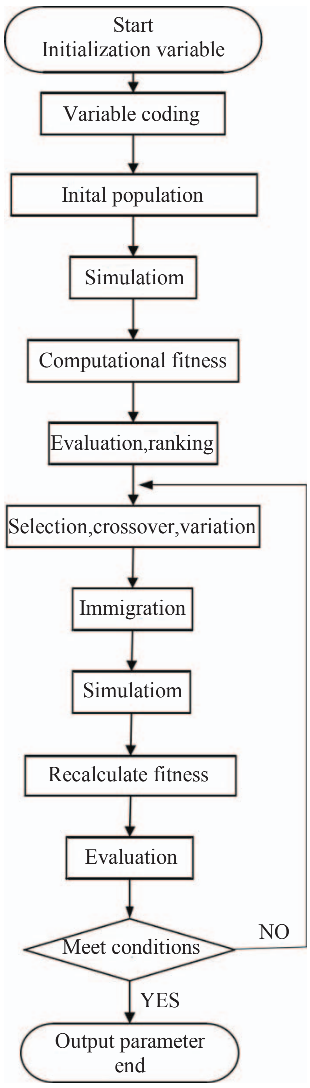

Fig. 6. Process of parameter tuning of an ADRC based on improved genetic algorithm

Step4: Adjust the control parameters. According to the calculated individual fitness value, the improved parameter calculation formula is used to obtain the crossover probability and mutation probability of the population.

Step5: Cross and variation. The crossover is still used as a single point crossover, and the variation uses site variation.

Step6: Chaos immigration. In order to avoid inbreeding and converge to the local optimal solution, an immigration operator is constructed in the process of genetic evolution, and an elite population is introduced outside the population to replace the bad individuals in the tube and participate in the mating of the population. Reproduction to ensure the quality of the population and prevent genetic diseases and decline caused by inbreeding. The chaotic optimization algorithm is used to produce the immigrant group, and the global optimization ability of the ADRC parameters is improved.

Step7: Terminate the condition judgment. When tuning the parameters of the auto-disturbance controller, selecting the maximum genetic algebra and the average fitness can not improve algebra as the termination condition of the genetic algorithm.

The parameter tuning process of the ADRC using adaptive multi-population genetic algorithm is shown in Fig. 6.

V. SIMULATION AND RESULT ANALYSIS

Taking the automatic rudder design of the unmanned boat as an example, the effect of the improved auto disturbance rejection controller is verified. During the unmanned boat driving, the variables that need to be controlled are speed, heading and rudder angle, and the interference is mainly divided into two parts, one from the outside wind, waves and flow; the other is from the parameters perturbation inside the controller. Take the course angle as control object, and the wind, wave and current are used as external disturbances to simulate and verify the improved genetic algorithm.

A random function is now used to simulate the external disturbance and treat it as an equivalent course angle. For simulation analysis, a controller using only the basic genetic algorithm was used as a comparison. In order to better simulate the situation of the unmanned boat during the actual navigation, use a function to simulate the change of direction.

The simulation results are as follows:

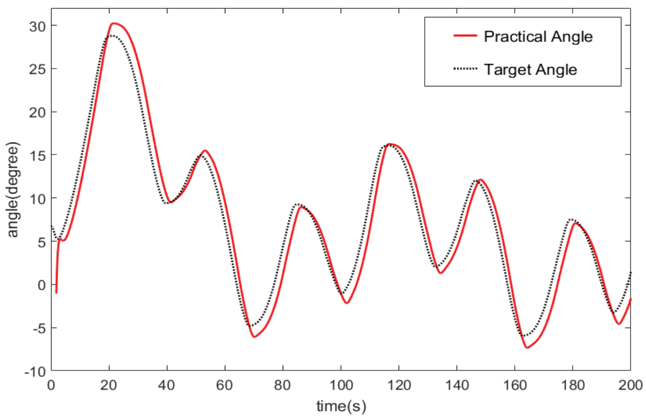

Fig. 7. Angle control with tradition genetic algorithm

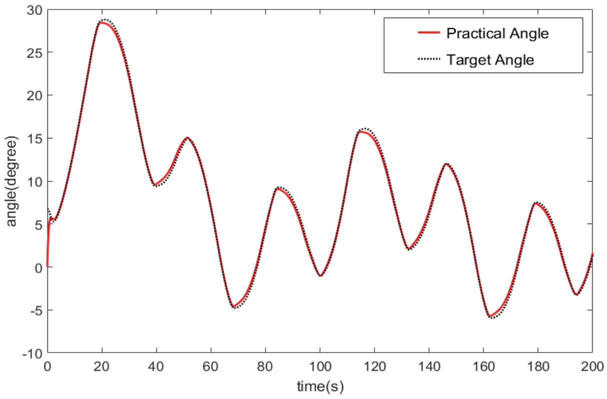

Fig. 8. Angle control with improved genetic algorithm

In the figure, the black dotted line is the target angle, and the red solid line is the actual angle. It can be seen from the figure that in the use of un-optimized genetic algorithm for parameter tuning, the actual output and input can't matched well, there is a significant lag phenomenon [23]. On the other hand, when the direction of navigation changes, there is an overshoot in the output. With the improved genetic algorithm for parameter tuning, we can see that the output curve can follow the target curve well, and there is basically no delay phenomenon [24].

Therefore, compared with the traditional genetic algorithm, the improved genetic algorithm greatly improves the efficiency of parameter tuning in the design of the ADRC, and improving the sensitivity of the ADRC. At the same time, the control accuracy is also significantly improved

VI. CONCLUSION

Aiming at the shortcomings of parameter adjustment in ADRC, an improved genetic algorithm for parameter tuning of ADRC was proposed. By using a variety of groups instead of a single population, the problem that the population is prone to premature ripening is basically solved. At the same time, the adaptive adjustment model is introduced into the genetic algorithm, which improves the global search ability of the genetic algorithm and greatly improves the optimization efficiency of the genetic algorithm. Considering the dynamic performance of the controller, the controller effect evaluation function is established. Under the same control performance, the required control energy of the ADRC is significantly smaller than that of the ADRC designed by other methods.

The simulation[25] results show that the proposed multi-group adaptive genetic algorithm has certain validity and practicability for parameter tuning of the ADRC.

REFERENCES

[1] Sun D. Comments on Active Disturbance Rejection Control[J]. IEEE Transactions on Industrial Electronics, 2007, 54(6):3428-3429.

[2] Safonov M G, Chiang R Y. Model Reduction for Robust Control: A Schur Relative-Error Method[J]. International Journal of Adaptive Control & Signal Processing, 2010, 2(4):259-272.

[3] Krstic M, Modestino J W, Massey J L, et al. Stabilization of Nonlinear Uncertain Systems[M]// Stabilization of nonlinear uncertain systems. Springer, 1998:169-175.

[4] Withers C S, Nadarajah S. Series solutions to linear integral equations[J]. Applied Mathematics & Computation, 2014, 230(3):218-230.

[5] Sayed K, Abdel-Salam M. Dynamic performance of wind turbine conversion system using PMSG-based wind simulator[J]. Electrical Engineering, 2017:1-9.

[6] Xia Y Q, Huang H P, Han J Q. ADRC control of uncertain systems with time-delay[J]. 2003.

[7] Tipsuwan Y, Chow M Y Control methodologies in networked control systems [J]. Control Engineering Practice, 2003, 11(10): 10991111.

[8] Madraswala H S, Deshpande A S. Genetic algorithm solution to unit commitment problem[C]// International Conference on Nascent Technologies in Engineering. IEEE, 2017:1-6.

[9] Cochran J K, Horng S M, Fowler J W. A multi-population genetic algorithm to solve multi-objective scheduling problems for parallel machines[J]. Computers & Operations Research, 2003, 30(7):1087-1102.

[10] Jiang G, Manping L. Class Environments and Students' Adaptability:A Hierarchical Linear Model[J]. Psychological Science, 2005.

[11] Albertweissenberger C, Cazalet C, Buchrieser C. Legionella pneumophila - a human pathogen that co-evolved with fresh water protozoa.[J]. Cellular & Molecular Life Sciences, 2007, 64(4):432.

[12] Lin F, Yang Q W. Improved genetic operator for genetic algorithm[J]. Journal of Zhejiang University-Science A(Applied Physics & Engineering), 2002, 3(4):431-434.

[13] Ethier S N, Kurtz T. Characterization and Convergence[J]. Markov Processes Wiley, 1986.

[14] Fischer A G, Ullsperger M. Real and fictive outcomes are processed differently but converge on a common adaptive mechanism[J]. Neuron, 2013, 79(6):1243.

[15] Ho S L, Yang S, Ni G, et al. A particle swarm optimization method with enhanced global search ability for design optimizations of electromagnetic devices[J]. IEEE Transactions on Magnetics, 2006, 42(4):1107-1110.

[16] Fang Y. Crossover probability estimation using mean-intrinsic-LLR of LDPC syndrome[J]. IEEE Communications Letters, 2009, 13(9):9-50.

[17] Sheng L, Yan-Yan L I. Parameter selection algorithm for support vector machines based on adaptive genetic algorithm[J]. Journal of Harbin Engineering University, 2007, 28(4):398-402.

[18] An V, Dalo, Lu A, et al. An improved genetic algorithm with initial population strategy and self-adaptive member grouping[J]. Computers & Structures, 2008, 86(11):1204-1218.

[19] Fu J, Lei L, Zhou G. A parallel Ant Colony Optimization algorithm with GPU-acceleration based on All-In-Roulette selection[C]// Third International Workshop on Advanced Computational Intelligence. IEEE, 2010:260-264.

[20] Vestã, L G I, Fischer C, Hã, Ibã, O, et al. Between- and within-site variation of density and bending properties of Picea abies structural timber from Norway[J]. Scandinavian Journal of Forest Research, 2016, 31(8):758-765.

[21] Chen wenwen. Simulation research on the automatic disturbance rejection controller of ship course nonlinear system [D]. Shandong university, 2008.

[22] kang zhongjian, wang qingwei, huang lei, et al. Parameter optimization of adrc based on improved genetic algorithm [J]. Information and control, 2008, 37(5):588-592.

[23] wang yudi. Design of automatic course rudder of surface unmanned ship based on self-anti-disturbance control algorithm [D]. Dalian maritime university, 2014.

[24] tan yue, tan guanzheng, ye yong, et al. Dual population genetic algorithm with chaotic local search strategy [J]. Computer application research, 2011, 28(2):469-471.

[25] jiang ping, hao jingyu, zong xiaoping, et al. Simulink modeling and simulation of self-interference rejection controller [J]. Automation technology and application, 2010, 29(2):1-4.