International Journal of Hybrid Information Technology
Vol.8, No.7 (2015), pp.411-422
http://dx.doi.org/10.14257/ijhit.2015.8.7.37

Optimization Design Based On Self-Adapted Ant Colony and Genetic Mix Algorithm for Parameters of PID Controller

Wang Xiao-Yu

College of Engineering/Xi'an International University, Xi'an, China
College of Mechanical and Electrical Engineering/ Xi'an University of Architecture and Technology, Xian, China
lshtttt@126.com

Abstract

This paper presents a method of optimized PID parameter self-adapted ant colony algorithm with aberrance gene, based on ant colony algorithm. This method overcomes genetic algorithm's defects of repeated iteration, slower solving efficiency, ordinary ant colony algorithm's defects of slow convergence speed, easy to get stagnate, and low ability of full search. For a given system, the results of simulation experiments which compare to the result of Z-N optimization and evolution of genetic algorithm optimization and evolution of ant colony system optimization, it has more excellent performance in finding best solution and convergence, the PID parameters also have optimality, system possesses dynamic controlling and performance. The experiments show that this method has its practical value on controlling other objection and process.

Keywords: *PID controller, ant colony algorithm, genetic algorithm, fitness function, ACS-GA algorithm*

1. Introduction

PID (Proportional-Integral-Differential Controller is currently the irreplaceable technology used in industrial process control and motion control owning to its advantage of simple algorithm, good robustness and convenient tuning [1].The most commonly used technique of tuning PID parameter is Z-N principle [2], whereas others are exits. Ziegler-Nichols technique is easy to implement but parameters need fine tuning. The optimization method based on neural networks [3], fuzzy logic [4], genetic algorithm [5] and ant colony algorithm [6] are also developed. The optimization of PID parameters are improved over classical Z-N method, but the issues of premature convergence, stagnation, the global search ability is not high are still outstanding. Reviewing the various situations, we proposed the optimization of PID parameter based on ant colony and genetic mix algorithm in the paper. first, the PID parameter is be calculated by genetic algorithm, and then running the improved ant colony algorithm, by auto adjusting the route choice probability and pheromone updating rule, finally to search out the optimal value of the PID parameter.

2. Ant Colony System and Ant Algorithms

2.1 Ant Colony System

Ant Algorithms was first initiated by Italia academician Coloni A, Dorigo M and Maniezzo V in 1992 as new simulated evolutionary algorithm in order to address some challenge issues in the area of discrete system optimization. The algorithm is distributed computation with feed forward and greedy search functions, and successfully accepted in many applications, such as system control, pattern identification, multi-target optimization and process planning [7].

Ant Algorithm was initially introduced to satisfy typical requirement in composition optimization (Traveling Salesman Problem, TSP) [8] over n cities. The TSP question could be explained by Ant Algorithm which was originally introduced by Dorigo M. The question requires one way shortest path over n cities based on known distance between every two cities. If the quantity of ants is m, d_{ij} is the distance between i and j cities, η_{ij} is the visibility of side (i,j). τ_{ij} is the pheromone trail strength of side (i, j). P_{ij}^k is the transition probability of ant k. αis the accumulated information over movement. ($\alpha \geq 0$, if $\alpha = 0$, then traditional the random greedy algorithm), β is relative significance of visibilities. ($\beta \geq 0$, if $\beta = 0$, then feed-forward process) , $allowed_k = \{n - tabu_k\}$ stands for allowed next city to select for ant k. then transition probability $P_{ij}^{\ k}(t)$ for ant K at time t would be:

$$P_{ij}^k(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}{\sum_{k \in allowed_k} [\tau_{ik}(t)]^\alpha \cdot [\eta_{ik}]^\beta}, & if \quad j \in allowed_k \\ 0, & else \end{cases} \tag{1}$$

At time N, the ant will complete one round of travel; the information at each travel path would be altered as follows:

$$\Delta\tau_{ij}^k(t,t+1) = \begin{cases} Q/L_k & if\ the \quad Kth \quad ant\,use \quad edge(i,j) \\ 0 & othersie \end{cases} \tag{2}$$

$$\tau_{ij}(t+n) = \rho_1 □\tau_{ij}(t) + □\tau_{ij}(t,t+n) \tag{3}$$

$$\Delta\tau_{ij}(t,t+n) = \sum_{k=1}^{m} □\tau_{ij}^{\ k}(t,t+n) \tag{4}$$

Where $\Delta\tau_{ij}(t,t+1)$ is the increase of information over path (i,j) in the travel round, $□\tau_{ij}^{\ k}(t,t+1)$ is the information for ant k at time (t,t+1) over path (i,j). L_k is the path length of ant K traveled in the round. $\Delta\tau_{ij}^k(t,t+n)$ is path of new ant traveled in the round. $(t,t+n)$ is the steps n for ant K over one round. ρ is the information vitality $(0 \leq \rho < 1), 1-\rho$ is the information volatility coefficient. ； Q is the trace numbers of ants.

The current research for ant colony algorithm is mainly to improve the convergence speed of the algorithm through experiments and stagnation, most of it is for improve the pheromone concentration. For example, the next city is be selected by the ant Algorithm rule of shifting the state of ant colony in reference [6]. The premature convergence is avoided caused by local optimization of algorithm by defining the up and lower limits of information density on the path by using the MAX/MIN ant system in reference [9]. The genetic algorithm with genetic factor for population is applied to the optimization parameters, the information density is changed according the aim function, It overcomes the stagnant phenomenon in optimization the PID parameters in reference [8]. The genetic algorithm is be using for generating the initial population and information density at the beginning, and then the ant algorithm is be using for optimization the PID parameters as its feature such as parallel, positive feedback and high efficiency in reference [10].

2.2 Ant Algorithm

Based on the above parameter definition, the solution procedure of TPS for Ant algorithm is as follows:

Setting t=0 at starting time to initiate the ants. Putting ants in certain cities and setting $\tau_{ij}(0)=0$, the starting point will be the first information element in tabu list $tabu_k$. Moving probability $P_{ij}^k(t)$ of each ant from city i to city j based on state transition rule. All ants will complete one round of travel after time n. Tabu List $tabu_k$ will record and compute the travel path L_k for ant k over the round. The information will be adjusted per formula (4), the shorter of the path, the large of $\Delta\tau_{ij}{}^k(t,t+n)$ will be. The tabu list will be cleared when shortest path is found. The same procedure operated in cycle, when cycling counter $N_C = NC_{\max}$, the ant will find shortest and best path.

Ant algorithm is combined distributed calculation with feed-forward greedy search algorithm and global convergence capability, but the algorithm is lack of initial information for PID parameters optimization, and lead to slow commutating. When ant volumes are large, the algorithm need longer time in order to get a better solution. In order to overcome the above shortage, we introduced here in the paper PID parameter optimization based on hybrid classified adaptive ants algorithm [20-27].

3. Optimization Design Based on Gene and Ant Colony Algorithm of Parameters of PID Controller

3.1 Classical PID Controller

PID controller is a linearity controller, R (t) is input and Y (t) is actual output, control deviation as follows:

$$e(t) = R(t) - Y(t)$$

The control value u (t) consisting of proportion (P), integral (I) and differential (D) used to control target. The PID control principle is:

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt} \tag{5}$$

Where K_p is proportional factor, K_i is integral factor and K_d is differential factor.

Absolute integral bias is used to evaluate the performance in process control.

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

Punitive function is introduced in order to control over tuning. Overshoot will be part of target value when over tuning happened. Therefore, the most optimized performance is:

$$J(ITAT) = \int_0^t t\left|e(t)\right|dt + Ct\left|e(t)\right|dt \tag{7}$$

After discrediting formula (8):

$$u(t) = K_p\left\{e(t) + \frac{T}{T_i}\sum_{j=1}^{k} e(j) + \frac{T_d}{T}\left[e(k) - e(k-1)\right]\right\} \tag{8}$$

Where C is punitive factor in formula (7), C=0 if $e(t) \geq 0$.

T_i, T_d Is time factor of integral and differential in formula (8).

3.2 Introduction to PID Parameter Optimization Based On Self-Adaptive and Ants Mix Algorithm

3.2.1. Basic Concept: Each ant is represented by PID three parameters K_P, K_i, K_d. Over the full searching area, each ant will dynamically adjust moving direction and path, and approach target till the best path. The update of concentration of substance information is determined by target function. The optimized performance indicator of the three PID data is determined by nodes data of the ant system.

In order to improve the system performance of the ant system, the ACS (Ant Colony System) concept with classified adaptive ant algorithm is introduced here in the paper [7]. The improved moving philosophy is as follows: An ant at node r will move based on pseudo random proportional state transition rules which is defined by formula (10), tend to select a shorter but more informative side as moving direction, and then determine next city s, or it will adopt probability search per formula (1).

Rules 1: Setting

$$s = \begin{cases} \arg\max\limits_{u \in allowed_k} \left\{ [\tau(r,u)]^{\alpha} \bullet [\eta(r,u)]^{\beta} \right\}, & < q_0 \\ S & other \end{cases} \tag{9}$$

q Is equally distributed random data in [0, 1]. q_0 Is parameter $0 \le q_0 \le 1$, S is random variable based on probability distribution per formula (9).

Rules 2

An ant will use formula (10) as partial updating rules when it moving from i city to j city.

$$\tau(r,s) \leftarrow (1-\rho) \bullet \tau(r,s) + \rho \bullet \Delta\tau_1(r,s)$$

$$\Delta\tau_1(r,s) = Q_1 / R_{PID1} \tag{10}$$

ρ Is partial information parameter, $0 < \rho < 1$ R_{PID1} is the number of nodes over the path. The value of ρ is self-adaptive changed by algorithm when the local optimal solution is no obviously improved during N time's cycle. P min can prevent inducing the convergence speed caused by the too small value of ρ.

$$\rho(t) = \begin{cases} 0.95\rho(t-1) & if\, 0.95\rho(t-1) \ge \rho_{\min} \\ \rho_{\min} & else \end{cases} \tag{11}$$

Rules 3

When the travel round finished, the best path the ant searched could be globally updated per formula (12)

$$\tau(r,s) \leftarrow (1-\alpha) \bullet \tau(r,s) + \alpha \bullet \Delta\tau_2(r,s)$$

$$\Delta\tau_2(r,s) = Q_2 / R_{PID2} \tag{12}$$

α is global information volatility parameter. $0 < \alpha < 1$. R_{PID2} is the global best path at the time in the area. The value of α is self-adaptive changed by algorithm when the local optimal solution is no obviously improved during N times cycle. $\alpha_{\min}$ can prevent inducing the convergence speed caused by the too small value of α.

$$\alpha(t)=\begin{cases}0.95\alpha(t-1) & if\,0.95\alpha(t-1)\geq\alpha_{\min}\\ \qquad\qquad\alpha_{\min} & else\end{cases} \tag{13}$$

In order avoid earlier information privileged , the information range $[\tau_{\min,}\tau_{\max,}]$is given a condition. When information is updated, if $\tau_{ij}>\tau_{\max}$, then $\tau_{\max}=\tau_{ij}$,if $\tau_{ij}<\tau_{\max}$,then $\tau_{\min}=\tau_{ij}$

The conclusion from formula (13)

$$\tau_{\max}=\frac{1}{1-\alpha}\bullet\frac{Q}{R_{PID2}}\bullet m,\tau_{\min}=\frac{\tau_{\max}}{n} \tag{14}$$

n is coding length.

3.2.2 The Creation of Nodes and Path: Assuming PID control parameter, K_P, $K_{\rm i}$, K_d as variables to be optimized and assuming each of the parameter will have 5 valid data. Based on the data sampling rules of Z-N principle in PID control system, setting data before decimal of each 5 of the K_P, $K_{\rm i}$, K_d will occupy one bit, data before decimal of each will occupy 4 bits. Ant algorithm optimizing path could be abstractive plotted in 0-XY section by the 9 parameters. The procedure is: draw 15 pieces of line with equal intervals, equal length and perpendicular to X-axis, L2, L3... L15. As shown in Figure 1.

In figure 1, L5, L6-L10, L11-L15 represents the bit 1 to 5 of K_P, $K_{\rm i}$, K_d. Equally divide the line, total 10 sections are left and each represents the possible 10 data 0-9. In section 0-XY, there are total 15*10 data nodes. $Knot(x_i,y_{i,j})$ Stands for node, x_i is horizontal axis of line L_i, (i=1-15， x_i =0-15). $y_{i,j}$ Is vertical axis of L_i for node j. Each node represents one data , it is data on vertical axis $y_{i,j}$. Assuming an ant travel from origin of coordinates O, it will complete one round when it travels to a point in L_i . The path which ant traveled could be presented as:

$$Path=\begin{Bmatrix}O,Knot(x_1,y_{1,j}),Knot(x_2,y_{2,j}),\\ \ldots,Knot(x_{15},y_{15,j})\end{Bmatrix}$$

Where node $Knot(x_i,y_{i,j})$ located at $\mathrm{L_i}$, the data presented by the path could be concluded by [11]

$$\begin{cases}K_p=y_{1,j}\times10^0+y_{2,j}\times10^{-1}+y_{3,j}\times10^{-2}+y_{4,j}\times10^{-3}+y_{4,j}\times10^{-4}\\ K_i=y_{6,j}\times10^0+y_{7,j}\times10^{-1}+y_{8,j}\times10^{-2}+y_{9,j}\times10^{-3}+y_{10,j}\times10^{-4}\\ K_d=y_{11,j}\times10^0+y_{12,j}\times10^{-1}+y_{13,j}\times10^{-2}+y_{14,j}\times10^{-3}+y_{15,j}\times10^{-4}\end{cases} \tag{15}$$

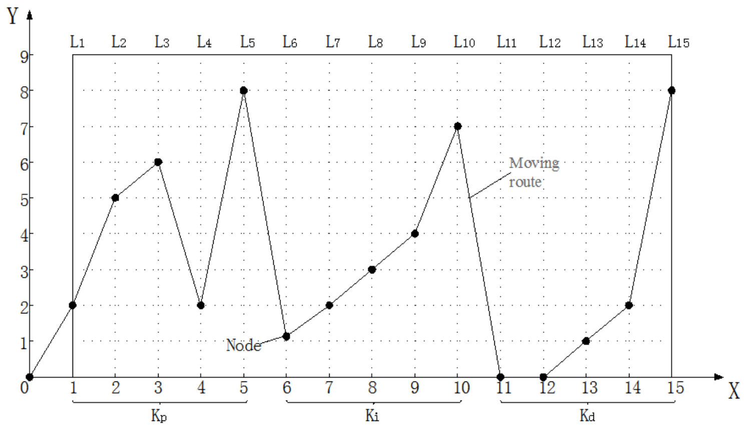


Figure 1. Nodes and Path Creation Schematics

as shown in Figure 1, the path PID parameter an ant traveled will be $K_P = 2.5628, K_i = 1.2347, K_d = 0.0128$.

3.2.3 Genetic Algorithm: The Generic Algorithm was originally proposed by Professor J.H Holland in 1975[7].The algorithm is global heuristic random searching algorithm simulates natural law of “, the survival of the fittest” in biological world. The algorithm target at certain object of the type and apply the optimized coding from natural law to the colony, continuously improve body fitness function value in cycles, searching for optimized method and best one through generic operation.

The algorithm consists of model setting, parameter coding, function of fitness design, genetic operator design, parameter setting and searching space. It’s characteristics are simple, high-efficiency, good robustness, parallel processing. Genetic algorithm has become an important branch of artificial intelligence research and was be used in signal processing, system identification, machine learning, fuzzy control, neural control and artificial life widely [13-17].

The Genetic algorithm is an iterative algorithm, It generated the initial solution randomly in the initial iteration period, and then generated the new solution randomly though simulated evolutionary and genetic operation. Each solution is be evaluated by the fitness function. The algorithm converges to an optimal solution after several iterative, it may be optimal or sub-optimal solution of the problem [13-19].

Usually, original colony randomly creates, and evaluated by adaptive function $f(x)$(x represents one sample). The selection, crossover and mutation become the basic operation in genetic algorithm. The main calculation procedure of traditional Genetic Algorithms as follows [13-19]:

1) Coding:

Before researching, the selected characteristic (gene) should be identified, and then, the data sheet of space solution is expressed as gene string type data of genetic space, the grouping of difference string compose different point.

2) Population initialization:

The evolution algebra counter is set t=0, the MAX evolution generation = T, generate N initial string structure data (as N individual) randomly form one population P (0). Genetic Algorithms begin iteration calculation from this status.

3) Individual evaluation:

The fitness of difference individual of population is be calculated and if the selected function of fitness is not best, the Genetic Algorithms maybe converged local optimization and can't solute the globe optimization.

4) Selection

Selection of Operator is be used for population, the chromosomes of optimized individual is be passed to offspring or randomly crossover to form new individual as parents generation. The roulette wheel selection which subject by Holland is be used as the selection operator now. In this method, the selected probability i of individual is directly proportional to fitness. The selected probability i of individual show individual fitness in proportion to combined of all individual fitness in the whole population. The individual can pair transactions for the crossover operation after be selected [12]:

5) Crossover

Crossover is two parent chromosomes exchange the partial gene according one rule to generate offspring. It is the main manner to producing new generation and decided the capacity of global searching. The new offspring inherit the characteristic of parent. Crossover shows the method of information exchange and is the core of Genetic Algorithms.

The crossover operator p_c used is one–cut point method usually in the coding manner. The actual manner is that one crossing point is set in individual gene string firstly, the digital after the crossing point of two individual is be exchanged and two new offspring is be generated. The single crossing point example as below:

Individual A: 1 0 0 1 ↑1 1 1→1 0 0 1 0 0 0 new individual
Individual B: 0 0 1 1 ↑0 0 0→ 0 0 1 1 1 1 1 new individual

6) Mutation

Mutation creates new individual by making changes in a single chromosome and provides a chance for creating new chromosomes. It is auxiliary manner for creates new individual and decided the local searching capacity of Genetic Algorithms. The mutation rate decide the proportion to new chromosome enter the population. The mutation rate is very low in Genetic Algorithms same with the biological world, usually the value is $0.001<p_m<0.01$.

There are two advantages when applying the mutation in Genetic Algorithms. Firstly, it make the Genetic Algorithms has local searching capacity random, The local searching capacity of mutation operator p_m can accelerate convergence to obtaining the best solution as the crossover operator p_c has approached the best solution. Secondly, it can keep the population diversity and avoid the precocious convergence [13-18].

Crossover operator p_c is main operator as the globe researching capacity and mutation operator p_m is auxiliary operator as the local researching capacity in Genetic Algorithms. Genetic Algorithms has the global and local balance researching capacity though using the pair of mutual cooperation and mutual competition operator.

7) The end condition: if t=T, the highest fitness individual which obtain in the process of evolution is the optimum solution. The value of generation is be preset 100~500 usually.

3.2.4 Analysis and Comparison of the Characteristics of Two Algorithms: ACS algorithm was an algorithm which was distributed computing method (avoiding premature convergence)and the positive feedback mechanism (fast finding better solutions) and greedy search algorithm (reducing search time). ACS was a strong parallel algorithm, it can start an independent solution search in the problem space of more points at the same time which not only increases the reliability of the algorithm, but also makes the algorithm has strong global search ability.

GA algorithm sets individual coding on the parameter set and can operate directly on the structure of the object.It uses the change rules of probability to determine the search direction from the string set and has large coverage which Can effectively avoid falling into local minimum point,It has the global optimal searching capability.

GA algorithm has strong practicability and large coverage which can effectively avoid the local minimum point.GA algorithm has self-organization, self-adaptation and self-learning characteristics with strong fault tolerance. It has strong extensibility(through pheromone cooperation).When the population size was large, GA algorithm need long time to search in order to get the optimal solution so the efficiency is lower than other traditional optimization methods. Premature convergence of GA was not conducive to find a better solution for easy appear stagnation phenomenon.

The paper describes the strategy of generic algorithm as follows [19]:

1) Genetic Coding: We apply decimal coding method here in the paper; the 3 PID parameters are presented by 5 decimal valid data which will be the operational targets.
2) Adaptive Function:

$$L_s = \sum_{i=1}^{15} d_i \qquad s = 1,2,...m \tag{16}$$

L_s is the sum of node data which ants traveled.

d_i is the node data of i node the ant roundly traveled, and sorted in ascending order by addictiveness.

3) The creation of initial colony: generated by random method.
4) Selection of operator: To select next node by Roulette rule per formula (10) .The probability to be selected for each individual objects are proportional to the value of adaptive function.
5) Crossover of operator p_c : as the primary operator of global searching capability.
6) Mutation of operator p_m : apply basic position mutation strategy as auxiliary operation in local, the probability is very low, $0.001 < p_m < 0.01$.

4. Procedure for PID Parameter Optimization Based On Hybrid Classified Adaptive Ants Algorithm

Step 1 To calculate PID parameter based on Z-N as $K_{P,Z-N}$ $K_{i,Z-N}$ and $K_{d,Z-N}$.

Step 2 Assuming number of ants m, data group $Path_k$ used to store 15 nodes vertical axis and its path for each ant k(k=1~m) is defined to have 15 elements.

Step 3 Parameter of Initial hybrid strategy: Setting timer t=0, cycle $N_C = 0$, the maximum cycle number is $NC_{\max}$ and concentration of each nodes information at initial time $\tau(x_i, y_{i,j}, 0)$ is c（i=1~15，j=0~9）, setting $\Delta\tau(x_i, y_{i,j}) = 0$, and then all ants will be set in origin O.

Step 4 Setting i=1. if $q < q_0$, then transition probability for these ants moving to each nodes of section L_i could be calculated by formula (9), or selecting next node per Roulette Wheel selection method, and store the data of node into tabu list $tabu_k$.

Step 5 When each ant finish every node, updating local partial information elements per formula (10), self-adaptive regulating local partial information volatility coefficient per formula (11).

Step 6 Setting i=i+1, i f i≤15, then jump to step 3, or to step 6.

Step 7 Based on data group $Path_k$(the path traveled by ant K), calculating relevant PID parameter $K_P{}^k$, $K_i{}^k$, $K_d{}^k$ per formula (15).Simulating in computer, the system performance index t_r^k, t_s^k and static deviation ess^k, overshoot ct^k could be concluded. Calculating targets function F_k for ant K by formula (7). Recording best path and optimized performance data, store $K_P{}^k$, $K_i{}^k$, $K_d{}^k$ into $K_P{}^*$, $K_i{}^*$, $K_d{}^*$

Step 8 Setting $t \leftarrow t+15$, $N_C \leftarrow N_C+1$, updating global information by formula (12), self-adaptive regulating global information volatility coefficient per formula (13).

Step 9 Hybridizing by single point cross strategy (starts when hybrid variable $\gamma < 0.000001$), generate new one.

Step 10 By basic transition strategy (starts when variation operator $p_m < 0.01$), calculating every parameter again. If the calculated performance index close to target function F_k, store variation and update information, the scope is determined by per formula (14), or ignore them.

Step 11 if $N_C < NC\max$ and whole ant colony are not yet converged to same path, then setting all ants to original point O, and jump to step 4. Otherwise, end the cycling, and output best path and relevant optimized PID parameter $K_P{}^*$, $K_i{}^*$, $K_d{}^*$.

5. Simulator

The optimization algorithm on this paper is be used for a electro-hydraulic servo system of jumbler as an example $G(s) = \dfrac{5268}{S(S^2+133S+110889)}$, simulating the algorithm from transition function.

In simulating test, setting system input as unit step signal. Setting $Q=1, \rho=0.7, \alpha=1$, ants number is 30, repeater times is 100. The scope of parameters of PID are: K_p is [0.00001 20], K_i, K_d is [0.00001 2].

For analysis performance of this algorithm on this paper, it is compared with N-Z, ACS, GA algorithm.

Figure 2 is the unit step responds chart. Table 1 is the unit step responds table.

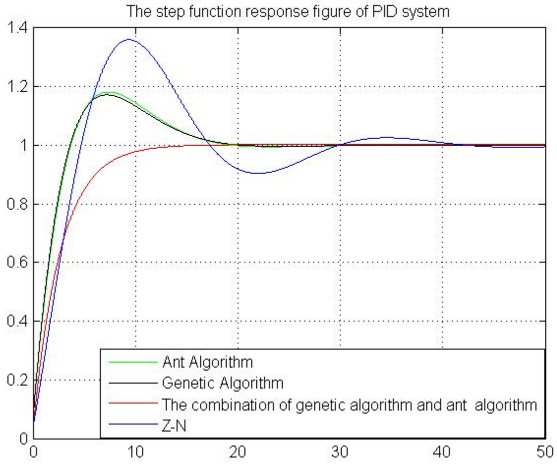


Figure 1. Unit Step Responds Chart

Table 1. Comparison Table of System Unit Step Performance

ALGORITHM	Kp	Ki	Kd	tr	ts	ess
Z-N	4.58	1.6	0.8	3.8	37.35	8.2574e-8
ACS	10.0	1.9944	1.6878	2.99	16.58	4.4409e-14
GA	10.5228	1.9999	1.5159	2.91	16.72	3.3307e-14
ACS-GA	8.0112	0.0071	0.6488	5.91	10.25	0.1676

Reviewing Table 1 and comparing ACS-GA step performance and other control algorithm, three PID parameters has optimized, static bias ess, control time ts and overshoot ct are reduced dramatically. Therefore, the algorithm has best control performance and robustness.

6. Conclusions

Genetic algorithm is easy to repeat and overlap, and slow down solving efficiency. Ant colony algorithm has slow speed convergence and will impact global searching capability. Therefore, PID parameter optimization based on self-adaptive and ants mix algorithm. Its basic Concept is the initial information of K_P^k, K_i^k, K_d^k are obtained through select and crossover of Genetic algorithm, and then the ant colony algorithm is be using, by self-adapter adjusting the route choice probability and pheromone updating rule, finally to search out the optimal value of the PID parameter. The simulating results indicate the algorithm has fast and global convergences, low sensitiveness of parameters, solving speed are also increased. It is better way for parameter optimization in practical engineering.

References

[1] Y.-H. Wang, "Modern electrical control and the application of PLC Technology [M]", Beijing University Press, (**2008**).

[2] J. G. Ziegler and N. B. Nichols, "Optimum setting for automatic controllers [J]", ASME Transactions, vol. 11, (**1942**), pp. 759-768.

[3] J. W. Lee, J. Young and K. S. Lee, "Anti-sway controller of an ATC using NN predictive PID control [C]", IECON Proceedings (Industrial Electronics Conference), IECON 2004-30th Annual Conference of IEEE Industrial Electronics Society, (**2004**), pp. 2998-3003.

[4] H.-X. Li, L. Zhang and K.-Y. Cai, "An improved robust fuzzy-PID controller with optimal fuzzy reasoning [J]", IEEE Transactions on systems, Man, and Cybernetics, Part B: Cybernetics, vol. 35, no. 6, (**2005**), pp. 1283-1294.

[5] W. K. Ho, C. C. Hang and L. S. Cao, "Tuning of PID controllers based on gain and phase margin specifications [J]", Automatica, vol. 31, no. 3, (**1995**), pp. 497-502.

[6] G.-Z. Tan and A. Li, "Fuzzy PID controller with in complete derivation and its application to intelligent bionic artificial leg [J]", G, Nichols N B. Optimum setting for automatic controllers, ASME Transactions, vol. 11, (**1942**), pp. 759-768.

[7] S.-y. Li, "Ant Colony Algorithms with Applications [M]", Harbin Institute of Technology, (**2004**).

[8] M. Dorigo, V. Maniezzo and A. Colorni, "Positive Feedback as a Search Strategy", Technical Report, pp. 91-106.

[9] S. Gao and J.-y. Yang, "Swarm Intelligence Algorithms and Applications [M]", China Water Power Press, (**2006**).

[10] S. Tao and L. Shaowei, "Research on Parameters Adjustment of PID Controller Based on ACO-GA Hybrid Strategy [J]", Journal of Projectiles, Rockets, Missiles and Guidance, vol. 29, no. 1, (**2009)**, pp. 73-76.

[11] P.-fu. Peng and Y.-p. Lin, "Optimal PID Control of Self-Adapted Ant Colony Algorithm Based on Genetic Gene [J]", ACTA ELECTRONICA SINICA, vol. 34, no. 6, (**2006**), pp. 1109-1113.

[12] J. Li and X.-h. Qi, "Four Kinds of Advanced PID Control Methods and Comparison of Their Performance [J]", Computing Technology and Automation, vol. 31, no. 3, (**2012**), pp. 19-24.

[13] N. A. Edwin Hou, "Computational Intelligence for Optimization [M]", Tsinghua University Press, **(1999**).

[14] J. Holland, “Adaption in Natural and Artificial Systems”, University of Michigan Press, Ann Arbor, 1975; MIT Press Cambridge, MA, (**1992**).

[15] M. Gen and R. Cheng, “Genetic Algorithms and Engineering Optimization [M]”, Tsinghua University Press, (**2004**).

[16] R. Durbin and D. Willshaw, “An Analogue Approach to the Traveling Salesman Problem Using an Elastic Net Method”, Nature, [J], vol. 326, (**1987**), pp. 689-691.

[17] J.-k. Ge and Y.-h. Qiu, “Summary of genetic algorithms research [J]”, Application Research of Computers, vol. 25, no. 10, (**2008**), pp. 2911-2916.

[18] J.-h. Zhang, “Research on Key Technologies of Multidisciplinary Design Optimization Based on Complex Mechanical Product and its Application in Construction Machinery [D]”, Xi’an University of Architecture and Technology, (**2011**).

[19] X. Zhang and J. Hong, “Optimization Design Based on Improved GAAA Algorithm for Parameters of PID Controller [J]”, Computer Measurement Controller, vol. 14, no. 11, (2006), pp. 1545-1547.

[20] C. C. Hang and W. K. Ho, “COALS, A comparison of two design methods for PID Controllers”, ISA TRANS, vol. 33, no. 2, (**1994**), pp. 147-151.

[21] K. J. Astrom, T. Hangglund, C. C. Hang and W. K. Ho, “Automatic tuning and adaptation for PID controller -A survey”, Control Engineering Practice, vol. 1, (**1993**), pp. 699-714.

[22] M. Dorigo, V. Maniezzo and A. Coloni, “The ant system: optimization by a colony of cooperation agents [J]”, IEEE Transactions on System, Man and Cybernetics-part B, vol. 26, no. 1, (**1996**), pp. 29-42.

[23] M. Dorigo and V. Maniezzo, “Distributed optimization by colonies [C]”, Proceedings of the Frist European Conference on Artificial L life London: M IT Press, (**1991**), pp. 134-142.

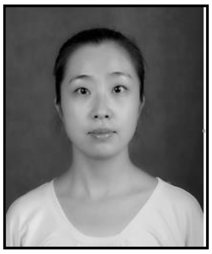
[24] Q. Zhong, C. Li and Y. Liu, “Fast Multi-swarm Optimization with Cauchy Mutation and Crossover Operation [A]”, Lecture Notes in Computer Science [c], Berlin/Heidelberg, Springer, (**2007**), pp. 344-352.

[25] K. R. C. Eberhart, “Particle Swarm Optimization [C]”, Proceedings of the IEEE Conference on Neural Networks, IV. Piscataway, NJ, (**1995**), pp. 1942-1948.

[26] A. Colorni, M. Dorigo, V. Maniezzo and M. Truian, “Ant system for job-shop scheduling”, Belhian J. Oper. RES Statist. Computer Science, vol. 34, no. 1, (**1994**), pp. 34-53.

[27] M. Dorigo, G. Caro Di and L. M. Gambardella, “Ant algorithms for discrete optimization”, Artificial Life, vol. 5, no. 2, (**1999**), pp. 137-172.

Author



Wang Xiao-yu, (1974-), female (Han), Shaanxi Xi’an, PhD, senior engineer, teacher of college of Engineering, Xi’an International University, main research field for the mechanical design and theory, control theory and control engineering; published 20 papers, include 6 papers for EI,2 papers for ISTP,3 papers for CSCD, 11 papers for the Core Journal of China .