Available online at www.sciencedirect.com

SCIENCE DIRECT

Journal of Bionic Engineering 3 (2006) 073–078

Novel Approach to Nonlinear PID Parameter Optimization Using Ant Colony Optimization Algorithm

Duan Hai-bin[1], Wang Dao-bo[2], Yu Xiu-fen[3]

1. *School of Automation Science and Electrical Engineering, Beijing University of Aeronautics and Astronautics, Beijing* 100083, *P. R. China*
2. *College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing* 210016, *P. R. China*
3. *Centre for Space Science and Applied Research, Chinese Academy of Sciences, Beijing* 100080, *P. R. China*

Abstract

This paper presents an application of an Ant Colony Optimization (ACO) algorithm to optimize the parameters in the design of a type of nonlinear PID controller. The ACO algorithm is a novel heuristic bionic algorithm, which is based on the behaviour of real ants in nature searching for food. In order to optimize the parameters of the nonlinear PID controller using ACO algorithm, an objective function based on position tracing error was constructed, and elitist strategy was adopted in the improved ACO algorithm. Detailed simulation steps are presented. This nonlinear PID controller using the ACO algorithm has high precision of control and quick response.

Keywords: Ant Colony Optimization, algorithm, pheromone, nonlinear PID, parameter optimization

1 Introduction

The widely used PID industrial controller uses a combination of proportional, integral and derivative action on the control error to form its output. Owing to its simple structure, easy tuning and effectiveness, this technology has been a mainstay for long among practicing engineers.

The linear combination of these components can at most achieve a compromised performance in terms of system response speed and stability. Although conventional PID is popular, it is difficult to obtain satisfying control results to use the controller for systems or processes which are nonlinear, have long delay, or time-varying and strong cross-coupling. In order to improve the performance of conventional PID, Han developed a kind of nonlinear PID controller with the use of nonlinear characteristics. A nonlinear combination can provide an additional degree of freedom to achieve a much improved system performance [1]. However, this improvement can be achieved only at the expense of higher complexity in the controller. Artificial intelligence approaches can alleviate some of the difficulties by introducing available information into the control design.

The Ant Colony Optimization (ACO) algorithm is a meta-heuristic algorithm for the approximate solution of combinatorial optimization problems that has been inspired by the foraging behaviour of real ant colonies[2–4]. In this algorithm, computational resources are allocated to a set of relatively simple agents that exploit a form of indirect communication mediated by the environment to find the shortest path from the ant nest to a set target. Ants can follow through to a food source because, while walking, they deposit pheromone on the ground, and they have a probabilistic preference for paths with larger amount of pheromone[5,6] (Fig. 1).

Considering Fig. 1a, ants arrive at a point where they have to decide whether to turn left or right. Since they have no clue of which is the better choice, they choose randomly. It can be expected that, on average,

Corresponding author: Duan Hai-bin
E-mail: hbduan@buaa.edu.cn

half of the ants decide to turn left and the other half to turn right. This happens both to ants moving from left to right (those whose name begins with an L) and to those moving from right to left (name begins with a R). Fig. 1b and Fig. 1c show what happens in the immediate instants, supposing all ants walk at approximately the same speed. The number of lines is roughly proportional to the amount of pheromone that the ants have deposited. Since the lower path is shorter than the upper one, more ants will visit it on average, and therefore pheromone accumulates faster. After a short period the difference in the amount of pheromone on the two paths is large enough to influence the decision of new ants coming into the system (Fig. 1d). From now on, new ants will prefer the lower path, since at the decision point they perceive a greater amount of pheromone on this lower path. This in turn increases, with a positive feedback, the number of ants choosing the lower, shorter path. Very soon all ants will use the shorter path. This process is thus characterized by a positive feedback loop, where the probability with which an ant chooses a path increases with the number of ants that previously chose the same path.

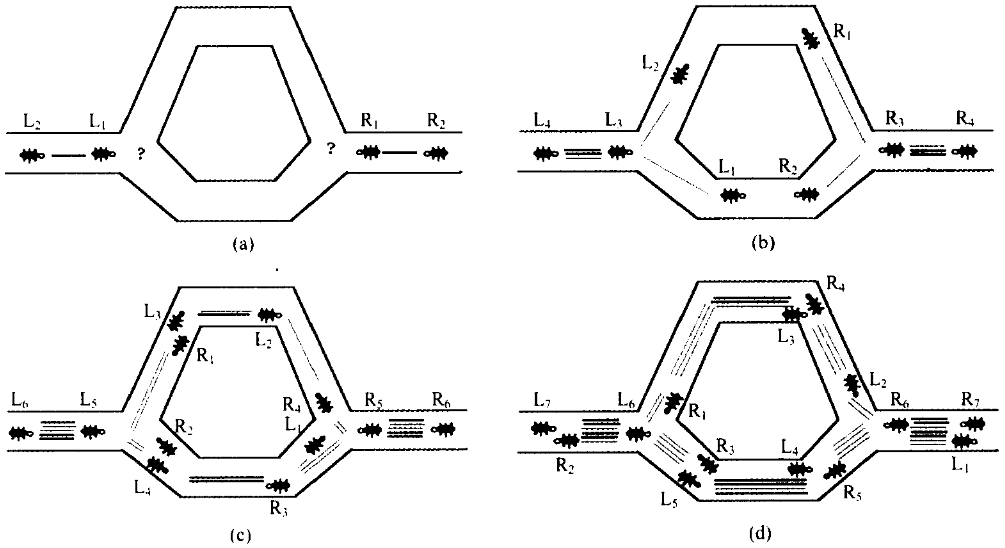

Fig. 1 How real ants find a shortest path.

This behaviour inspired the ACO algorithm in which a set of artificial ants cooperate in the solution of a problem by exchanging information via pheromone deposited on graph edges. The ACO algorithm has been applied to combinatorial optimization problems such as the Travelling Salesman Problem (TSP), Quadratic Assignment Problem (QAP), and so on.

We present here a novel parameter optimization strategy for a nonlinear PID controller using the ACO algorithm.

2 Nonlinear PID controller

The control law of the common nonlinear PID controller is presented as

$$u = K\left(\left|\frac{\dot{e}}{T_{\mathrm{i}}}\right|^{\lambda}\operatorname{sign}(\dot{e})+|e|^{\lambda}\operatorname{sign}(e)+\left|\frac{\int e\mathrm{d}t}{T_{\mathrm{d}}}\right|^{\lambda}\operatorname{sign}\left(\int e\mathrm{d}t\right)\right) \quad (1)$$

where λ is a constant, K is the proportional gain, T_{i} is the integral time constant, T_{d} is the derivation time constant, u is the combination of error, differentor of error and integrator of error. In order to improve the control quality, a novel type nonlinear PID controller can be constructed as shown in Fig. 2.

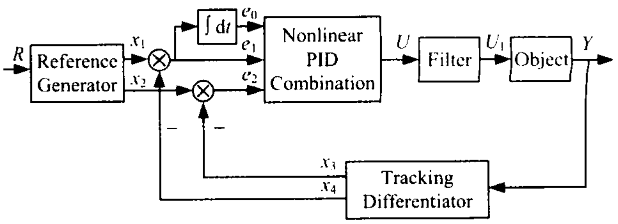

Fig. 2 Nonlinear PID controller.

The differential tracker is used to track the output position and its differential. The reference generator is used to generate reference input signals[1]. From Fig. 2, the input error signals for the nonlinear PID combination can be expressed as

$$\begin{cases} e_1 = x_1 - x_4 \\ e_2 = x_2 - x_3 \\ e_0 = \int_0 e_1(\tau)\mathrm{d}\tau \ , \end{cases} \tag{2}$$

where e_1 denotes angular position tracking error, e_2 denotes angular velocity tracking error.

The reference generator is expressed as

$$\begin{cases} \dot{x}_1 = x_2 \\ A_1 = x_1 - R + \dfrac{x_2|x_2|}{2R_1} \\ \dot{x}_2 = -R_1 \mathrm{sat}(A_1, \delta_1) \ . \end{cases} \tag{3}$$

Similarly, for tracking differentiator

$$\begin{cases} \dot{x}_3 = x_4 \\ A_2 = x_3 - Y + \dfrac{x_4|x_4|}{2R_2} \\ \dot{x}_4 = -R_2 \mathrm{sat}(A_2, \delta_2) \ , \end{cases} \tag{4}$$

where Y denotes the feedback position of the object, and

$$\mathrm{sat}(A, \delta) = \begin{cases} \mathrm{sgn}(A) & \text{if} \quad |A| \geq \delta \\ \dfrac{A}{\delta} & |A| < \delta \ , \end{cases} \tag{5}$$

where δ is a constant and, empirically, δ= 0.1 to 0.2[7]. Therefore, we get the nonlinear PID combination

$$\begin{cases} \dot{x}_5 = x_1 - x_4 \\ e_0 = x_5 \ , \end{cases} \tag{6}$$

$$\begin{cases} U = k_\mathrm{i}\mathrm{fal}(e_0, \alpha, \delta) + k_\mathrm{p}\mathrm{fal}(e_1, \alpha, \delta) + k_\mathrm{d}\mathrm{fal}(e_2, \alpha, \delta) \\ \dot{x}_6 = -\rho_0(x_6 - u_1) \ , \end{cases} \tag{7}$$

where α is a constant, empirically, α = 0.5 to 1.0. k_i denotes integral coefficient, k_p denotes a proportional coefficient, k_d denotes a differential coefficient, ρ_0 is a constant. The proportional term makes the controller respond to the error while the integral and derivative help to eliminate steady state error and prevent overshoot respectively. fal(e,α,δ) is a selected nonlinear function. Fig. 3 shows the characteristics of the fal(e,α,δ) function. A linear relationship is thus effectively used when $e < |\delta|$. Therefore, the fal(e,α,δ) function can provide a smoother control action when e is near zero.

Suppose the system error $\varepsilon = R - Y$ and let the objective function be

$$J = \int_0 \varepsilon^2(t)\mathrm{d}t \ , \tag{8}$$

which is also named a cost function.

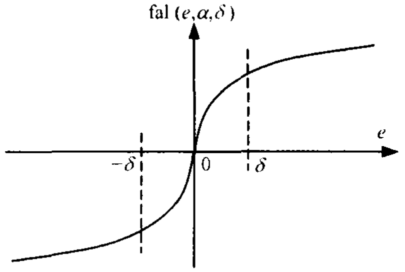

Fig. 3 Diagram of fal(e, α, δ) function characteristics.

3 Improved ACO algorithm

The ACO algorithm has first been applied to the Travelling Salesman Problem (TSP)[8–12]. The aim is to find the shortest path that traverses all cities (nodes) in the problem exactly once, returning to the starting node. In the classic TSP, let $V = \{a,\ldots, z\}$ be a set of nodes, $A = \{(i,j): i,j \in V\}$ be the edge set. The distance between each pair of nodes i and j is represented by $d(i,j)$ and $d(i,j) = d(j,i)$ is a cost associated with edge $(i,j) \in A$. The TSP problem is to find a minimal cost closed tour that visits each node once. Nodes $i \in V$ are given by their coordinates (x_i, y_i) and $d(i,j)$ is the Euclidean distance between i and j, when we have a Euclidean TSP. In solving nonlinear PID problems, let $d(i,j) = \Delta J(i,j)$, and $\Delta J(i,j)$ is defined as

$$\Delta J(i,j) = J_i - J_j \ \forall \ i,j \, . \tag{9}$$

Let m be the number of ants in the colony, and m ants are scattered randomly on these nodes. At discrete time steps, all ants select their next nodes then simultaneously move to their next nodes. Ants deposit pheromone on each edge they visit to indicate the utility of these edges[9–12]. The accumulated strength of pheromone on edge (i,j) is denoted by $\tau(i,j)$.

We assume that a real ant k located at node i chooses its next node j by applying Eq. (10) and Eq. (11), which are also called the state transition rule. Eq. (10) is a greedy selection technique favouring links with the best combination of short distance and large pheromone levels. Eq. (11) balances this by allowing a probabilistic selection of the next node.

$$s = \begin{cases} \arg\max\limits_{u \in J_k(i)} \{\tau(i,u)[\Delta J(i,u)^\beta]\} & \text{if} \quad q \leq q_0 \\ \text{Eq. (11)} & \text{otherwise,} \end{cases} \tag{10}$$

$$p_k(r,s)=\begin{cases}\dfrac{\tau(i,j)[\Delta J(i,j)]^{\beta}}{\sum\limits_{u\in J_k(r)}\tau(i,u)[\Delta J(i,u)]^{\beta}} & \text{if} \quad s\in J_k(i)\\ 0 & \text{otherwise,}\end{cases} \tag{11}$$

where $q\in[0,1]$ is a uniform random number, q_0 is the proportion of occasions when the greedy selection technique is used. It determines,the relative importance of exploitation versus exploration, the smaller q_0, the higher the probability of using the rule of Eq. (10). If $q\leq q_0$, then the best edge is chosen through a transition by exploiting accumulated knowledge, if $q>q_0$, the new edge is chosen through a transition by exploration biased towards short and high trail edges. β weighs the relative importance of the heuristic value. If $\beta=0$, only pheromone amplification will occur and the distance between nodes has no direct influence on the choice. $J_k(i)$ is the set of nodes yet to be visited by ant k.

When the ants move between the nodes, the pheromone level on the selected edge $\tau(i,j)$ is updated according to the local updating rule of Eq. (12). In this way, ants will make better use of their pheromone trail information; without local updating, all ants will search in a narrow neighbourhood of the best previous tour.

$$\tau(i,j)\leftarrow(1-\rho)\cdot\tau(i,j)+\rho\tau_0\ , \tag{12}$$

where ρ is the local pheromone decay parameter, and $\rho\in(0,1)$, τ_0 is the initial value of pheromone deposited on each of the edges.

Once all ants have constructed a tour, global updating of the pheromone takes place. Here, we use elitist strategy: edges that give the best solution are rewarded with a relatively large increase in their pheromone level. This is expressed as

$$\tau(i,j)\leftarrow(1-\gamma)\cdot\tau(i,j)+\gamma\cdot\Delta\tau(i,j)\ , \tag{13}$$

where γ is the global pheromone decay parameter, and $\gamma\in(0,1)$, $\Delta\tau(i,j)$ is given by

$$\Delta\tau(i,j)=\sum_{k=1}^{m}\Delta\tau_k(i,j)\ , \tag{14}$$

in which

$$\Delta\tau_k(i,j)=\begin{cases}\dfrac{Q}{\Delta J_k(i,j)} & \text{if} \quad k\text{th ant use edg } (i,j)\\ & \text{and } (i,j)\in \text{globally best tour}\\ 0 & \text{otherwise.}\end{cases} \tag{15}$$

$\Delta\tau(i,j)$ is used to reinforce the pheromone on the edges of the global best solution, $\Delta\tau_k(i,j)$ is the quantity of pheromone in an ant, Q is a problem dependent parameter, $\Delta J_k(i,j)$ is value of objective function, which is computed according to Eq. (9).

This continues until a certain number of iterations, or the solution, have been achieved.

4 Steps of solving nonlinear PID with improved ACO algorithm

The steps of the improved ACO algorithm in optimizing the parameters of the nonlinear PID controller can be described as follows:

Step 1. We set the maximum number of iterations N_C = Max., the number of iterations $n_c\leftarrow 0$ and the initial accumulated strength of pheromone on edge (i,j), $\tau(i,j)\leftarrow\tau_0$. At this level each ant is positioned on a starting node. We also initialize the nonlinear PID parameters δ_1, δ_2, R_1, R_2, α. This is the initialization step.

Step 2. At this level each loop is called a step. For each ant, we compute its transition probability given by Eq. (10) and Eq. (11). This is the construction step. The control law of the common nonlinear PID controller is presented as following:

Step 3. We incrementally build a solution and a local pheromone updating rule according to Eq. (12) until all ants have built a complete solution and we record the best solution found by now. This is the local trail update step.

Step 4. A global pheromone is updated by applying the global updating rule of Eq. (13), Eq. (14) and Eq. (15). We set the new number of iterations $n_c\leftarrow n_c+1$. This is the global trail update step.

Step 5. If the termination condition ($n_c>N_C$) is not met, return to Step 2. Otherwise, print the current best parameter solutions of the nonlinear PID controller, which are k_p, k_i and k_d. This is the termination step.

5 Simulation example

The parameter optimization strategy proposed in the previous section can be appraised by simulation. We take a novel flight simulation turntable as an example, which has high requirements for tracking, *i.e.* high precision, fast tracking response and high reliability. The flight simulation turntable is an important device and a typical high performance servo system used in the hardware

-in-the-loop simulation of a flight control system. It can simulate dynamic characteristics and various flying postures. The turntable is driven by three DC motors which have three rotating parts, the roll axis including simulation object; the pitch axis including roll motor; and the yaw axis including pitch motor.

The simplified transfer function of the flight simulation turntable is given as

$$G(s)=\frac{6491}{s^2+198.21s+6028}\ . \tag{16}$$

From Eq. (16), it is clear that the flight simulation turntable is a typical second order servo system.

The relative parameters are $\delta_1=0.14$, $\delta_2=0.17$, $R_1=500$, $R_2=300$, $\alpha=0.8$. The main aim is to calculate k_p, k_i and k_d expressed in Eq. (7). The ACO algorithm has been coded in Matlab 6.5 and implemented on a PC with 256 MB RAM under Windows 2000. The ACO parameters were set to the following values: $\tau=0$, $\beta=4$, $\rho=0.5$, $\gamma=0.6$, $Q=200$, $m=20$, $N_C=150$. The sampling time is 0.8 ms. Through simulation, we could get the optimized corresponding nonlinear PID parameters as

$$\begin{cases} k_p^*=8.4 \\ k_i^*=42.7 \\ k_d^*=23.2\ . \end{cases} \tag{17}$$

To assess the effectiveness of the optimized nonlinear PID parameters, we tested the tracking of the whole system, with the standard square wave response (Fig. 4), and the corresponding system velocity output (Fig. 5). For the stringent square wave, tracking error is very small, and the velocity output changes quickly when the motion switches direction.

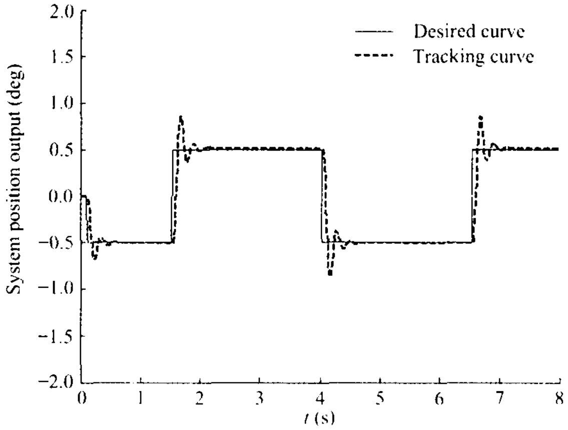

Fig. 4 Tracking square wave through simulation.

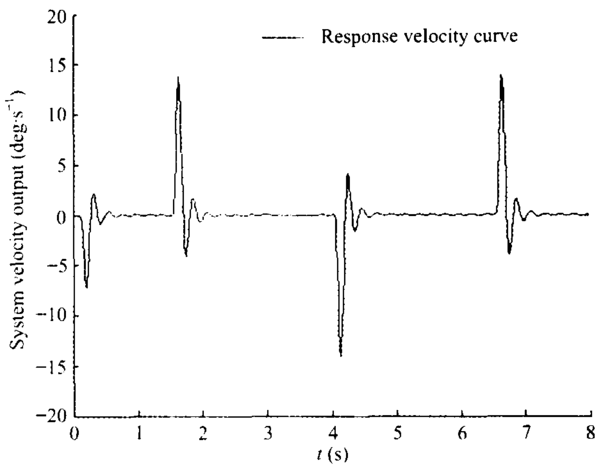

Fig. 5 System velocity output through simulation.

6 Conclusions

The ACO algorithm shows great advantages in solving combination optimization problems because it is realized easily, and is extraordinarily adaptable. It is significant to enlarge the application of ACO algorithm further.

In this paper, we have introduced an ACO algorithmic approach for optimizing the parameters of a nonlinear PID controller. From the results of simulation, we conclude that this proposed nonlinear PID controller using the improved ACO algorithm is very effective. It possesses good control and robust performance, and can be widely used to control different kinds of object and process.

Acknowledgements

This work was partially supported by "333" Talent Project Foundation of P. R. China and National Aeronautical Science Foundation of P. R. China.

The authors are grateful to reviewers for their helpful comments and suggestions.

References

[1] Han J Q. Nonlinear PID controller. *Acta Automatica Sinica*, 1994, **20**, 487–490.

[2] Colorni A, Dorigo M, Maniezzo V. Distributed optimization by ant colonies. *Proceedings of the First European Conference on Artificial Life*, (editors) Varela F, Bourgine P. Elsevier Publishing, Paris, 1992, 134–142.

[3] Bonabeau E, Dorigo M, Theraulaz G. Inspiration for optimization from social insect behavior. *Nature*, 2000, **406**,

39–42.

[4] Katja V, Ann N. Colonies of learning automata. *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, 2002, **32**, 772–780.

[5] Dorigo M, Gambardella L M. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1997, **1**, 53–66.

[6] James M, Marcus R. Anti-pheromone as a tool for better exploration of search space. *Proceedings of the 3rd International Workshop ANTS*, Brussels, 2002, 100–110.

[7] Tan K K, Lee T H, Zhou H X. Micro-positioning of linear-piezoelectric motors based on a learning nonlinear PID controller. *IEEE/ASME Transactions on Mechatronics*, 2001, **6**, 428–436.

[8] Dorigo M, Maniezzo V, Colorni A. Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, 1996, **26**, 29–41.

[9] Duan H B, Wang D B. A novel improved ant colony algorithm with fast global optimization and its simulation. *Information and Control*, 2004, **33**, 241–244, (in Chinese).

[10] Duan H B, Wang D B, Zhu J Q, Huang X H. Development on ant colony algorithm theory and its application. *Control and Decision*, 2004, **19**, 1321–1326, 1340, (in Chinese).

[11] Duan H B, Wang D B, Yu X F. Research on the optimum configuration strategy for the adjustable parameters in ant colony algorithm. *Journal of Communication and Computer*, 2005, **2**, 32–35.

[12] Ma L, Yao J, Fan B Q. The application of ant algorithm in traffic assignment. *Bulletin of Science and Technology*, 2003, **19**, 377–380, (in Chinese).