Proceedings of 2014 IEEE
International Conference on Mechatronics and Automation
August 3 - 6, Tianjin, China

PID Controller Design Based on Prey-Predator Pigeon-Inspired Optimization Algorithm

Hang Sun, and Haibin Duan, *Senior Member, IEEE*
Science and Technology on Aircraft Control Laboratory
School of Automation Science and Electrical Engineering, Beihang University, Beijing, 100191, P. R. China

sunhang823@163.com; hbduan@buaa.edu.cn

***Abstract* - Pigeon-Inspired Optimization (PIO) algorithm is a recently proposed bio-inspired swarm intelligence optimizer. High convergence speed is its most outstanding advantage. However, PIO algorithm can easily trap into a local optimal solution, which is the main defect that limits its further application. To overcome this defect, a Prey-Predator PIO algorithm is proposed in this paper, which combines the standard Pigeon-Inspired Optimization algorithm and the Prey-Predator strategy. This new algorithm can avoid the disadvantage which standard Pigeon-Inspires optimization has. In this paper, comparative experiments on the Proportion-Integral-Derivative (PID) parameter adjustment are conducted by using Particle-Swarm Optimization (PSO), PIO and Prey-Predator PIO, and the comparative results demonstrate our proposed approach is more feasible and effective.**

Index Terms - Pigeon-Inspired Optimization (PIO), Prey-Predator, Proportion-Integral-Derivative (PID), parameter adjustment.

I. INTRODUCTION

Proportion-Integral-Derivative (PID) controller has simple structure and can fulfill a large number of requirements in industrial field [1]. PID is still the most classic and popular basic control method in automatic control engineering aspect. Parameters optimization is the key to design the PID control system. Traditional PID controller parameters are usually optimized artificially by experiments. This kind of optimization requires not only skills, but also plenty of time. Even more important, when the control parameters need change along with the feature of the plant, PID controller has no adaptive ability. Recently, boil-inspired swarm intelligence optimizers have been used to adjust PID parameters. It has already shown strong vitality and great potential in complex optimization problems and practical application.

Researches in insect and swarm intelligence have provided computer scientists with powerful method for designing distributed control and optimization algorithm[2]. Many computer scientists have done a large number of researches on this aspect. M. Dorigo proposed Ant Colony Optimization (ACO) by simulation ant colonies' foraging behavior in 1991[3]. Particle Swarm Optimization (PSO) algorithm is firstly proposed by Psychologist Kennedy and Electrical engineer Eberhart, in 1995[4]. Many other bio-inspired optimizations are also proposed in recent years, such as Artificial Bee Colony (ABC) algorithm, Differential Evolution (DE) algorithm and Genetic Algorithm (GA). Bio-inspired intelligent algorithms have already become a heated topic that attracts much attention in scientific researches because of their excellent performance [5].One of the inevitable trends for the future information technology is intelligence, and the effective way to achieve intelligence is to simulate the various intelligent behaviors in nature. Many of the adaptive optimization phenomena in nature inspire us that many highly complex optimization problems can be perfectly solved with the self-evolution in organisms and ecological systems[6].

Pigeon-Inspired Optimization (PIO) algorithm is a new swarm intelligence algorithm inspired by the homing behaviors of pigeons, proposed by H. Duan[7]. Pigeons are one of the most worthy researched birds in the world, especially on their homing behaviors[8]. Investigations of pigeons' skills to ascertain different magnetic fields demonstrate that the pigeons' impressive homing abilities practically depend on tiny magnetic particles in their beaks. More specifically, pigeons have unique noses which can distinguish the direction because there are iron crystals in pigeons' beaks. Evidences suggest that pigeons have a special system where signals from magnetite particles are transmitted from their nose to the brain by using their trigeminal nerve. Researches on pigeons' behaviors show that pigeons can follow some landmarks, such as main roads, railways and rivers rather than head for their destinations directly. Pigeons' ability to distinguish differences in altitude between the sun at the point of release and at the home base interprets that the sun is also involved in pigeon navigation. PIO is inspired by these above features of pigeons. PIO can solve a large number of industrial control problems with high convergence and it has a vast prospect for development.

However, PIO can easily trap into local optimal solution. To overcome this disadvantage of PIO, a new ameliorative Pigeon-Inspired Optimization algorithm, Prey-Predator Pigeon-Inspired Optimization (PPPIO), is firstly proposed to optimize the PID parameters in this paper. Compared with PIO algorithm, PPPIO algorithm has better global search ability and can also avoid trapping into local optimal solution effectively. In this paper, some comparative experiments about the PID parameter adjustment are made by PSO, PIO and PPPIO. The results of the comparative experiments demonstrate the approach of PPPIO is feasible and effective.

The remainders of this paper are organized as follows. Section II introduces the mathematical model of the PIO. Subsequently, Section III describes PPPIO for the PID

978-1-4799-3977-0/14/$31.00 ©2014 IEEE

controller design. Experiments and analysis are given in section IV. Concluding remarks are contained in section V.

II. Pigeon-Inspired Optimization Algorithm (PIO)

Homing pigeons can easily find their homes by using three homing tools: magnetic field, sun and landmarks[6]. Pigeons can sense the earth field by using magneto receptions to shape the map in their mind. Their compasses to adjust the directions rely on the altitude of the sun. As getting near to the destination, they depend less on magnetic particles and the sun. However, when getting near to the destination, they use landmarks to locate themselves. If they are familiar to the landmarks, they will fly straightly to the destination. If they are unfamiliar with the landmarks, they will follow the pigeons which are familiar with the landmarks. Map and compass operator model is presented based on magnetic field and the sun, while landmark operator model is presented based on landmarks.

In PIO model, virtual pigeons are used naturally. In this map and compass operator, the rules are defined with the position X_i and the velocity V_i of pigeon *i*, and the positions and velocities in a D-dimension search space are updated in each iteration. The new position X_i and velocity V_i of pigeon *i* at the *t*-th iteration can be calculated with the following Eq. (1) and (2).

$$V_i(t) = V_i(t-1)\cdot e^{-Rt} + rand\cdot(X_g - X_i(t-1)) \tag{1}$$

$$X_i(t) = X_i(t-1) + V_i(t) \tag{2}$$

where R is the map and compass factor, *rand* is a random number, and X_g is the current global best position, and which can be obtained by comparing all the positions among all the pigeons.

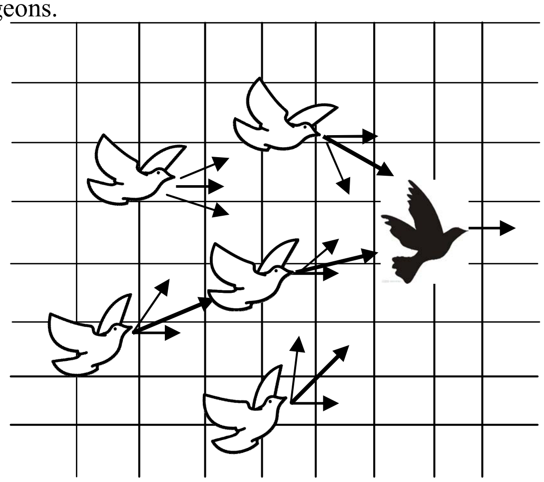

Figure1: Map and compass operator model of PIO

As shown in Figure 1, the best positions of all pigeons are guaranteed by using map and compass. Comparing all the flied positions, it is obvious that the right-centered pigeon's position is the best one. Each pigeon can adjust its flying direction by following this specific pigeon according to Eq.(1), which is expressed by the thick arrows. The thin arrows are its former flying direction, which has relation to $V_i(t-1)\cdot e^{-Rt}$ in Eq.(2). The vector sum of these two arrows is its next flying direction.

In the landmark operator, half of the number of pigeons is decreased by N_p in every generation. However, the pigeons are still far from the destination, and they are unfamiliar the landmarks. Let $X_c(t)$ be the center of some pigeon's position at the *t*-th iteration, and suppose every pigeon can fly straight to the destination. The position updating rule of for pigeon *i* at the *t*-th iteration can be given by

$$N_P(t) = \frac{N_P(t-1)}{2} \tag{3}$$

$$X_c(t) = \frac{\sum X_i(t)\cdot fitness(X_i(t))}{N_P \sum fitness(X_i(t))} \tag{4}$$

$$X_i(t) = X_i(t-1) + rand\cdot(X_c(t) - X_i(t-1)) \tag{5}$$

where $fitness(X_i(t))$ is the quality of the pigeon individual. For the minimum optimization problems, we can choose $fitness(X_i(t)) = 1/(f_{\min}(X_i(t)) + \varepsilon)$. For maximum optimization problems, we can choose $fitness(X_i(t)) = f_{\max}(X_i(t))$. For each individual pigeon, the optimal position of the *Nc*-th iteration can be denoted with X_p, and X_p=min (X_{i1}, X_{i2}, …, X_{iNc}).

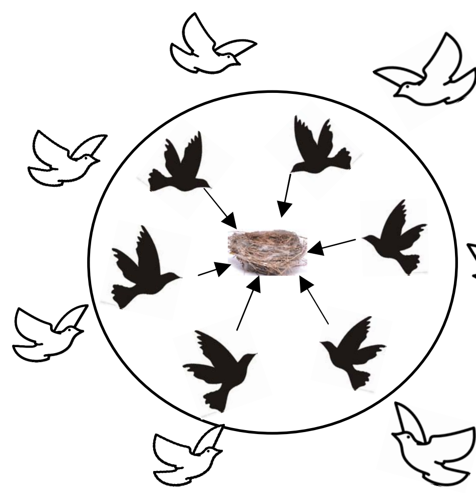

Figure2: Landmark operator model

As shown in Figure 2, the center of all pigeons (the nest in the center of the circle) is their destination in each iteration. Half of all the pigeons (the pigeons out of the circle) that are far from their destination will follow the pigeons that are close to their destination, which also means that two pigeons may be at the same position. The pigeons that are close to their destination (the pigeons in the circle) will fly to their destination very quickly. The procedure of PIO is shown in Figure 3.

III. Prey-Predator Pigeon-Inspired Optimization for PID Controller

A. Prey-Predator strategy concept

Predatory behavior is one of the most common phenomena in nature, and many optimization algorithms are inspired by the Prey-Predator strategy from ecology[9][10]. In nature, predators hunt prey to guarantee their own survival, while the preys need to be able to run away from predators. On the other hand, predators help to control the prey population while creating pressure in the prey population.

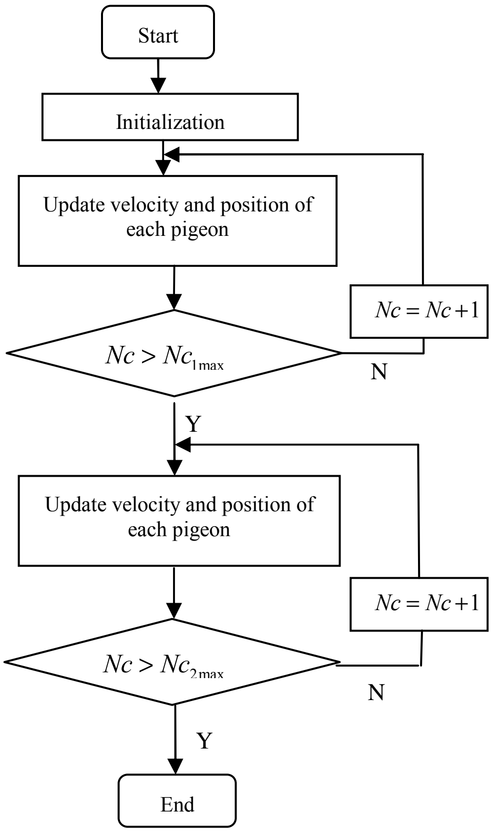

Figure3: The procedure of PIO

In this model, an individual in predator population or prey population represents a solution, each prey in the population can expand or get killed by predators based on its fitness value, and a predator always tries to kill preys with least fitness in its neighborhoods, which represents removing bad solutions in the population. In this paper, the Prey-Predator is used to increase the diversity of the population, the predators are modeled based on the worst solution as Eq.(6) demonstrates:

$$P_{predator} = P_{worst} + \rho\left(1 - t/t_{\max}\right) \tag{6}$$

where $P_{predator}$ is the predator (a possible solution), P_{worst} is the worst solution in the population, t is the current iteration, while $t_{\max}$ is the maximum number of iterations and ρ is the hunting rate. To model the interactions between predators and preys, Eq.(7) and Eq.(8) is also used and this provides the solutions to maintain a distance from the predator

$$P_{K+1} = P_K + \rho e^{-|d|},\ d > 0 \tag{7}$$

$$P_{K+1} = P_K - \rho e^{-|d|},\ d < 0 \tag{8}$$

where d is the distance between the solution and the predator, and k is the current iteration.

B. PID control system

PID controllers are the most common type of controllers by far. It has the advantage of simple structure, strong robustness and high reliability [11][12]. PID control system is widely used in process control and motion control, particularly, in deterministic control system with mathematical model.

The following section gives a further introduction of the principle of PID control system.

In PID control systems, R is input; Y is output; PID position equation is as follow:

$$u(n) = K_p(e(n) + \frac{1}{T_i}\sum_{k=0}^{n} e(k)T + T_d\frac{e(n) - e(n-1)}{T}) \tag{9}$$

$u(n)$ is control quantity; $e(n)$ is error; K_p is proportion coefficient; T_i integration coefficient; T_d is differentiation coefficient; T is the sampling period; PID increment equation is as follows:

$$\begin{aligned} \Delta u(n) &= K_p[(e(n) - e(n-1)) + \frac{T}{T_i}e(n) \\ &+ \frac{T_d}{T}(e(n) - 2e(n-1) + e(n-2))] \end{aligned} \tag{10}$$

where $K_i = K_p / T_i$, $K_d = K_d T_d$. There are 3 parameters K_p, T_i, T_d need to be determined in Eq. (9) and Eq. (10), which is the key to achieve the expected control effect.

Use the time integration of the absolute error value as the minimum objective function. In order to prevent the control energy is too high, add the square of control input to the objective function. The optimal parameter selection is as follows:

$$J = \int_0^{\infty} (k_1 |e(t)| + k_2 u^2(t))dt + k_3 \cdot t_u \tag{11}$$

$e(t)$ represents the system error; $u(t)$ represents the controller output; t_u represents the risetime; k_1 , k_2 , k_3 represent the weight.

In order to avoid overshoot, the punishing function is adopted. Once overshoot appears, it is taken into the optimal index. So the optimal indexes are as follows:

If $e(t) < 0$

$$J = \int_0^{\infty} (k_1 |e(t)| + k_2 u^2(t) + k_4 |e(t)|)dt + k_3 \cdot t_u \tag{12}$$

k_1, k_2 k_3, k_4 represent the weight and $k_4 >> k_1$.

K_p ranges from 0 to 20, K_i and K_d range from 0 to 1.

C. Proposed Prey-Predator Pigeon-Inspired Optimization (PPPIO)

Due to the versatility, flexibility and robustness in solving optimization problems, PIO algorithm has already aroused intense interest. However, there still exist some defects on this algorithm, such as the tendency to converge to local optimal solutions. In order to overcome these defects of PIO algorithm, PPPIO was firstly proposed in this paper, which integrates PIO with the concept of Prey-Predator strategy. After the mutation of each generation, data was renewed by using the Prey-Predator behavior, in order to choose better solutions in to next generation. In this way, our proposed algorithm takes the advantage of the characteristics of the Prey-Predator strategy to make the individuals of sub generations distribute ergodically in the defined space and thus to avoid from the premature of the individuals. Structure diagram of PPPIO for PID controller system design is shown in Figure 4 as follows:

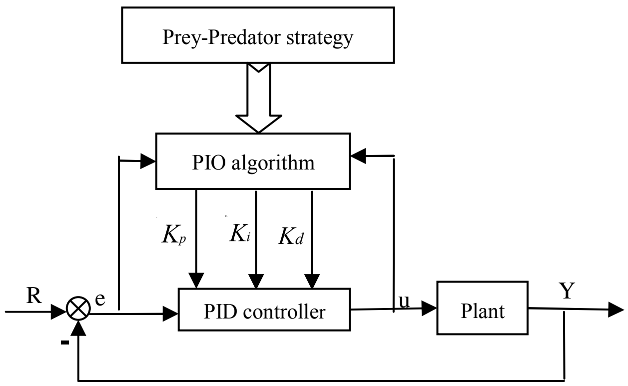

Figure4: PPPIO for PID controller system design structure diagram

PPPIO implementation procedure is as follow:

Step1: According to the feedback control system, initialize the plant.

Step2: Initialize parameters of pigeon-inspired optimization algorithm, such as solution space dimension D, the population size N_p, map and compass factor R, the number of iteration $Nc_1 \max$ and $Nc_2 \max$ for two operators, and $Nc_2 \max > Nc_1 \max$.

Step3: Set each pigeon with a randomized velocity and position. Compare the fitness of each pigeon and find the current best solution.

Step4: Operate map and compass operator. Firstly, we update the velocity and position of every pigeon by using Eq. (1) and (2). Then we compare all the pigeon's fitness and find the new best solution.

Step5: Use Eq.(7) and (8) to make the individuals of sub generations distribute ergodically in the defined space. In this way, we can avoid trapping into local optimal solution.

Step6: If $Nc > Nc_{1\max}$, stop the map and compass operator and operate next operator. Otherwise, go to step 4.

Step7: Rank all pigeons according their fitness values. Half of pigeons whose fitness are low will follow those pigeons with high fitness according to Eq.(3). We then find the center of all pigeons according to Eq.(4), and this center is the desirable destination. All pigeons will fly to the destination by adjusting their flying direction according to Eq.(5). Next, store the best solution parameters.

Step8: If $Nc > Nc_{2\max}$, stop the landmark operator, and output the results. If not, go to step 7.

Figure 5 shows the procedure of PPPIO.

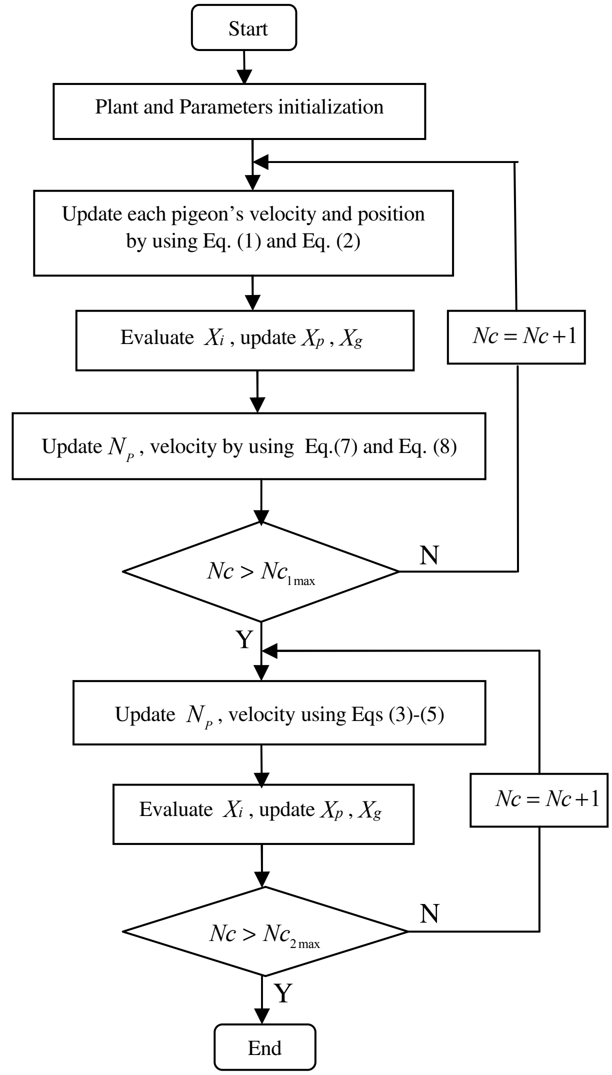

Figure5: The procedure of PPPIO

III. EXPERIMENT AND ANALYSIS

Assuming the plant is a second order system

$$G(s) = \frac{400}{s^2 + 50s} \tag{13}$$

The sampling time is $1ms$, the input command is a step signal. Use PSO, PIO and PPPIO for parameter tuning respectively. Draw the optimization process of function J and the PID step response image after tuning.

Assuming k_1=0.999, k_2=0.001, k_3=2.0, k_4=100, the total particle number is 30, the iteration number is 100.

Table I gives the initial parameters of three algorithms.

TABLE I

: INITIAL PARAMETERS OF THREE ALGORITHMS

Algorithm	Initial parameters
PSO	$c_1=2$, $c_2=2$, $w_{\max}=0.9$, $w_{\min}=0.1$, $v_{\max}=0.001$, $v_{\min}=-v_{\max}$
PIO	$R=6$, $M=30$, $D=3$
PPPIO	$R=6$, $M=30$, $D=3$, $\rho=0.0001$

Figure 6 shows the optimization process of function J, Figure 7 shows the PID response after adjusted. Table II shows the optimal parameters after 100 iterations. Comparative figures and table are as follows:

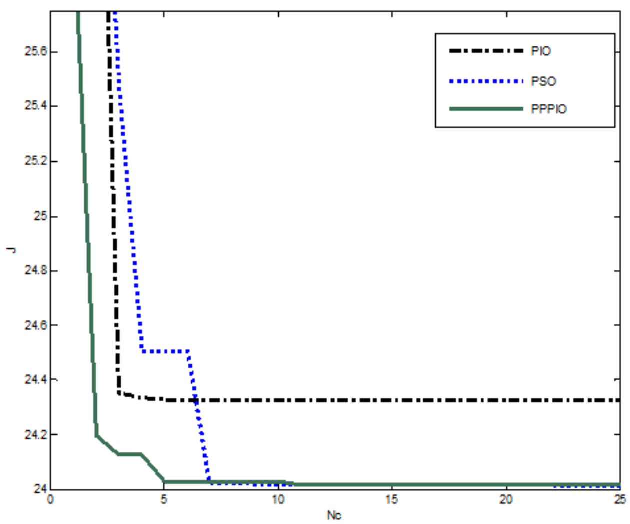

Figure6: The optimization process of function J

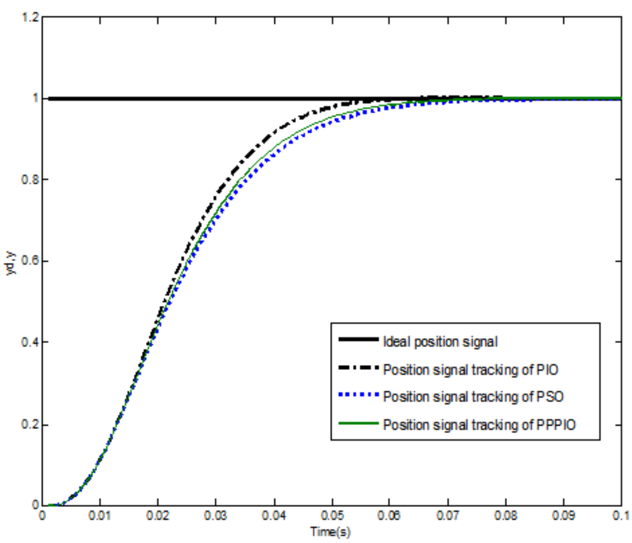

Figure7: PID step response after adjusted

TABLE II
COMPARATIVE RESULTS BY THREE ALGORITHMS

algorithm	PSO	PIO	PPPIO
K_p	20.0000	19.1819	19.9868
K_d	0.2538	0.2625	0.2473
K_i	0.0005	0.6074	0.0006
J	24.0132	24.3659	24.0124

From the above simulation result, PSO algorithm, PIO algorithm and PPPIO algorithm have not much difference between the optimal parameters after tuning the PID. The step responses have the same good quality. However, the convergence speed of PIO algorithm and PPPIO algorithm is obviously faster than PSO algorithm in the optimization process. Sometimes PIO algorithm might trap into a local optimal situation, but PPPIO algorithm can effectively avoid it from trapping into local optimal situation. Thus it can be seen in PID parameter tuning aspect, PPPIO algorithm shows better performance than PIO algorithm and PSO algorithm.

IV. CONCLUSION

In this paper, a new bio-inspired swarm intelligence algorithm PPPIO is applied to the PID parameter adjustment. Compared with PIO algorithm and PSO algorithm, PPPIO algorithm has better global search ability and faster convergence speed, which demonstrates the feasibility of combining Prey-Predator strategy and Pigeon-Inspires Optimization.

ACKNOWLEDGMENT

This work was partially supported by National Key Basic Research Program of China (973 Project) under grant #2014CB046401, and Aeronautical Foundation of China under grant #20135851042.

REFERENCES

[1] H. Duan, D. Wang, X. Huang J. Zhu, "Research and realization on parameters optimization of PID controller based on ant colony algorithm," *Engineering juournal of Wuhan University*, vol. 37, no. 5, pp. 97-100, 2004

[2] E. Bonabeau, M. Dorigo, G. Theraulaz, "Inspiration for optimization from social insect behavior," *Nature,* vol .406, no. 6, pp. 39-42, 2000

[3] A. Colorni, M. Dorigo, V. Maniezzo, et al, "Distributed optimization by ant colonies," *Proceedings of the 1st European conference on artificial life,* Paris, pp. 134-142, 1991.

[4] R. Eberhart, J. Kennedy, "A new optimizer using particle swarm theory," *Proceedings of the 6th International Symposium on Micro-Machine and Human Science,* Nagoya, pp. 39-43, 1995.

[5] H. Duan, P. Li, "Bio-inspired computation in unmanned aerial vehicles," *Springer-Verlag,* Berlin, Heidelberg, 2014.

[6] H. Duan, S. Shao, B. Su, L. Zhang, "New development thoughts on the Bio-inspired intelligence based control for unmanned combat aerial vehicle," *Science China technological sciences,* vol. 53, no.8, pp. 2025–2031, 2010.

[7] H. Duan, P. Qiao, "Pigeon-inspired optimization: A new swarm intelligence optimizer for air robot path planning," *International Journal of intelligent Computing and Cybernetics,* vol. 7, no. 1, 2014

[8] C. Mora, C. Davison, J. Wild, M. Walker, "Magentorecrption and its trigeminal medition in the homing pigeon," *Nature,* vol. 432, no. 7016, pp. 508-511, 2004.
[9] W. Zhu, H. Duan, "Chaotic predator-prey biogeography-based optimization approach for UCAV path planning," *Aerospace science and technology,* pp. 153-161, 2014.
[10] H. Duan, Y. Yu, Z. Zhao, "Parameters identification of UCAVs fight control system based on predator-prey particle swarm optimization," *Science China Information Science,* vol. 56, no.1, pp. 012202, 2013.
[11] K. Astrom, T. Hagglund, C. Hang, W. Ho, "Automatic tuning and adapation for PID controllers-A survey," *Control Engineering Practice,* vol. 1, no. 4, pp. 699-714, 1993.
[12] W. Wang, J. Zhang, T. Chai, "A survey of advanced PID parameter tuning methods," *Acta Automatic Sinica ,*vol. 26, no. 3, pp. 347-355, 2000.