Proceedings of 2012 IEEE
International Conference on Mechatronics and Automation
August 5 - 8, Chengdu, China

Tuning of PID Controller Based on Fruit Fly Optimization Algorithm

Jiuqi Han and Peng Wang
Research Center of Precision Sensing and Control
Institute of Automation, Chinese Academy of Sciences
Beijing, 100190, China
{jiuqi.han & peng.wang}@ia.ac.cn

Xin Yang
State Key Lab. of Management & Control for Complex Systems
Institute of Automation, Chinese Academy of Sciences
Beijing, 100190, China
xin.yang@ia.ac.cn

***Abstract* - The Proportional - Integral - Derivative (PID) controllers are one of the most popular controllers used in industry because of their remarkable effectiveness, simplicity of implementation and broad applicability. PID tuning is the key issue in the design of PID controllers and most of the tuning processes are implemented manually resulting in difficulty and time consuming. To enhance the capabilities of traditional PID parameters tuning techniques, modern heuristics approaches, such as Genetic Algorithm (GA) and Particle Swarm Optimization (PSO), are employed recent years. In this paper, a novel tuning method based on Fruit Fly Optimization Algorithm (FOA) is proposed to optimize PID controller parameters. Each fruit fly's position represents a candidate solution for PID parameters. When the fruit fly swarm flies towards one location, it is treated as the evolution of each iterative swarm. After hundreds of iteration, the tuning results - the best PID controller parameters can be obtained. The main advantages of the proposed method include ease of implementation, stable convergence characteristic, large searching range, ease of transformation of such concept into program code and ease of understanding. Simulation results demonstrate that the FOA-Based optimized PID (FOA - PID) controller is with the capability of providing satisfactory closed - loop performance.**

Index Terms –PID tuning, Fruit Fly Optimization Algorithm (FOA), Optimization.

I. INTRODUCTION

Proportional-Integral-Derivative (PID) controller is a generic control loop feedback mechanism widely used in industrial control systems, especially for systems with accurate mathematical models. The PID controller calculation involves three separate parameters: proportional, integral and derivative values [1]. The proportional value calculates the value of the current error, the integral value determines the result of the sum of recent errors, and the derivative value determines the reaction based on the rate at which the error has been changing. The weighted sum of these three actions is used to be imported into the controlled system.

The key issue for PID controllers is the accurate and efficient tuning of parameters. In practice, the controlled systems usually have some features, such as nonlinearity, time-variability and time delay, which make controller parameters tuning more complex. Moreover, in some cases, system parameters and even system structures can vary with time and environment [2]. Thus, the goal of PID controller tuning is to determine parameters that meet the closed-loop system performance specifications over a wide range of operating conditions [3]. Among the conventional PID tuning methods, the Ziegler-Nichols (ZN) method may be the most well known technique. For a wide range of practical processes, this tuning approach works quite well. However, sometimes it does not provide good tuning and tends to produce a big overshoot [4]. To enhance the capabilities of traditional PID parameters tuning techniques, several intelligent approaches have been suggested, such as those based on Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO).

Fruit Fly Optimization Algorithm (FOA) was proposed by Pan in 2011 [5], which is a stochastic searching algorithm based on principles of natural selection. After firstly introduced, FOA has been used to solve optimization and machine-learning problems. Pan used it to solve the financial distress problem in [5], and Lin used it to do the analysis of logistics service satisfaction in web auction in [6].

This paper attempts to develop a new PID tuning method based on FOA. The position of each fruit fly represents a set of PID controller parameters. After several iterations, the fruit fly swarm will arrive at a location, which has the best value of the smell concentration. This location can be used as the optimal set of PID controller parameters. The experimental results demonstrate the effectiveness of the FOA-PID in control engineering applications.

The main structure of this article is as follows: The first section introduces the research motivation and background of this article. The second section gives the PID tuning process based on optimization methods. In section three, FOA is introduced and FOA-PID is proposed. Experimental results are presented in section four and conclusion is in section five.

II. TUNING OF PID USING OPTIMIZATION METHODS

The tuning of PID controller parameters has been commonly researched and discussed for many years. The tuning process usually needs lots of human power, efforts and time, and in the worst case, the bad tuning parameters could lead to poor controlling performance or even crash in control system. As mentioned earlier, the main aim of PID tuning is to meet the closed-loop system performance specifications, such as peak overshoot, settling time, rising time, and the robust performance of the control loop over different conditions. However, it is often difficult to simultaneously achieve all of these desirable qualities practically [1]. For example, if the PID controller is adjusted to provide a good transient response to a set point change, it usually results in a bad steady state

978-1-4673-1278-3/12/$31.00 ©2012 IEEE

characteristic. On the other hand, if the control system is made to a small steady state error by choosing conservative values for the PID controller, it may result in a slow closed loop response to a set point change. To solve this problem, more and more optimization methods have been used to tune PID parameters. Compared with conventional Ziegler-Nichols (Z-N) method, the optimization methods, such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO), have been proved their excellence in giving better results by improving the steady state characteristics and performance indices.

Particle Swarm Optimization (PSO) is one of the optimization and evolutionary computation techniques. The basic PSO is developed from researches on swarm such as fish schooling and bird flocking [7]. It was firstly introduced in 1995 in [8], and plenty of improved versions were proposed [9]. Ant Colony Optimization (ACO) is a meta-heuristic algorithm for the approximate solution of combinatorial optimization problems which was inspired by the foraging behavior of real ant colonies [10].

Genetic Algorithm (GA) is a stochastic global search method that mimics the process of natural evolution [11], which is one of the important optimization methods. John Holland formally introduced this method in the United States in the 1970s at the University of Michigan [12]. GA includes three major operators: selection, crossover, and mutation. The technique of GA is proposed as a means of auto-tuning PID controllers by A. Jones and P. Oliveira [13]. The continuous genetic algorithm was proposed to improve the operation time and efficiency for PID adjustment [14][15]. GA uses the chromosomes fitness value to create a new population consisting of the fittest members, and each chromosome consists of three separate strings constituting a P , I and D term. Fig. 1 shows the flowchart of the GA based PID tuning (GA-PID) method.

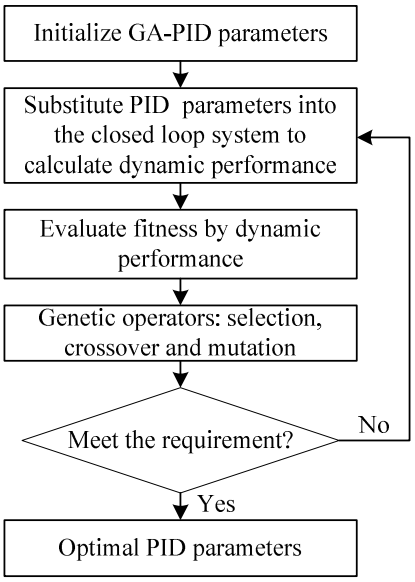

Fig. 1 Flowchart of GA based PID tuning

III. TUNING OF PID BASED ON FRUIT FLY OPTIMIZATION ALGORITHM

A. Overview of Fruit Fly Optimization Algorithm (FOA)

The Fruit Fly Optimization Algorithm (FOA) is a new method for finding global optimization based on the food finding behavior of the fruit fly [5]. The fruit fly itself is superior to other species in sensing and perception, especially in olfactory and vision. The olfactory organ of the fruit fly can find all kinds of scents floating in the air, and it can even smell food source from 40km away. Then, after it gets close to the food location, it can also use its sensitive vision to find food and the company's flocking location, and fly towards the right direction too.

Fig. 2 shows the illustration of the group iterative food searching of fruit fly [6]. FOA randomly generates a fruit fly swarm's initial location at first. Then, each fruit fly is assigned a direction and distance for their movement. Since the food location is unknown, the distance to the origin is estimated. After they reach the new positions, we can find the best position with the results of calculation and judgment. Repeating this process and we could get the optimal solution finally. Fruit fly's foraging characteristics have been summarized and programmed into the following steps as shown in Fig. 3.

B. Tuning of PID Based-on FOA (FOA-PID)

In this paper, we introduce FOA to tune PID parameters (K_P , K_I , K_D), and propose a FOA-PID algorithm. In FOA-PID algorithm, initial swarm of fruit flies is firstly produced in search space represented by matrix, and then through iteration, the optimal solutions can be obtained. Each fruit fly's position represents a candidate solution for PID parameters. When the fruit fly swarm flies towards one location, it is treated as the evolution of each iterative swarm. After hundreds of iteration, the tuning results – the best PID controller parameters can be obtained, and they can yield the result in minimization of judgment and a fast system response.

Fig. 4 shows the block diagram of a controlled system with unit negative feedback based on the proposed FOA-PID algorithm.

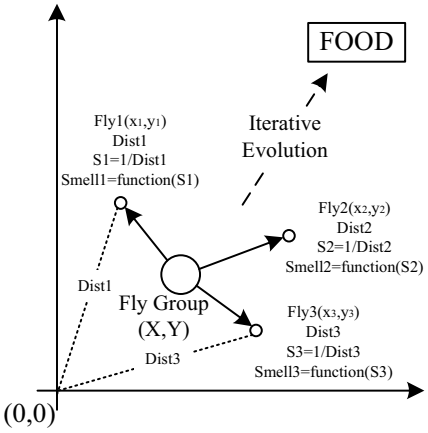

Fig. 2 Illustration of the group iterative food searching of fruit fly

FRUIT FLY OPTIMIZATION ALGORITHM

Input:

★Number of iteration G

★Population size of the fruit fly swarm M

Initialization:

★Randomly generate a fruit fly swarm's initial location $X(1)$, $Y(1)$

For $i=1:1:M$

★Randomly assign each fruit fly a direction and distance

$X_i = X(1) + Random\ \ value$, $Y_i = Y(1) + Random\ \ value$

★Estimate the distance (Dist) to the origin and the judged value of smell concentration (S)

$$Dist_i = \sqrt{\left(X_i^2 + Y_i^2\right)}; S_i = 1/Dist_i$$

★Calculate $Smell_i = Function(S_i)$ and find out the best smell concentration

$bestSmell = \max(Smell)$

★Set the best smell concentration $Smellbest = bestSmell$

Searching:

For $k=1:1:G$

For $i=1:1:M$

★ Randomly assign $X_i = X(k) + Random\ \ value$, $Y_i = Y(k) + Random\ \ value$

★ Estimate $Dist_i = \sqrt{\left(X_i^2 + Y_i^2\right)}; S_i = 1/Dist_i$

★ Calculate $Smell_i = Function(S_i)$ and find out the best smell concentration

$[bestSmell\ \ bestIndex] = \max(Smell)$

★ If $bestSmell$ is better than $Smellbest$

update $Smellbest = bestSmell$

set $X(k+1) = X(bestIndex), Y(k+1) = Y(bestIndex)$

Output: $Smellbest$, $X = X(G+1)$, $Y = Y(G+1)$

Fig. 3 Fruit Fly Optimization Algorithm (FOA)

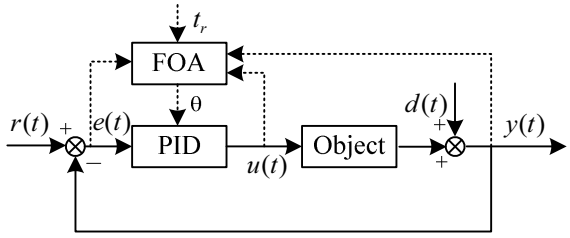

Fig. 4 Block diagram of FOA-PID controller tuning

The variable $e(t) = r(t) - y(t)$ represents the tracking error, i.e., the difference between the desired input value $r(t)$ and the actual output value $y(t)$. The error signal $e(t)$ will be sent to the PID controller, and the controller computes the proportional, the integral and the derivative of this error signal. The PID controller transfer function is

$$u(t) = K_P\left[e(t) + \frac{1}{T_I}\int_0^t e(t)dt + \frac{T_D de(t)}{dt}\right] \quad (1)$$

where K_P is the proportional gain, T_I is the integral time coefficient, T_D is the derivative time coefficient, $e(t)$ is the tracking error, $K_I = K_P/T_I$ is the integral gain, and $K_D = K_P T_D$ is the derivative gain. With the PID controller and the object, the dynamic performance of the controlled system can be obtained. Therefore, we define the judgment function as follows

$$J = \begin{cases} \int_0^\infty \left(w_1|e(t)| + w_2 u^2(t)\right)dt + w_3 t_r, \text{if } ey(t) > 0 \\ \int_0^\infty \left(w_1|e(t)| + w_2 u^2(t) + w_4|ey(t)|\right)dt + w_3 t_r, \text{otherwise} \end{cases} \quad (2)$$

where w_1, w_2, w_3 and w_4 are the combining weights, t_r is the rising time of the step response of the closed-loop system. If $\max(y(t)) < r(t)$, t_r describes the interval in which $y(t)$ increases from $0.1\times r(t)$ to $0.9\times r(t)$. Otherwise, t_r is the time when $y(t)$ reaches $r(t)$ firstly. When $ey(t) < 0$, it represents overshoot of $y(t)$, so it must be constricted. Here we assume $w_4 \gg w_1$ so that the overshoot $ey(t)$ is constricted more than the tracking error $e(t)$ to improve the dynamic behaviors.

We defined three controller parameters K_P, K_I and K_D to compose a position of an individual fruit fly $\theta = [K_P \quad K_I \quad K_D]$, hence, there are three members in an individual. These members are assigned as positive real values. If the population size of the fruit fly swarm is M, then the dimension of a swarm is $M\times 3$. The judgment can be calculated by the dynamic performance of the closed-loop system directly. We conclude the FOA-PID process in Fig. 5.

FOA-PID

Input:

★Number of iteration G

★Population size of the fruit fly swarm M

Initialization:

★Randomly generate a fruit fly swarm's initial location

$\theta(1) = [K_P(1) \quad K_I(1) \quad K_D(1)]$

For $i=1:1:M$

★Randomly assign $K_{Pi} = K_P(1) + Random\ \ value$, $K_{Ii} = K_I(1) + Random\ \ value$ and $K_{Di} = K_D(1) + Random\ \ value$

★Calculate the dynamic performance of the controlled system with M different PID controllers

★Use $r(t)$, $y(t)$, $y(t-1)$, $u(t)$ and t_r to calculate the judgment J

★Find out the best judgment value $bestJ = \min(J)$ and set $Jbest = bestJ$

Searching:

For $k=1:1:G$

For $i=1:1:M$

★Randomly assign $K_{Pi} = K_P(k) + Random\ \ value$, $K_{Ii} = K_I(k) + Random\ \ value$ and $K_{Di} = K_D(k) + Random\ \ value$

★Calculate the dynamic performance of the controlled system

★Calculate the judgment J

★Find out the best J, $[bestJ\ bestIndex] = \min(J)$

★If J is better than the $Jbest$ and update $Jbest = bestJ$

set $K_P(k+1) = K_P(bestIndex), K_I(k+1) = K_I(bestIndex), K_D(k+1) = K_D(bestIndex)$

Output: $Jbest$, $K_P = K_P(G+1), K_I = K_I(G+1), K_D = K_D(G+1)$

Fig. 5 Tuning of PID Controller Based-on Fruit Fly Optimization Algorithm (FOA-PID)

IV. Simulation Results

A. Typical Objects in Simulation Experiments

To verify the efficiency of the proposed FOA-PID algorithm, comparison experiments between FOA-PID and other optimization-based tuning methods, such as GA-PID are provided.

The tuning methods are tested using different kinds of control systems, such as the second-order system in (3) and the third-order system in (4).

Object 1: the second-order system. The transfer function is:

$$G_1(s)=\frac{400}{s^2+50s}. \tag{3}$$

Object 2: the third-order system. The transfer function is:

$$G_2(s)=\frac{s+1}{s^3+2s^2+2s+1}. \tag{4}$$

B. Initialization of Parameters for FOA-PID

In the experiments, the following parameters are used to verify the performance of the FOA-PID controller.

The tuning parameters are K_P, K_I and K_D and the population size is $M=40$. The numbers of iteration and the sampling time are $G=100$, $ts=0.001s$ for object 1 and $G=500$, $ts=0.02s$ for object 2 respectively. The random initialization fruit fly swarm location zones for object 1 are $[0,10]$, $[0,1]$ and $[0,1]$. And the random initialization fruit fly swarm location zones for object 2 are $[0,5]$, $[0,5]$ and $[0,5]$.

C. Comparison of GA-PID and FOA-PID

In order to demonstrate the advantages of the proposed FOA-PID, we also implement the GA-PID on the same systems, and the following parameters have been used in GA-PID.

The length of coding is 10 (Gray Code), the crossover rate is $p_c=0.9$, the mutation rate is $p_m=0.01$, the initial ranges of parameters of PID controllers are $[0,10]$, $[0,1]$, $[0,1]$ for object 1 and $[0,5]$, $[0,5]$, $[0,5]$ for object 2.

In addition, the members of every individual, population size, the numbers of iteration and the sampling time are set as the same with FOA-PID.

When the system has no overshoot, the weights in (2) are $w_1=0.999$, $w_2=0.001$ and $w_3=2$ respectively. Otherwise, the weights are $w_1=0.999$, $w_2=0.001$, $w_3=2$ and $w_4=100$ respectively. The simulation programs were run on the Matlab 7.11.0 (R2010b).

The simulation results that show the optimal solution are summarized in Table I (object 1) and Table II (object 2). Table I and Table II show that FOA-PID has much larger ranges of tuning parameters, although the initial ranges are the same with GA-PID. Fig. 6 and Fig. 8 show the trend of the values of J in different iterations. It is apparent that the value of J in FOA-PID is less easily to fall into the local minimum than that of GA-PID. Fig. 6 shows that FOA-PID gets the best value after the iteration 50, but GA falls into the local minimum. Therefore, FOA-PID controller can search the optimal parameters more quickly and efficiently than GA-PID. From Fig. 6 and Fig. 8, we can see that the final results of evolution with FOA are better than GA.

Table II shows that the settling time ($\sigma=2\%$) of FOA-PID is less than that of GA-PID (0.6s vs 1.72s) and the best J of FOA-PID is also less than that of GA-PID (22.8780 vs 39.8601). From Fig.7 and Fig. 9, we can see that the FOA-PID controller can create very perfect step response of the system, indicating that the FOA is better than GA for parameters optimization of PID controller.

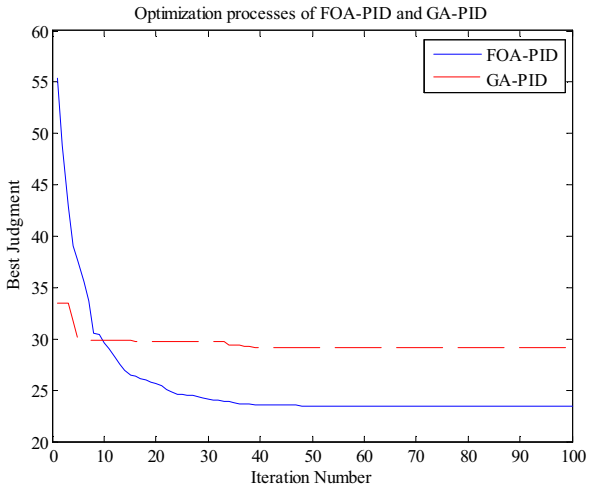

Fig. 6 The trend lines of the values of the judgment for the object 1

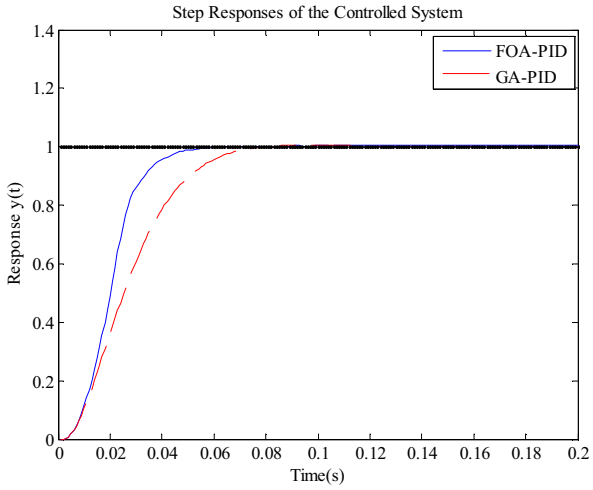

Fig. 7 The step responses of the system with optimal PID controllers using FOA and GA for the object 1

TABLE I
RESULTS OF SIMULATION ON OBJECT 1

G1	K_P	K_I	K_D	σ	T_S (s)	BestJ
FOA-PID	171.22	25.179	1.5245	0.01%	0.045	23.4059
GA-PID	9.8338	0	0.1447	0.45%	0.066	29.1328

TABLE II

RESULTS OF SIMULATION ON OBJECT 2

G2	K_P	K_I	K_D	σ	T_S (s)	BestJ
FOA-PID	80.712	3.8895	13.748	0.05%	0.6	22.8780
GA-PID	5	1.3050	2.7859	0.93%	1.72	39.8601

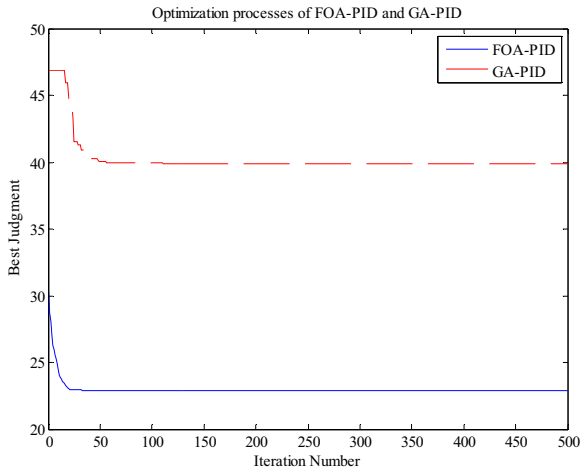

Fig. 8 The trend lines of the values of the judgment for the object 2

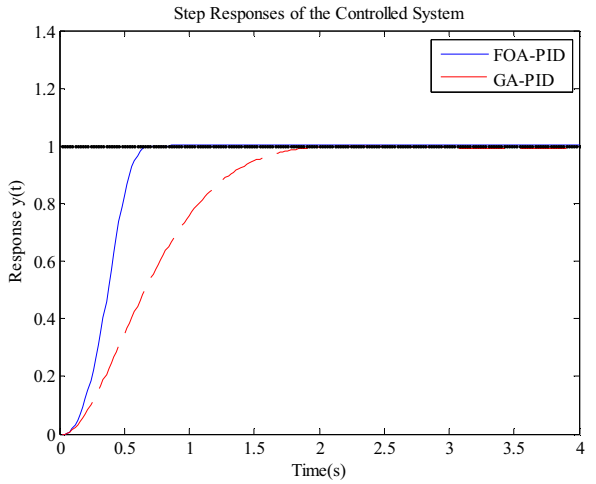

Fig. 9 The step responses of the system with optimal PID controllers using FOA and GA for the object 2

V. CONCLUSION

The major contribution of this article is to present a novel method for determining the PID controller parameters based on Fruit Fly Optimization Algorithm (FOA), namely FOA-PID. This method not only has the feature of being easy to understand, but also is easy to be written into program code. The experimental results show that the FOA-PID can avoid the shortcoming of premature convergence and can obtain higher quality solution than GA-PID. Therefore, the designed FOA-PID controller shows superior performance over GA-PID, in terms of the settling time and overshoot. That is to say that FOA can solve the searching and tuning problems of PID controller parameters more easily and quickly than GA. However, the initial swarm location, the fly direction and distance zones are assigned as fixed values by experience. Thus, more improved work should be done in the future.

ACKNOWLEDGMENT

This work was partly supported by the NNSF (National Natural Science Foundation) of China under the grant 61100098, and the Knowledge Innovation Program of the Chinese Academy of Sciences under the grant GY-110502.

REFERENCES

[1] B. Nagaraj and P. Vijayakumar, "A comparative study of PID controller tuning using GA, EP, PSO and ACO," *Journal of Automation, Mobile Robotics & Intelligent Systems*, vol. 5, no. 2, pp. 42-48, 2011.

[2] J. Zhang, J. Zhuang, H. Du, and S. Wang, "Self-organizing genetic algorithm based tuning of PID controllers," *Information Sciences*, vol. 179, issue 7, pp. 1007-1018, 2009.

[3] S. Bassi, M. Mishra, and E. Omizegba, "Automatic tuning of proportional-integral-derivative (PID) controller using particle swarm optimization (PSO) algorithm," *International Journal of Artificial Intelligence & Applications (IJAIA)*, vol. 2, no. 4, pp. 25-32, 2011.

[4] M. Solihin, L. Tack, and M. Kean, "Tuning of PID controller using particle swarm optimization (PSO)," *Proceeding of the International conference on Advanced Science, Engineering and Information Technology*, pp. 458-461, January 2011.

[5] W. Pan, "A new fruit fly optimization algorithm: taking the financial distress model as an example," *Knowledge-Based Systems*, vol. 26, pp. 69-74, 2012.

[6] S. Lin, "Analysis of service satisfaction in web auction logistics service using a combination of fruit fly optimization algorithm and general regression neural network," *Neural Computing & Applications*, pp. 1-9, 2011.Available at: http://dx.doi.org/10.1007/s00521-011-0769-1.

[7] C. Ou and W. Lin, "Comparison between PSO and GA for parameters optimization of PID controller," *Proceedings of the 2006 IEEE International Conference on Mechatronics and Automation*, pp. 2471-2475, June 2006.

[8] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proceedings of IEEE International Conference on Neural Networks*, vol. 4, pp. 1942-1948, 1995.

[9] Y. Shi and R. Eberhart, "A modified particle swarm optimizier," *Proceedings of IEEE International Conference on Evolutionary Computation*, pp. 69-73, May 1998.

[10]S. Girirajkumar, K. Ramkumar, and O. Sarma, "Real time application of ants colony optimization," *International Journal of Computer Applications*, vol. 3, no. 8, pp. 34-46, 2010.

[11]I. Griffin and J. Bruton, "On-line PID controller tuning using genetic algorithms", Available at: http://elm.eeng.dcu.ie/~brutonj/Reports/IGriffin-MEng_03.pdf.

[12]J. Holland, "Genetic algorithms", *Scientific American*, vol. 267, no. 1, pp. 66-72, 1992.

[13]A. Jones and P. Oliveira, "Genetic auto-tuning of PID controllers", *Proceedings of International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*, pp. 141-145, September 1995.

[14]J. Park and Y. Choi, "An on-line PID control scheme for unknown nonlinear dynamic systems using evolution strategy", *Proceedings of IEEE International Conference on Evolutionary Computation*, pp. 759-763, May 1996.

[15]S. Chen, "Particle swarm optimization for PID controllers with robust testing", *Proceedings of International Conference on Machine Learning and Cybernetics*, vol. 2, pp. 956-961, August 2007.