

# Intelligent Tuning of PID Controller With Disturbance Rejection Using Bacterial Foraging

Dong Hwa Kim, Jae Hoon Cho

Dept. of Instrumentation and Control Eng., Hanbat National University,  
16-1 San Duckmyong-Dong Yuseong-Gu, Daejeon City, Korea, 305-719.

E-mail: [kimdh@hanbat.ac.kr](mailto:kimdh@hanbat.ac.kr), Homepage: [ial.hanbat.ac.kr](http://ial.hanbat.ac.kr)

Tel: +82-42-821-1170, Fax: +82-821-1164

## Abstract

In this paper, design approach of PID controller with rejection function against external disturbance in motor control system is proposed using bacterial foraging based optimal algorithm. Up to the present time, PID Controller has been used to operate for AC motor drive because of its implementational advantages in practice and simple structure. However, it is not easy to achieve an optimal PID gain with no experience, since the gain of the PID controller has to be manually tuned by trial and error in the industrial system with disturbance. To design disturbance rejection tuning, disturbance rejection conditions based on  $H_\infty$  are illustrated and the performance of response based on the bacterial foraging is computed for the designed PID controller as ITSE (Integral of time weighted squared error). Hence, parameters of PID controller are selected by bacterial foraging based optimal algorithm to obtain the required response.

**Key Words:** PID control; Disturbance control; Bacterial algorithm, Optimal algorithm; Motor control.

## 1. Introduction

A Proportional – Integral – Derivative (PID) controller has been widely used in the most industrial processes despite continual advances in control theory. This is not only due to the simple structure which is theoretically easy to understand but also to the fact that the tuning technique provides adequate performance in the vast majority of applications. However, it cannot effectively control such a complicated or fast running system such as motor control system, since the response of a plant depends on only the three parameters (P, I, and D) and its gain has to be manually tuned by trial and error in the industrial world. Most of the PID tuning rules developed in the past years use the conventional method such as frequency-response methods [1]. This method needs a highly technical experience to apply since they provide simple tuning formulae to determine the PID controller parameters. In case of the Ziegler-Nichols rule tuning technique, it often leads to a rather oscillatory response to set-point changes as the following system [2]:

- (a) Plant which the system has non-linearities such as directionally dependent actuator and plant dynamics
- (b) System that various uncertainties, such as modeling error and external disturbances, are involved in the system.

As a result of these difficulties, the PID controllers are rarely tuned optimally and the engineers need to research for a highly tuning technology.

To improve the performance of PID controller tuning for processes with changing dynamic properties, several tuning strategies for PID controller have been proposed, for example, automatic tuning PID, adaptive PID, and intelligent controller. These controllers have recalibration features to cope with little a priori knowledge and significant changes in the process dynamics [4].

However, the PID controller parameters are still computed using the classic tuning formulae and, as noted above, these do not provide good control performance in all situations, for example, for unstable systems with time delay. De Paor used the Modified Smith Predictors to cope with unstable and integrating process with long time delay.

In order to provide consistent, reliable, safe and optimal solution to industrial control problems as described above, many approaches for PID control schemes and tuning techniques have been presented. These schemes generally consist of four basic parts: model estimation, desired system specifications, optimal tuning mechanism and an online PID controller.

Over the past 50 years, several methods for determining PID controller parameters have been developed for stable processes that are suitable for auto-tuning and adaptive control [1-10]. Some employ information about open-loop step response, for example, the Coon-Cohen reaction curve method; other methods use knowledge of the Nyquist curve, for example, the Ziegler-Nichols frequency-response method [3]. However, these tuning methods use only a small amount of information about the dynamic behavior of the system, and often do not provide good tuning. It is known that gain and phase margins (GPM) have served as important measures of robustness [3,7,8]. The phase margin is related to the damping of the system from classical control theories, and therefore also serves as a performance measurement. Their solutions are normally obtained numerically or graphically by trial-and-error use of Bode plots.

Despite the fact that many PID tuning methods are available for achieving the specified GPM, they can be divided into two categories. Firstly, approximation of  $\tan^{-1}$  function is adopt to simplify the problem, but they are only applicable to the simple models [4,9]. Secondly, inverse function mapping was done by fuzzy neural network (FNN). The FNN identify the relationship between GPM and PID controllers, which is

available for general linear system [5]. For solving the problem, the first obstacle is the difficulty in finding the stabilizing region of PID controllers. The solution is a necessary first step to any rational design of PID controllers based on GPM.

On the other hand, since natural selection of bacterial foraging tends to eliminate animals with poor foraging strategies for locating, handling, and ingesting food, optimization models can be provided for social foraging where groups of parameters communicate to cooperatively forage in engineering.

In this paper, an intelligent tuning method of PID controller by bacterial foraging based optimal algorithm is suggested for robust control with disturbance rejection function on control system of motor control loop.

## 2. PID Controller Tuning With Disturbance Rejection Function

### A. Condition for Disturbance Rejection

In Fig. 1, the disturbance rejection constraint can be given by [7,8]

$$\max_{d(t) \in D} \frac{\|Y\|}{\|d\|} = \left\| \frac{w(s)}{1 + K(s,c)G(s)} \right\|_{\infty} < \delta. \quad (1)$$

Here,  $\delta$  is constant defining by the desired rejection level and  $\|\bullet\|_{\infty}$  denotes the  $H_{\infty}$ -norm, which is defined as

$$\|G(s)\|_{\infty} = \max_{\omega \in [0, \infty)} |G(j\omega)|. \quad (2)$$

The disturbance rejection constraint becomes

$$\begin{aligned} & \left\| \frac{w(s)}{1 + K(s,c)G(s)} \right\|_{\infty} = \\ & \max_{\omega \in [0, \infty)} \left( \frac{w(j\omega)w(-j\omega)}{1 + K(j\omega,c)G(j\omega,c)K(-j\omega,c)G(-j\omega,c)} \right)^{0.5} \\ & = \max_{\omega \in [0, \infty)} (\sigma(\omega,c))^{0.5} \end{aligned} \quad (3)$$

The controller  $K(s, c)$  is written as

$$K(s, c) = c_1 + \frac{c_2}{s} + c_3s \quad (4)$$

The vector  $c$  of the controller parameter is given by

$$c = [c_1, c_2, c_3]^T \quad (5)$$

Hence, the condition for disturbance rejection is given as

$$\max_{\omega \in [0, \infty)} (\sigma(\omega,c))^{0.5} < \delta.$$

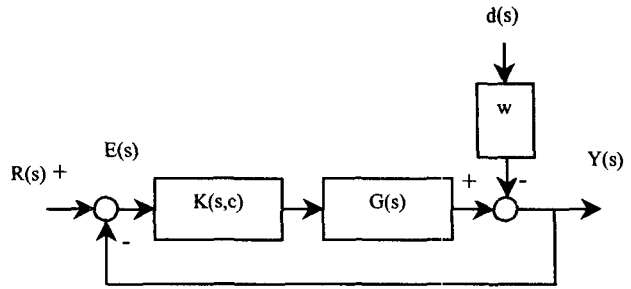


Fig. 1. Control system with disturbance.

### A. Performance Index for Disturbance Rejection Controller Design

The performance index defined as ITSE (Integral of the Time-Weighted Square of the Error) is written by

$$PI = \int_0^{\infty} t(E(t))^2 dt, \quad (6a)$$

$$E(s) = \frac{B(s)}{A(s)} = \frac{\sum_{j=0}^m b_j s^{m-1}}{\sum_{i=0}^n a_i s^{n-1}}. \quad (6b)$$

Because  $E(s)$  contains the parameters of the controller ( $c$ ) and plant, the value of performance index,  $PI$  for a system of  $n$ th order can be minimized by adjusting the vector  $c$  as follows [7]:

$$\min_c PI(c) \quad (7)$$

The optimal tuning proposed in this paper is to find the vector  $c$ , such that the ITSE performance index,  $PI(c)$  is a minimum using bacterial algorithm and the constraint  $\max_{\omega \in [0, \infty)} (\sigma(\omega,c))^{0.5} < \delta$  is satisfied through real coded bacterial algorithms.

## 3. Behavior Characteristics and Modeling of Bacteria Foraging

Since selection behavior of bacteria tends to eliminate animals with poor foraging strategies and favor the propagation of genes of those animals that have successful foraging strategies, they can be applied to have an optimal solution through methods for locating, handling, and ingesting food. After many generations, a foraging animal takes actions to maximize the energy obtained per unit time spent foraging. That is, poor foraging strategies are either eliminated or shaped into good ones. Optimization models are also valid for social foraging where groups of animals communicate to cooperatively forage, in the face of constraints presented by its own physiology such as, sensing and cognitive capabilities and environment.

As mentioned in the above, foraging can be modeled as an

optimization process, which sometimes operates in swarms, and the relevance of these areas to optimization.

Foraging theory is described in Refs. 5-7. Foraging behavior of bacteria can be found using, for instance, dynamic programming. Search and optimal foraging decision-making of animals can be used to engineering. Selection behavior or bacteria forage as individuals and others forage as groups. While to perform social foraging an animal needs communication capabilities, it can gain advantages in that it can exploit essentially the sensing capabilities of the group, the group can gang-up on large prey, individuals can obtain protection from predators while in a group, and in a certain sense the group can forage with a type of collective intelligence. This paper describes the optimal parameter selection of a PID controller using bacteria foraging.

*A. Over view of Chemotactic Behavior of E. coli.*

This paper considers the foraging behavior of E. coli, which is a common type of bacteria as in reference 4-5. Its behavior to move comes from a set of up to six rigid 100–200 rps spinning flagella, each driven as a biological motor. An E. coli bacterium alternates between running and tumbling. Running speed is 10–20 μm/sec, but they cannot swim straight. When we can summarize the chemotactic actions of bacteria as the following description:

- If in neutral medium, alternate tumbles and runs, its action is having search.
- If swimming up a nutrient gradient (or out of noxious substances), swim longer (climb up nutrient gradient or down noxious gradient), its behavior seeks increasingly favorable environments.
- If swimming down a nutrient gradient (or up noxious substance gradient), then search action is avoiding unfavorable environments.

So, it can climb up nutrient hills and at the same time avoid noxious substances. The sensors it needs for optimal resolution are receptor proteins which are very sensitive and high gain. That is, a small change in the concentration of nutrients can cause a significant change in behavior. This is probably the best-understood sensory and decision-making system in biology.

Mutations in E. coli affect the reproductive efficiency at different temperatures, and occur at a rate of about 10<sup>-7</sup> per gene and per generation. E. coli occasionally engages in a conjugation that affects the characteristics of a population of bacteria. Since there are many types of taxes that are used by bacteria such as, aerotaxis (it are attracted to oxygen), light (phototaxis), temperature (thermotaxis), magnetotaxis (it it can be affected by magnetic lines of flux. Some bacteria can change their shape and number of flagella which is based on the medium to reconfigure in order to ensure efficient foraging in a variety of media. Bacteria can form intricate stable spatio-temporal patterns in certain semisolid nutrient substances. They can eat radially their way through a medium if placed together initially at its center. Moreover, under certain conditions, they will secrete cell-to-cell attractant signals so that they will group and protect each other. These bacteria can swarm.

*B. Optimization Function of Bacterial Swarm Foraging*

The main goal based on bacterial foraging is to apply in order to find the minimum of  $P(\phi)$ ,  $\phi \in R^n$ , not in the gradient  $\nabla P(\phi)$ . Here, when  $\phi$  is the position of a bacterium, and  $J(\phi)$  is an attractant-repellant profile. That is, it means where nutrients and noxious substances are located, so  $P < 0$ ,  $P = 0$ ,  $P > 0$  represent the presence of nutrients. A neutral medium, and the presence of noxious substances, respectively can showed by

$$H(j, k, l) = \{\phi^i(j, k, l) | i = 1, 2, \dots, N\}. \tag{1}$$

Equation represents the positions of each member in the population of the  $N$  bacteria at the  $j$ th chemotactic step,  $k$ th reproduction step, and  $l$ th elimination-dispersal event. Let  $P(i, j, k, l)$  denote the cost at the location of the  $i$ th bacterium  $\phi^i(j, k, l) \in R^n$ . Reference [20, 21] let

$$\phi^i = (j+1, k, l) = \phi^i(j, k, l) + C(i)\varphi(j), \tag{2}$$

so that  $C(i) > 0$  is the size of the step taken in the random direction specified by the tumble. If at  $\phi^i(j+1, k, l)$  the cost  $J(i, j+1, k, l)$  is better (lower) than at  $\phi^i(j, k, l)$ , then another chemotactic step of size  $C(i)$  in this same direction will be taken and repeated up to a maximum number of steps  $N_s$ .  $N_s$  is the length of the lifetime of the bacteria measured by the number of chemotactic steps. Functions  $P_c^i(\phi)$ ,  $i = 1, 2, \dots, S$ , to model the cell-to-cell signaling via an attractant and a repellent is represented by [20, 21]

$$P_c(\phi) = \sum_{i=1}^N P_{cc}^i = \sum_{i=1}^N \left[ -L_{attract} \exp \left( -\delta_{attract} \sum_{j=1}^n (\phi_j - \phi_j^i)^2 \right) \right] + \sum_{i=1}^N \left[ -K_{repellant} \exp \left( -\delta_{attract} \sum_{j=1}^n (\phi_j - \phi_j^i)^2 \right) \right], \tag{3}$$

When we where  $\phi = [\phi_1, \dots, \phi_p]^T$  is a point on the optimization domain,  $L_{attract}$  is the depth of the attractant released by the cell and  $\delta_{attract}$  is a measure of the width of the attractant signal.  $K_{repellant} = L_{attract}$  is the height of the repellent effect magnitude), and  $\delta_{attract}$  is a measure of the width of the repellent. The expression of  $P_c(\phi)$  means that its value does not depend on the nutrient concentration at position  $\phi$ . That is, a bacterium with high nutrient concentration secrets stronger attractant than one with low nutrient concentration. Model use the function  $P_{ar}(\phi)$  to represent the environment-dependent cell-to-cell signaling as

$$P_{ar}(\phi) = \exp(T - P(\phi))P_c(\phi) \tag{4}$$

where  $T$  is a tunable parameter. Model consider minimization of  $P(i, j, k, l) + P_{ar}(\phi^i(j, k, l))$ , so that the cells will try to find nutrients, avoid noxious substances, and at

the same time try to move toward other cells, but not too close to them. The function  $P_{ar}(\phi^i(j, k, l))$  implies that, with  $M$  being constant, the smaller  $P(\phi)$ , the larger  $P_{ar}(\phi)$  and thus the stronger attraction, which is intuitively reasonable. In tuning the parameter  $M$ , it is normally found that, when  $M$  is very large,  $P_{ar}(\phi)$  is much larger than  $J(\phi)$ , and thus the profile of the search space is dominated by the chemical attractant secreted by E. coli. On the other hand, if  $T$  is very small, then  $P_{ar}(\phi)$  is much smaller than  $P(\phi)$ , and it is the effect of the nutrients that dominates. In  $P_{ar}(\phi)$ , the scaling factor of  $P_c(\phi)$  is given as in exponential form.

This paper describes the method in the form of an algorithm to search optimal value of PID parameter.

[step 1] Initialize parameters  $n, N, N_C, N_S, N_{re}, N_{ed}, P_{ed}, C(i) (i=1,2,\dots,N), \phi^i$ , and random values of PID parameter.

Where,

$n$ : Dimension of the search space ( Each Parameter of PID controller),

$N$ : The number of bacteria in the population,

$N_C$ : chemotactic steps,

$N_{re}$ : The number of reproduction steps,

$N_{ed}$ : the number of elimination-dispersal events,

$P_{ed}$ : elimination-dispersal with probability,

$C(i)$ : the size of the step taken in the random direction specified by the tumble. The controller parameter is searched in the range of  $K_p=[0\ 30], T_i=[0\ 30],$  and  $T_d=[0\ 30]$ .

[step 2] Elimination-dispersal loop:  $l=l+1$

[step 3] Reproduction loop:  $k=k+1$

[step 4] Chemotaxis loop:  $j=j+1$

[substep a] For  $i=1,2,\dots,N$ , take a chemotactic step for bacterium  $i$  as follows.

[substep b] Compute  $ITSE(i, j, k, l)$ .

[substep c] Let  $ITSE_{last}=ITSE(i, j, k, l)$  to save this value since we may find a better cost via a run.

[substep d] Tumble: generate a random vector  $\Delta(i) \in R^n$  with each element  $\Delta_m(i), m=1,2,\dots, p$ , a random number on  $[-1, 1]$ .

[substep e] Move: Let

$$\phi^i(j+1, k, l) = \phi^i(j, k, l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$$

This results in a step of size  $C(i)$  in the direction of the tumble for bacterium  $i$ .

[substep f] Compute  $ITSE(i, j+1, k, l)$ .

[substep g] Swim

i) Let  $m=0$  (counter for swim length).

ii) While  $m < N_s$  (if have not climbed down too long).

• Let  $m=m+1$ .

• If  $ITSE(i, j+1, k, l) < ITSE_{last}$  (if doing better),

let  $ITSE_{last}=ITSE(i, j+1, k, l)$  and let

$$\phi^i(j+1, k, l) = \phi^i(j+1, k, l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$$

and use this  $\phi^i(j+1, k, l)$  to compute the new  $ITSE(i, j+1, k, l)$  as we did in [substep f]

• Else, let  $m=N_s$ . This is the end of the while statement.

[substep h] Go to next bacterium  $(i, 1)$  if  $i \neq N$  (i.e., go to [substep b] to process the next bacterium).

[step 5] If  $j < N_C$ , go to step 3. In this case, continue chemotaxis, since the life of the bacteria is not over.

[step 6] Reproduction:

[substep a] For the given  $k$  and  $l$ , and for each  $i=1,2,\dots,N$ , let

$$ITSE_{health}^i = \sum_{j=1}^{N_c+1} ITSE(i, j, k, l)$$

be the health of bacterium  $i$  (a measure of how many nutrients it got over its lifetime and how successful it was at avoiding noxious substances). Sort bacteria and chemotactic parameters  $C(i)$  in order of ascending cost  $ITSE_{health}$  (higher cost means lower health).

[substep b] The  $S_r$  bacteria with the highest  $ITSE_{health}$  values die and the other  $S_r$  bacteria with the best values split (and the copies that are made are placed at the same location as their parent).

[step 7] If  $k < N_{re}$ , go to [step 3]. In this case, we have not reached the number of specified reproduction steps, so we start the next generation in the chemotactic loop.

[step 8] Elimination-dispersal: For  $i=1,2,\dots,N$ , with probability  $P_{ed}$ , eliminate and disperse each bacterium (this keeps the number of bacteria in the population constant).

To do this, if you eliminate a bacterium, simply disperse one to a random location on the optimization domain. If  $l < N_{ed}$  then go to [step 2]; otherwise end.

### 5. Simulations and Discussions

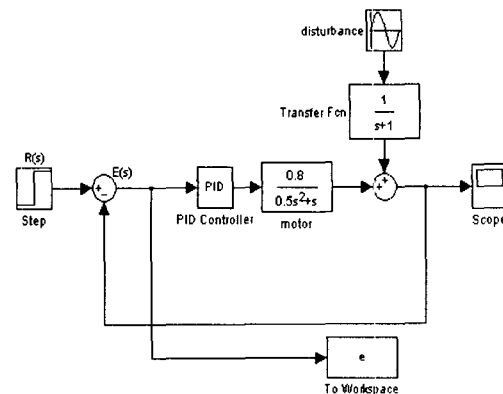


Fig. 2. Simulink block diagram for simulation of bacterial based optimization.

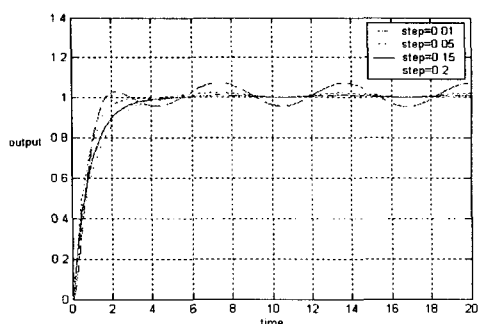


Fig. 3. Step response by variation of chemotactic step size.

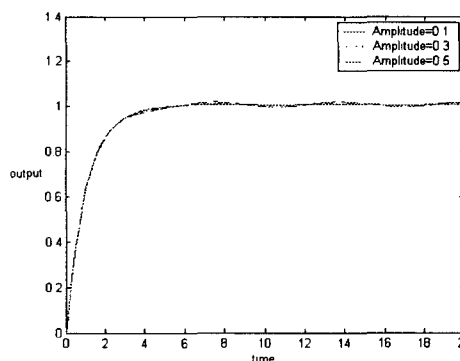


Fig. 7. Step response to a type of sine wave disturbance.

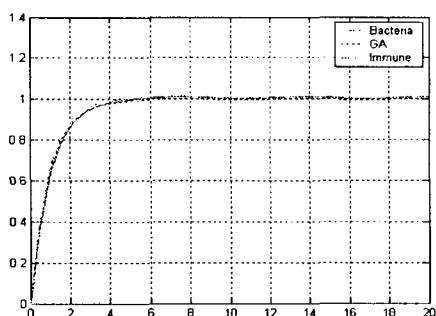


Fig. 4. Comparison of each optimal algorithm. (GA, Immune algorithm, Bacteria Foraging)

Table 1. The value of PID parameter and ITSE to variation of chemotactic step size.

Chemotactic step size	ITSE	Kp	Ti	Td
Che_size=0.01	0.094163	3.0605	0.076235	1.1411
Che_size=0.05	0.003656	13.704	0.2733	8.773
Che_size=0.15	0.000678	30	0.23208	25.844
Che_size=0.2	0.000668	29.901	0.25813	30

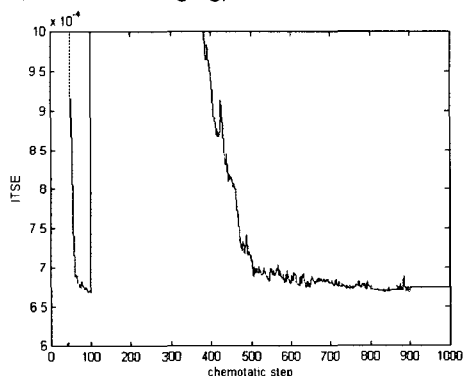


Fig. 5. Search process of performance index (ITSE) by Bacteria Foraging.

Table 2. Comparison of PID parameter and ITSE of each optimal algorithm.

	Bacteria Foraging	GA[1]	Immune Algorithm
Kp	29.901	29.992	29.739
Ti	0.25813	0.0001	0.39477
Td	30	28.3819	27.277
ITSE	0.000668	0.000668	0.0006352

Fig. 3 shows the step response to variation of chemotactic size. When step size is 0.15 response is best response. Fig. 4 is comparison of results by GA (genetic algorithm), immune algorithm, and bacterial foraging. Fig. 5 is representing search process of performance index (ITSE) by bacteria foraging and Fig. 6 is search process to have optimal PID parameters by bacteria foraging. Fig. 7 is step response to a type of sine wave disturbance.

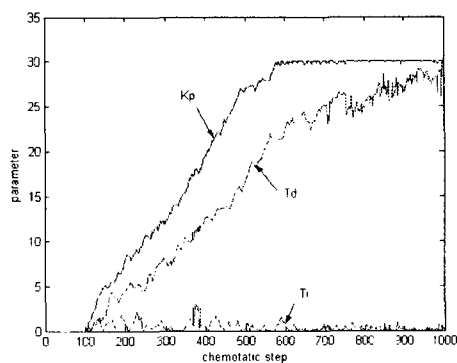


Fig. 6. Search process of optimal PID parameters by Bacteria Foraging.

### 6. Conclusions

Up to now, the PID controller has been used to operate the process loops including motor control. However, achieving an optimal PID gain is very difficult for the control loop with disturbances. Since the gain of the PID controller has to be tuned manually by trial and error. Tuning of the PID controller may not cover a plant with complex dynamics, such as large dead time, inverse response, and a highly nonlinear characteristic without any control experience. Since natural selection of animal tends to eliminate animals with poor foraging strategies for locating, handling, and

ingesting food, they obtain enough food to enable them to reproduce after many generations, poor foraging strategies are either eliminated or shaped into good ones redesigned. Therefore, optimization approach can be provided for social foraging where groups of parameters communicate to cooperatively forage in engineering.

In this paper, an intelligent tuning method of PID controller by bacterial foraging based optimal algorithm is suggested for robust control with disturbance rejection function on control system of motor control loop. Simulation results are showing satisfactory responses. The object function can be minimized by gain selection for control, and the variety gain is obtained as shown in Table 1 and 2. The suggested controller can also be used effectively in the control system as seen from Figs. 3-7.

### References

- [1] ACPA, Special Issue on Advance in PID control, Asian Journal of Control, Vol. 4, No. 4, 2002.
- [2] K. J. Astrom, T. Hagglund, C. C. Hang, and W. K. Ho, Automatic Tuning and Adaptation for PID Controllers-A Survey," IFAC J. Control Eng. Practice, Vol.1, no.4, pp.699-714, 1993.
- [3] K. J. Astrom, T. Hagglund, PID Controllers: Theory, Design, and Tuning. Research Triangle Park, NC: ISA, 1995.
- [4] K. J. Astrom and T. Hagglund, "Automatic Tuning of Simple Regulators with Specifications on Phase and Amplitude Margins," Automatica, Vol. 20, No.5, pp. 645-651, 1984.
- [5] K. J. Astrom and T. Hagglund, "The Future of PID Control," IFAC J. Control Engineering Practice, Vol. 9, pp. 1163-1175, 2001.
- [6] S. Yamato and I. Hashimoto, "Present Status and Future Needs the View from Japanese Industry," in Proc. CPCIV, Proce. 4th Int. Conf. Chem. Process Contr., Arkun and Ray, Eds., TX, 1991.
- [7] J. X. Xu, Y. M. Pok, C. Liu, and C. C. Hang, "Tuning and Analysis of a Fuzzy PI Controller Based on Gain and Phase Margins," IEEE Trans. on Systems, Man, and Cybernetics- Part A: Systems and Humans, Vol. 28, No. 5, pp. 685-691, 1998.
- [8] J. X. Xu, C. Liu, and C. C. Hang, "Tuning of Fuzzy PI Controllers Based on Gain/Phase Margin Specifications and ITAE Index," ISA Transactions, Vol. 35, pp. 79-91, 1996.
- [9] J. G. Ziegler and N. B. Nichols, "Optimum Settings for Automatic Controllers," Trans. ASME, Vol. 64, pp. 759-768, 1942.
- [10] M. Zhung and D. P. Atherton, "Automatic Tuning of Optimum PID Controllers," IEE Proc. D, Control Theory and Applications, Vol. 140, No. 3, pp. 216-224, 1993.
- [11] STEPHENS, D. W., and KREBS, J. R., Foraging Theory, Princeton University Press, Princeton, New Jersey, 1986.
- [12] ALCOCK, J., Animal Behavior: An Evolutionary Approach, Sinauer Associates, Sunderland, Massachusetts, 1998.
- [13] BELL, W. J., Searching Behavior: The Behavioral Ecology of Finding Resources, Chapman and Hall, London, England, 1991.
- [14] GRUNBAUM, D., Schooling as a Strategy for Taxis in a Noisy Environment, Evolutionary Ecology, Vol. 12, pp. 503-522, 1998.
- [15] BONABEAU, E., DORIGO, M., and THERAULAZ, G., Swarm Intelligence: From Natural to Artificial Systems, Oxford University Press, New York, NY, 1999.
- [16] ADAMI, C., Introduction to Artificial Life, Springer Verlag, New York, NY, 1998.
- [17] RESNICK, M., Turtles, Termites, and Traffic Jams: Explorations in Massively Parallel Microworlds, MIT Press, Cambridge, Massachusetts, 1994.
- [18] LEVY, S., Artificial Life: A Report from the Frontier where Computers Meet Biology, Vintage Books, New York, NY, 1992.
- [19] ARIN, S., DENEUBOURG, J. L., GOSS, S., and PASTEELS, J. M., Functional self-Organization Illustrated by Interest Traffic in Ants: The Case of the Argentine Ant, Biological Motion, Lecture Notes in Biomathematics, Edited by W. Alt JOTA: VOL. 115, NO. 3, DECEMBER 2002 627 and G. Hoffmann, Springer Verlag, Berlin, Germany, Vol. 89, pp. 533-547, 1990.
- [20] PASSINO, K. M., Biomimicry of Bacterial Foraging for Distributed Optimization University Press, Princeton, New Jersey, 2001.
- [21] PASSINO, K. M., Biomimicry of Bacterial Foraging for Distributed Optimization and Control, IEEE Control Systems Magazine, 2002.
- [22] Farmer, J. D., Packard, N. H and Perelson, A. S.: The immune system, adaptation, and machine learning, Vol. Physica. D, No. 22, (1986) 187 – 204.
- [23] Kazuyuki Mori and Makoto Tsukiyama: Immune algorithm with searching diversity and its application to resource allocation problem. Trans. JIEE, Vol. 113 - C, No. 10, (1993).
- [24] Dong Hwa Kim: Intelligent tuning of a PID controller using a immune algorithm. Trans. KIEE , vol. 51-D, no.1, (2002).