

Bacterial Foraging Oriented by Particle Swarm Optimization Strategy for PID Tuning

Wael Korani
wkorani@acm.org
Department of Electrical and Machine
University of Cairo, Egypt

ABSTRACT

Proportional integral derivative (PID) controller tuning is an area of interest for researchers in many disciplines of science and engineering. This paper presents a new algorithm for PID controller tuning based on a combination of the foraging behavior of *E coli* bacteria foraging and Particle Swarm Optimization (PSO). The *E coli* algorithm depends on random search directions which may lead to delay in reaching the global solution. The PSO algorithm may lead to possible entrapment in local minimum solutions. This paper proposed a new algorithm Bacteria Foraging oriented by PSO (BF-PSO). The new algorithm is proposed to combine both algorithms' advantages in order to get better optimization values. The proposed algorithm is applied to the problem of PID controller tuning and is compared with conveniently Bacterial Foraging algorithm and Particle swarm optimization.

Categories and Subject Descriptors: I.2 [Computing Methodologies]: ARTIFICIAL INTELLIGENCE; I.2.8 [Problem Solving, Control Methods, and Search]: Control theory

General Terms: Algorithms, Performance.

Keywords: Bacterial Foraging, Particle Swarm Optimization, Tuning of PID controller.

1. INTRODUCTION

As a result of extensive investigation to devise methods of choosing optimum controller setting for the PID controller, Ziegler and Nichols showed how they could be estimated using open and closed loop tests on the plants. The method is referred to as ZN rules. The ZN setting usually experiences excessive overshoot of the plant response. With the ease of computation, numerical optimization methods become significant in devising formula for PI and PID controller parameter tuning. The squared error integral criteria are the most common for such optimization.

Several optimization techniques using the swarming principle have been adopted to solve a variety of engineering problems in the past decade. Ant Colony Optimization (ACO) was introduced around 1991-1992 by M. Dorigo and colleagues as a novel nature-inspired metaheuristic for the solution of hard combinatorial optimization problems. Farooq et al developed a bee inspired algorithm for routing in telecommunication network. The work is inspired by

the way these insects communicate. Swarming strategies in bird flocking and fish schooling are used in the Particle Swarm Optimization (PSO) introduced by Eberhart and Kennedy [1]. A relatively newer evolutionary computation algorithm, called Bacterial Foraging scheme has been proposed and introduced recently by K.M.Passino [2].

In this paper, the use of both PSO and (*E coli*) based optimization for PID parameter tuning is investigated. A new algorithm bacterial foraging oriented by particle swarm optimization (BF-PSO) is proposed that combine the above mentioned optimization algorithms.

2. BASIC PARTICLE SWARM OPTIMIZATION (PSO)

The Particle Swarm Optimization (PSO) model [1] consists of a swarm of particles, which are initialized with a population of random candidate solutions. They move iteratively through the d-dimension problem space to search the new solutions. Each particle has a position represented by a position-vector X_k^i where (i is the index of the particle), and a velocity represented by a velocity-vector V_k^i . Each particle remembers its own best position P_{Lbest}^i . The best position vector among the swarm then stored in a vector P_{Global}^i . During the iteration time k , the update of the velocity from the previous velocity to the new velocity is determined by.

$$V_{k+1}^i = V_k^i + C_1 R_1 (P_{Lbest}^i - X_k^i) + C_2 R_2 (P_{Global}^i - X_k^i) \quad (1)$$

The new position is then determined by the sum of the previous position and the new velocity.

$$X_{k+1}^i = X_k^i + V_{k+1}^i \quad (2)$$

Where R_1 and R_2 are random numbers. A particle decides where to move next, considering its own experience, which is the memory of its best past position, and the experience of the most successful particle in the swarm .

3. BASIC BACTERIAL FORAGING OPTIMIZATION (BF)

The selection behavior of bacteria tends to eliminate poor foraging strategies and improve successful foraging strategies. After many generations a foraging animal takes actions to maximize the energy obtained per unit time spent foraging. This activity of foraging led the researchers to use it as optimization process. The *E coli* bacterium has a control system that enables it to search for food and try to avoid

Noxious substances. The bacteria distributed motion can modeled as the following four stages:

3.1 Swarming and Tumbling via flagella (N_s)

The flagellum is a left-handed helix configured so that as the base of the flagellum (i.e. where it is connected to the cell) rotate *counterclockwise*, as shown in figure 1-a

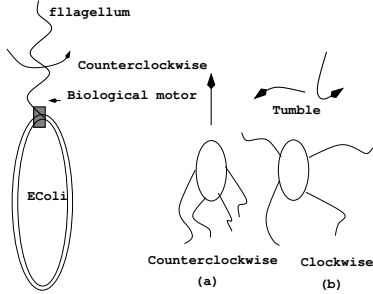


Figure 1: E coli bacteria while it's swimming or tumbling

from the free end of the flagellum looking toward the cell, it produces a force against the bacterium pushing the cell. This mode of motion is called *swimming*. Bacteria swims either for maximum number of steps N_s or less depending on the nutrition concentration and environment condition. But if the flagellum rotate *clockwise* each flagellum pulls on the cell as shown in figure 1-b, so that the net effect is that each flagellum operates relatively independently of the others and so the bacterium "tumble". Tumbling mode indicates a change in the future swim direction. Alternates between this two modes of operation in the entire life time.

3.2 Chemotaxis (N_c)

A chemotaxis step is a set of consequence swim steps following by tumble. A maximum of swim steps with a chemotactic step is predefined by N_s . The actual number of swim steps is determined by the environment. If the environment shows good nutrients concentration in the direction of the swim, the (*E. Coli*) bacteria swim more steps. The end of the chemotactic step is determined by either reaching the maximum number of steps N_s or by reaching a poor environment. When the swim steps is stopped a tumble action takes place.

To represent a tumble, a *random* unit length vector with direction $\Delta(n,i)$ is generated. where j be the index for the chemotactic step, i is the index of bacterium that has the maximum number of bacteria S . This vector is used to define the direction of movement after a tumble. Let N_c be length of the lifetime of the bacteria as measured by the number of chemotaxis steps they take during their life. Let $c^i > 0$ $i = 1, 2, \dots, S$ denote a basic chemotactic step size that we will use to define the lengths of steps during runs. The step size is assumed to be constant. The position of each bacterium is denoted by $P(n, i, j, k, ell)$ where n is the dimension of search space, k is the index of reproduction step and ell is the index of elimination-dispersal events. The new bacterium position after tumbling is given by.

$$P_{n,j+1,k,ell}^i = P_{n,j,k,ell}^i + \Delta(n, i) * c^i \quad (3)$$

3.3 Reproduction (N_{re})

After N_c chemotactic steps, a reproduction step is taken. Let N_{re} be the number of reproduction steps to be taken. For convenience, we assume that S is a positive even integer. Let

$$S_r = \frac{S}{2} \quad (4)$$

be the number of population members who have had sufficient nutrients so that they will reproduce (split in two) with no mutations. For reproduction, the population is sorted in order of ascending accumulated cost (higher accumulated cost represents that it did not get as many nutrients during its lifetime of foraging and hence, is not as "healthy" and thus unlikely to reproduce). The S_r least healthy bacteria die and the other S_r healthiest bacteria each split into two bacteria, which are placed at the same location.

3.4 Elimination and dispersal (N_{ed})

Elimination event may occur for example when local significant increases in heat kills a population of bacteria that are currently in a region with a high concentration of nutrients. A sudden flow of water can disperse bacteria from one place to another. The effect of elimination and dispersal events is possibly destroying chemotactic progress, but they also have the effect of assisting in Chemotaxis, since dispersal may place bacteria near good food sources.

The bacterial foraging algorithm has been tested for control applications like harmonic estimation for a signal distorted with additive noise [3], and adaptive control [2]. The combination of bacteria foraging and genetic algorithm is used to tune a PID controller of an automatic voltage regulator [4]. In this paper (*E coli*) is used for tuning PID controller of the plant transfer function and the results are reported.

4. BACTERIAL FORAGING OPTIMIZATION ORIENTED BY PARTICLE SWARM OPTIMIZATION

The (BF-PSO) combines both algorithms BF and PSO. This combination aims to make use of PSO ability to exchange social information and BF ability in finding a new solution by elimination and dispersal.

For initialization, the user selects S , N_s , N_c , N_{re} , N_{ed} , P_{ed} , C_1 , C_2 , R_1 , R_2 and $c(i)$, $i = 1, 2 \dots S$. Also initialize the Position $P_{n,1,1,1}^i$, $i = 1, 2 \dots S$ and Velocity randomly initialized. The (BF-PSO) models bacterial Population Chemotaxis, swarming, reproduction, elimination and dispersal oriented by PSO is given below (Initially, $j = k = ell = 0$). Implicit subscribes will be dropped for simplicity.

1. Initialize parameters n , S , N_c , N_s , N_{re} , N_{ed} , P_{ed} , $c(i)$ ($i = 1, 2 \dots S$), $\Delta(n, i)$, C_1 , C_2 , R_1 , R_2 . where,
 - n : Dimension of the search space,
 - S : The number of bacteria in the population,
 - S_r : Half the total number of bacteria ,
 - N_s : Maximum number of swim length,
 - N_c : Chemotactic steps,
 - N_{re} : The number of reproduction steps,

- N_{ed} : Elimination and dispersal events,
- P_{ed} : Elimination and dispersal with probability,
- $c(i)$: The step size taken in the random direction,
- $C1, C2$: PSO random parameter,
- $R1, R2$: PSO random parameter.

2. Generate a random direction $\Delta(n,i)$ and position.

For ($ell=1$ to N_{ed})

For ($k=1$ to N_{re})

For ($j=1$ to N_c)

For ($i=1$ to S)

Evaluate the cost function

$$J(i, j) = \text{Func}(P(i, j))$$

Store the best cost function in J_{last}

$$J_{last} = J(i, j)$$

The best cost for each bacteria will be selected to be the local best J_{local}

$$J_{local}(i, j) = J_{last}(i, j)$$

Update position and cost function

$$P(i, j+1) = P(i, j) + c(i) * \Delta(n, i)$$

$$J(i, j+1) = \text{Func}(P(i, j+1))$$

while ($m < N_s$)

$$\text{If } J(i, j+1) < J_{last}$$

then

$$J_{last} = J(i, j+1)$$

Update position and cost function

$$P(i, j+1) = P(i, j+1) + c(i) * \Delta(n, i)$$

$$J(i, j+1) = \text{Func}(P(i, j+1))$$

Evaluate the current position and local cost for each bacteria

$$P_{current}(i, j+1) = P(i, j+1)$$

$$J_{local}(i, j+1) = J_{last}(i, j+1)$$

else

$$P_{current}(i, j+1) = P(i, j+1)$$

$$J_{local}(i, j+1) = J_{last}(i, j+1)$$

end if

$$m = m + 1$$

end while

next i (next bacteria)

Evaluate the local best position (P_{lbest}) for each bacteria and global best position (P_{gbest}).

Evaluate the new direction for each bacteria

$$V = w * V + C1 * R1(P_{lbest} - P_{current}) + C2 * R2(P_{gbest} - P_{current})$$

$$\Delta = V$$

next j (next chemotactic)

for ($i = 1$ to S)

$$J_{health}^i = \sum_{j=1}^{N_c+1} (i, j, k, ell)$$

end

The S_r bacteria with the highest J_{health} remove and the other S_r bacteria with the best values copies.

next k (next reproduction)

With probability P_{ed} , eliminates and disperse each bacterium.

next ell (next elimination)

5. PID TUNING BY PSO, BF AND (BF-PSO)

The prime objective of this work is to test the performance of the developed bacterial foraging oriented by particle swarm optimization algorithm PID controller tuning.

Attempt has been made to achieve globally minimal error squared error integral criteria in the step response of a process which is cascaded with PID controller by tuning the K_p proportional gain, K_i integral gain and K_d differential gain values.

Usually, the choice of the controller coefficients is implemented by approximate methods, which in turn will not guarantee globally optimal solution for control applications. The values of K_p , K_i and K_d derived through the BF, PSO and (BF-PSO) methods after ensuring the presence of all the poles of the transfer function confined to the left half of the S plane.

The performance of the developed algorithm is tested with transfer functions of systems of different orders. The cost function here is the square of integral error. The closed loop PID controller cascaded with the process is tuned for values K_p , K_i and K_d . Results obtained by using (BF-PSO) algorithm are presented in table 1. Table 2 presents

the tuning results using PSO and BF. These parameters are randomly initialized in the same range for all methods.

For the first plant, the values of K_p , K_d found by BF and (BF-PSO) are nearly the same. The solution provided by PSO is drifted by about 20% of the values founded by BF and (BF-PSO). This is reflected in the cost values shown in table 3 where PSO has the worst cost function.

The result founded by the three algorithms nearly the same for plant 2 and 7. It is clear the PSO algorithm is unstable for plant 4 but the solution founded by the other two algorithm almost the same. Also it is obvious that the result is obtained by PSO in plant 3 is nearer to the best one and the BF algorithm has a bad response.

In all cases (BF-PSO) results in a lower overshoot compared to other methods.

Table 1: PID values obtained by BF-PSO

plant N.O	Transfer Function	BF-PSO		
		K_D	K_I	K_P
1	$\frac{5}{s^4+3s^3+7s^2+5s}$	0.60	—	0.87
2	$\left(\frac{s+5}{s^4+17s^3+60s^2+10s}\right)$	14.7	—	3.99
3	$\left(\frac{300(s+100)}{s(s+10)(s+40)}\right)$	0.03	—	0.16
4	$\left(\frac{6}{s^4+3s^3+4s^2+3s+1}\right)$	0.40	0.07	0.3
5	$\left(\frac{250s+500}{s^3+12s^2+100s+10}\right)$	0.11	0.03	0.40
6	$\left(\frac{s+5}{s^4+17s^3+60s^2+5s+5}\right)$	12.71	0.34	3.76
7	$\left(\frac{1}{s^2+0.1s+1}\right)$	1.26	0.83	1.17

Table 2: PID values obtained by PSO and BF

plant N.O	BF			PSO		
	K_D	K_I	K_P	K_D	K_I	K_P
1	0.53	—	0.80	0.62	—	0.56
2	14.69	—	4.05	14.41	—	4.36
3	0.24	—	0.08	0.03	—	0.17
4	0.43	0.08	0.30	15.92	0.48	0.10
5	0.13	0.10	0.37	0.21	0.18	0.44
6	5.82	0.27	2.83	13.00	0.12	3.78
7	1.25	0.78	1.20	3.58	1.11	0.72

Table 3: cost function obtained by BF-PSO and PSO and BF

plant N.O	Cost function		
	BF-PSO	PSO	BF
1	15.2633	19.1436	15.7906
2	21.6921	22.3756	21.8843
3	1.0525	2.8484	13.4133
4	20.8912	4646.5	20.9543
5	0.6999	1.5457	1.2674
6	21.0880	25.6203	22.8060
7	7.3021	15.3445	7.7158

6. CONCLUSION

In this paper, a new BF oriented by PSO optimization algorithm is proposed. This algorithm combines PSO and BF techniques in order to make use of PSO ability to exchange social information and BF ability in finding a new solution by elimination and dispersal. The proposed technique is applied to the PID parameter tuning for a set of test plants.

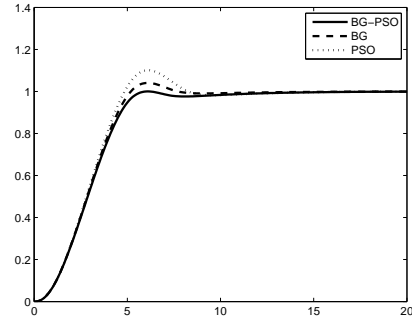


Figure 2: step response of a plant (2)

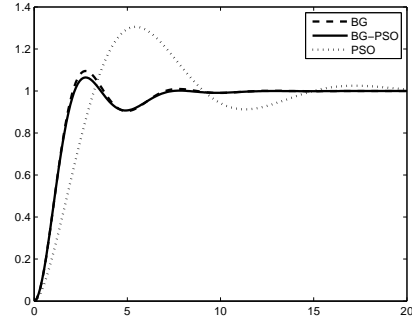


Figure 3: step response of a plant (7)

Simulation results demonstrate that the proposed algorithm out performance both conventional PSO and BF.

7. FUTURE WORK

The work presented in this paper may be extended by considering the following parts.

- Application of developed algorithm in other conditional application ex. system identification , model predictive control.
- Considering valuable step size of the bacterium to agree the digitization effect of the solution.
- Application of the develop algorithm to non linear plant with time delay.

8. REFERENCES

- [1] J. Kennedy and R. C. Eberhart, "Particle Swarm Optimization," in *Proc. of the IEEE Int. Conf. on Neural Networks*. Piscataway, NJ: IEEE Service Center, 1995, pp. 1942–1948.
- [2] K.M.Passino, "Biomimicry of Bacterial Foraging for Distributed Optimization and Control," *IEEE Control Systems Magazine*, vol. 22, no. 3, pp. 52–67, June 2002.
- [3] S. Mishra, "A hybrid Least Square-Fuzzy Bacterial Foraging Strategy for Harmonic Estimation," *IEEE Trans. Evolutionary Computation*, vol. 9, no. 1, pp. 61–73, 2005. [Online]. Available: <http://dx.doi.org/10.1109/TEVC.2004.840144>
- [4] D. H. Kim, A. Abraham, and J. H. Cho, "A hybrid Genetic Algorithm and Bacterial Foraging Approach for Global Optimization," *Inf. Sci.*, vol. 177, no. 18, pp. 3918–3937, 2007. [Online]. Available: