OPTIMAL PID TUNING BY USING BACTERIA FORAGING OPTIMIZATION ALGORITHM

Mohammed Alhanjouri[1], Ramzi Al ghamri[2]

1: Associate Professor, Computer Eng. Dept., IUG, Palestine, mhanjouri@iugaza.edu.ps
2: Master Student, Electrical Eng. Dept., IUG, Palestine, ramzialghamri@hotmail.com

ABSTRACT:

Recently, many algorithms used to make tuning for PID controller parameters for different applications. PID Controller gives us control of both transient and steady state response, and decreases the characteristics elements such as overshoot, steady state error, settling time, and rising time, and makes the system stable and robust. But the problem is how to choose the PID parameters to achieve a desired response. In this work we will present the Bacteria Foraging Optimization Algorithm (BFOA) as a novel algorithm using the social foraging behaviour for Escherichia coli bacteria to make tuning for PID parameters for an experimental plant system. Comparing with Practical Swarm Optimization (PSO), this proposed algorithm was more efficient in improving the step response characteristic for this system.

KEYWORDS: PID controller, Bacteria Foraging, Experimental Plant Application, Optimal Control.

I. INTRODUCTION

PID Controller is widely used in industrial systems due to its simplest and yet most efficient solution for many real words control problems. PID Controller gives us control of both transient and steady state response, and decreases the characteristics elements such as overshoot, steady state error, settling time, and rising time, and makes the system stable and robust. But the problem is how to choose the PID parameters to achieve a desired response.

To solve this problem, several strategies have been proposed by researchers over the past decades. The PI and I control parameters are tuned based on Hybrid Particle Swarm Optimization (HPSO) algorithm method [1]. PSO based multi stage fuzzy controller is proposed for solution of LFC problem in power system in [2]. Designing of PID controller for LFC in interconnected power system using PSO has been discussed in [3]. Hybrid Nero Fuzzy (HNF) approach is employed in [4] for an Automatic Generation Control (AGC) of interconnected power system with and without generation rate constraint (GRC). Application of real coded GA for optimizing the gains of a PI controller for two area thermal reheat power system has been discussed in [5]. Fuzzy Logic Controller is designed for automatic LFC of two area interconnected power system in [6].

Bacteria Foraging Optimization Algorithm (BFOA) is proposed as a solution to many problems due to its unique dispersal and elimination technique which can find favourable region, when the population involved is small. These unique features make

BFOA more efficient over other algorithms, and more suitable as optimization tool for control systems.

In this work we will propose a new optimization algorithm known as BFOA for optimal designing of PID controllers for an experimental plant system. The effectiveness of the proposed algorithm in designing PID controllers is supported by the observed simulation results, which show the ability of the proposed BFOA in damping oscillations, and in reducing settling time and overshooting.

II. PID CONTROLLER

PID Control (proportional-integral-derivative) is the widest type of automatic control and is a generic control loop feedback mechanism widely used in industrial control systems. Even though it has a relatively simple algorithm/structure as shown in Figure 1, there are many small variations in how it is applied in industry. A PID controller will correct the error between the output and the desired input or set point by calculating and give an output of correction that will adjust the process accordingly. A PID controller has the general form:

$$\mathrm{u(t)} = K_P\, \mathrm{e(t)} + K_i \int_0^t e(\tau)\, d\tau + K_d\, \frac{\mathrm{de}}{\mathrm{dt}} \qquad (1)$$

Where K_p is proportional gain, K_i is the integral gain, and K_d is the derivative gain. A PID controller form in S domain is:

$$G_c(s) = K_p + K_d s + \frac{K_i}{s} \qquad (2)$$

The PID controller calculation (algorithm) involves three separate parameters; the Proportional, the Integral and Derivative values. The Proportional value determines the reaction to the current error, the Integral determines the reaction based on the sum of recent errors and the Derivative determines the reaction to the rate at which the error has been changing.

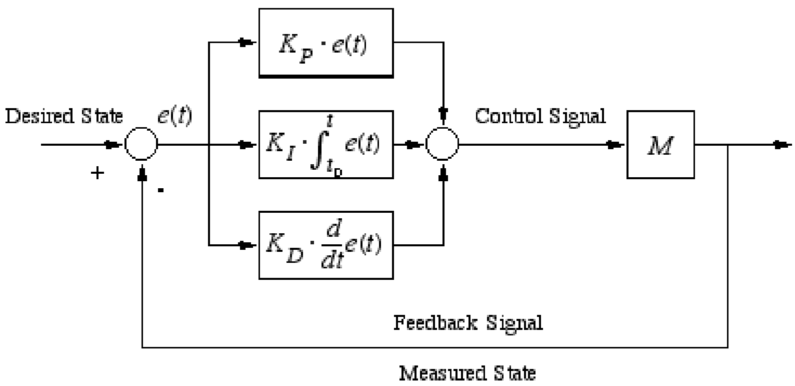

Figure 1: Block diagram of PID Controller.

III. BACTERIA FORAGING OPTIMIZATION

Many of species depends in its survival on their fitness: the law of evolution supports those species who have better food searching ability and either eliminates or reshapes those with poor search ability. Application of group foraging strategy of a swarm of *E.coli* bacteria in multi-optimal function optimization is the notion of the new algorithm. So by understanding and modeling of foraging behavior in any of the evolutionary species, leads to its application in any nonlinear system optimization algorithm. The foraging strategy of Escherichia coli bacteria present in human can be described by four processes, namely chemotaxis, swarming, reproduction, and elimination dispersal [7-8].

A. Chemotaxis

The movement of E.coli bacteria during its search for food can be defined in two ways: swimming and tumbling. It can swim for a period of time in the same direction or it may tumble (moving in random directions), and alternate between these two modes of operation for the entire lifetime. Figure 2 show the swim and tumble of bacteria

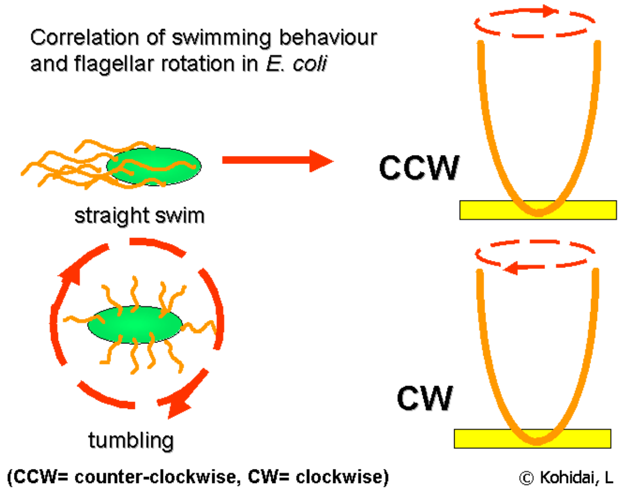

Figure 2: Swim and tumble of a bacterium

B. Swarming

The swarming means that the bacteria congregate into groups and move as a concentric patterns with high bacterial density. When the bacterium till a richest food location in the search period should try to attack other bacteria so that together they converge at the desired location.

C. Reproduction

After several stages of chemotactic, we reach to a reproduction stage. In this stage the best set of bacteria divided into two groups, the least healthy bacteria eventually die while each of the healthier bacteria asexually split into two bacteria, which are then placed in the same location. This keeps the population of bacteria constant in the evolution process.

D. Elimination and Dispersal

During the evolution process, sudden events may be occur such as rising of temperature in the local environment, may be kill a group of bacteria and/or disperse them to a new environment. Elimination and dispersal are parts of the population level and its occur by a probability ratio. In our applications this phenomenon helps us in reducing the behaviour of stagnation often seen in such parallel search algorithms.To simulate this phenomenon in BFOA some bacteria are liquidated at random with a very small probability while the new replacements are randomly initialized over the search space.

E. The BFOA Algorithm

a) Parameters

[Step 1] Initialize parameters p, S, N_c, N_s , N_{re} , N_{ed} , P_{ed}, *C(i)(i=1,2…S)*,θ_i .

p: Dimension of the search space,
S: Total number of bacteria in the population,
N_c: The number of chemotactic steps,
N_s : The swimming length.
N_{re}: The number of reproduction steps,
N_{ed}: The number of elimination-dispersal events,
P_{ed}: Elimination-dispersal probability,
C(i): The size of the step taken in the random direction specified by the tumble.

b) Algorithm

[Step 2] Elimination-dispersal loop: $l = l + 1$
[Step 3] Reproduction loop: $k = k + 1$
[Step 4] Chemotaxis loop:$j = j + 1$

[a] For i =1,2…S take a chemotactic step for bacterium i as follows.
[b] Compute fitness function, J *(i, j, k, l)*.
Let, J *(i, j, k, l)* =J *(i, j, k, l)* + J_{cc} *(*θ_i *(j, k, l), P (j, k, l))*
(i.e. add on the cell-to cell attractant–repellant profile to simulate the swarming behavior).
[c] Let J_{last} =$J(i, j, k, l)$ to save this value since we may find a better cost via a run.
[d] Tumble: generate a random vector $\Delta(i) \in R_p$with each element(i) ,$m =$ 1,2, …. , p , $m\Delta= a$ random number on [-1, 1].
[e] Move: Let

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \quad (3)$$

This results in a step of size $C(i)$ in the direction of the tumble for bacterium *i*.
[f] Compute $J(i.l+1,k,l)$ and let
J (*i*, *j*+ 1, *k*, *l*) =J (*i*, *j*, *k*, *l*) + J_{cc} (θ_i (*j*+ 1, *k*, *l*), *P*(*j*+ 1, *k*, *l*)).
[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 $J\ (i, j+1, k, l) < J_{last}$ (if doing better), let $J_{last} = J\ (i, j+1, k, l)$ and let

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \qquad (4)$$

And use this θ_i (j+1, j, k) to compute the new $J\ (i, j+1, k, l)$ as we did in [f]

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

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

[Step 5] If $j < N_c$, go to step 4. In this case continue chemotaxis since the life of the bacteria is not over.

[**Step 6**] Reproduction:

[a] For the given k and l, and for each i = 1,2,..., S , let

$$J^i_{health} = \sum_{j=1}^{N_c+1} J(i,j,k,l) \qquad (5)$$

be the health of the 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 $J\ health$ (higher cost means lower health).

[b] The S_r bacteria with the highest *health* J values die and the remaining S_r bacteria with the best values split (this process is performed by 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 of the chemotactic loop.

[**Step 8**] Elimination-dispersal: For i = 1, 2..., S with probability P_{ed} , eliminate and disperse each bacterium (this keeps the number of bacteria in the population constant). To do this, if a bacterium is eliminated, simply disperse another one to a random location on the optimization domain. If $l < N_{ed}$, then go to step 2; otherwise end.

IV. PLANT APPLICATION AND OPTIMAL RESULTS

In this section we will apply BFOA on PID tuning parameter for a plant and actuator system which has a transfer function as follows:

$$T(s) = \frac{5}{s^4+3s^3+7s^2+5s} \qquad (6)$$

And its block diagram as shown in Figure3, to ensure the effectiveness of the proposed BFOA on the transient response of the system, which has the step response before applying PID controller as shown in Figure 4.

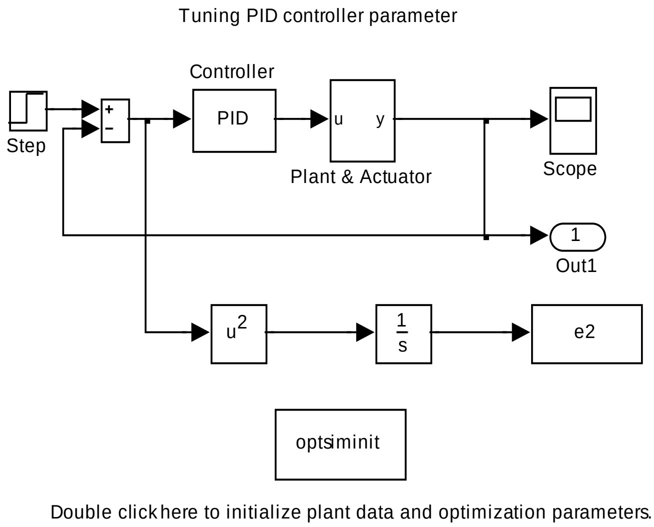

Figure 3: Block diagram from simulink for plant and actuator system.

From Figure 4 of step response behaviour without PID controller, we can observe many important parameters. The overshoot is the first important parameter needs to be observed, it is about 50% at time is 3.65 sec, and the second important parameter is the settling time that noted as 16.7 sec, while the rise time in this system, without PID controller, is 1.29sec. In fact, all researches try to reduce these parameters as possible, but this needs more investigation and techniques to be applied and compared.

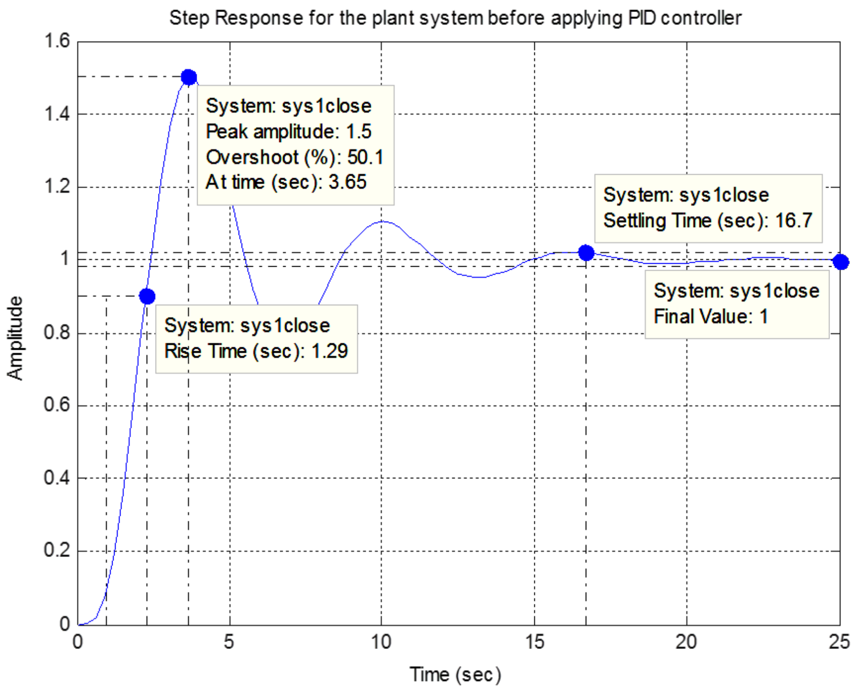

Figure 4: Step Response for the plant system before applying PID controller

One of these techniques that is applied to improve the PID is BFOA that we used in this work, and we compared it with step response in Figure4 from side, and with PSO from other side to determine the best technique for improving PID controller. Figure 5 illustrates the step response by using BFOA on PID controller parameters to control of the plant system. Figure 5 shows a good improvement on overshot, settling time, and rise time parameters related to values of Figure4.

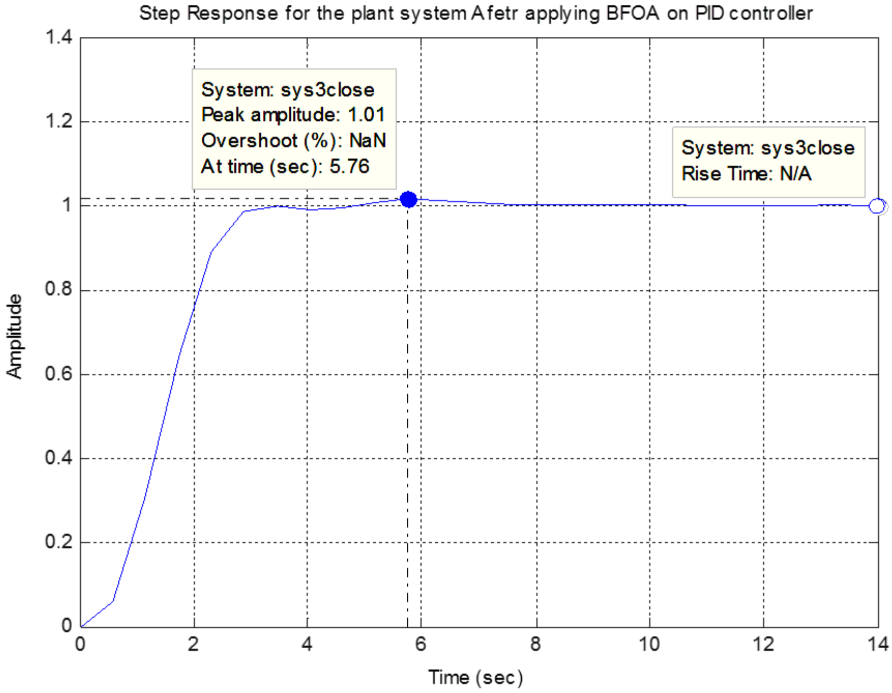

Figure 5: Step Response for the plant system in BFOA based PID controller

Table 1: Performance of the BFOA-PSO-PID Controller

Results	BFOA	PSO
[P I D]	[0.6697 0 0.4827]	[0.8425 0 0.4937]
Max overshoot (%)	0	12
Settling time (s)	7	7
Rise time (s)	1.1	1.1
Steady state error	0	0

Also we implement PSO algorithm to solve PID control to compare it with both step response without PID and step response by using BFOA based PID controller. The comparison between three techniques that used in this paper is presented in Figure6. By looking to the results, we can note that the original system (without PID) has the largest maximum overshot while the BFOA has the smallest with 0%. On the other hand, BFOA and PSO have the same values for settling time and rise time while the original system has the worst time for both settling and rise. All values for maximum overshot, rise time settling time, and steady state are summarized in Table 1 for BFOA or PSO based PID controller for the plant system.

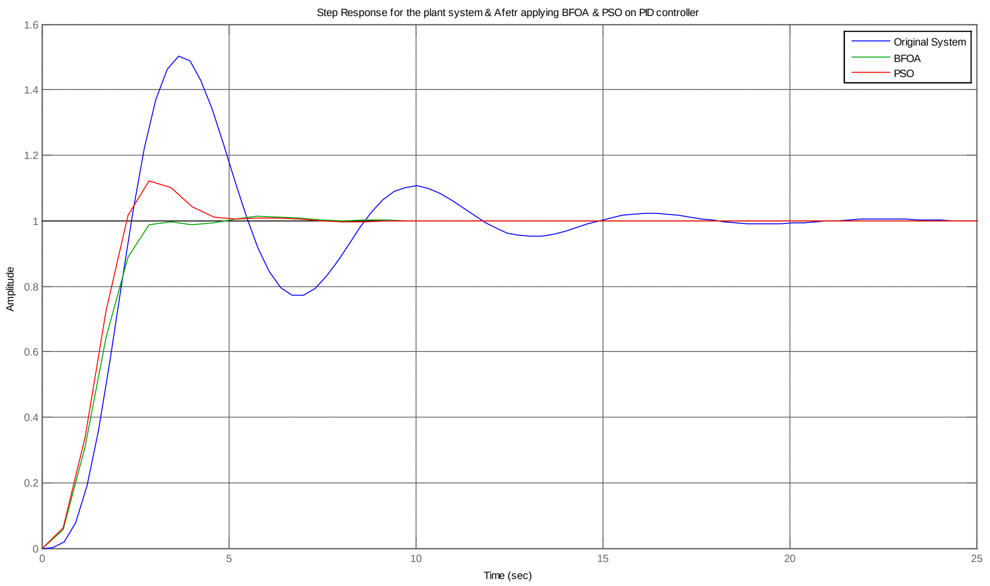

Figure 6: Comparison of Step Response between Plant System, BFOA & PSO based PID controller

The results of PFOA are produced by using the following bacteria parameters:
Number of bacteria =10
Number of chemotatic steps =5
Number of reproduction steps = 4
Probability of elimination and dispersal = 0.25
Number of elimination and dispersal events = 2

V. CONCLUSION

In this paper a new design method for determining PID controller parameters using BFOA is presented. We applied it on a plant system, and the results show its effectiveness over the other algorithms for searching of the optimal control and improving the dynamic performance of the system with 0% maximum overshot, 7 second of settling time, 1.1 second of rise time, and steady state error is 0. On other side, PSO has one drawback in maximum overshot of 12%. The worst performance is produced by original system without PID.

VI. REFERENCES

[1] S. Taher, R. Hematti, A. Abdolalipour, and S. H. Tabei, “Optimal Decentralized Load Frequency Control Using HPSO Algorithms in Deregulated Power Systems”, American Journal of Applied Sciences, vol. 5, no.9, 2008, pp 1167-1174.

[2] H. Shayeghi, A. Jalili, and H.A.Shayanfar, “Multi stage Fuzzy Load Frequency Control Using PSO” Int. J. of Energy Conversion and Management”, vol. 49, 2008, pp. 2570-2580.

[3] K. Sabahi, A. Sharifi, M. Aliyari SH, M. Teshnehlab, and M. Aliasghary, “Load Frequency Controller in Interconnected Power System Using Multi- Objective PID Controller”, Journal of Applied Sciences, 2008,pp. 1-7.

[4] G. Panda, S. Panda, and C. Ardil, “ Automatic Generation Control of Interconnected Power System with Generation Rate Constraints by Hybrid Neuro Fuzzy Approach”, Int. J. of Electrical Power and Energy Systems Engineering, vol. 2, no. 1, 2009, pp. 13-18.

[5] S. Ramesh, and A. Krishnan, “Modified Genetic Algorithm Based Load Frequency Controller for Interconnected Power System”, Int. J. of Electrical and Power Engineering, vol. 3, no. 1, 2009, pp. 26-30.

[6] P. Aravindan and M. Y. Sanavullah, “Fuzzy Logic Based Automatic Frequency Control of Two Area Power System with GRC”, Int. J. of Computational Intelligence Research, vol. 5, no. 1, 2009, pp. 37-44.

[7] D. H. Kim, A. Abraham, and J. H. Cho “A Hybrid Genetic Algorithm and Bacterial Foraging Approach for Global Optimization”, Information Sciences, vol. 177, no. 18, September 2007, pp. 3918–3937.

[8] S. Mishra, M. Tripathy, and J. Nanda, “Multi-machine Power System Stabilizer Design by Rule Based Bacteria Foraging”, Int. J. of Electric Power Systems Research vol. 77, no. 1, 2007, pp. 1595–1607.

[9] Salim Ali and S.M Abed-Elzaim " Optimal PID Tuning for Load Frequency control Using Bacteria Foraging Optimization Algorithm ", Int. M. E. Power System Conference, Cairo University, Egypt, Dec. 19-21, 2010, pp. ID 191.

[10] Mehdi Nasri, Hossein Nezamabadi, and Malihe Maghfoori, "A PSO-Based Optimal Design of PID Controller for a Linear Brushless DC Motor", Word Academy of Science, engineering and Technology 26 2007.