# Improving Transient Stability of a System Using Bacterial Foraging Optimization Technique

[1]Chitransh Tiwari, [2]Ram Kumar Denjare

[1]M.Tech. Scholar, [2]Assistant Professor
Electrical and Electronics Engineering Department,
Kalinga University Raipur, India

**Abstract- This paper deals with an optimal tuning of Proportional Integral Derivative (PID) controller of a plant using Bacteria Foraging Optimization Algorithm (BFOA). As constancy of frequency and voltage are important factors in determining the quality of power supply. PID is a feedback based controller which gets the error value and calculates the output based on the characteristics of the error it is widely used in plants as it is simple and gives good result. The robustness of the system is also examined by applying variable load changes to a system instead of fixed step change in load. The simulation results of the proposed BFOA tuned PID controller is presented. The analysis clearly reveals that the transient performances and the robustness are much improved with the proposed approach over others.**

***IndexTerms*—Bacteria foraging, PID controller, ITAE, ITSE, ISE, IAE.**

## I. INTRODUCTION

The Proportional-Integral-Derivative (PID) controller has been proved the most popular controller of this century for its remarkable effectiveness, easiness of implementation and vast applicability. But it is also hard to tune the PID controller. A number of tuning methods are done manually which are difficult and time consuming. For using PID controller efficiently, the optimal tuning of its parameters has become a significant research area. Optimization problems have been resolved with the aid of numerous techniques. Today's, an alternative approach to the traditional methods for operations research, meta-heuristic methods are implanted to simplify optimization difficulties.

Nature Inspired strategies are those, which are inspired by natural and biological events for example, immune system, foraging behavior of ants and other insects. Swarms can be considered as any collection of interacting agents or individuals and implemented strategy is inspired by intelligent behavior of insects .

Bacteria Foraging Optimization (BFO) Algorithm has been applied for tuning the subjected parameters of PID controller. This tuning method based on research of foraging behavior of E.colli bacteria proposed by Kevin M.Passino and Liu exploits a bacterial foraging and swarming behavior. It exhibits linked social foraging process along with distributed non-gradient optimization. Stochastic optimization approaches are needed to deal with more complicated open loop unstable systems . Hence the tuning objectives and parameter variations can be directly incorporated to find the optimum PID controller parameter values.

## II. TUNING OF PID CONTROLLER

The PID controller calculation involves three separate control parameters, i.e. proportional, integral and derivative values .The proportional value determines the reaction of the current error, the integral value determines the reaction based on the sum of recent errors and derivative value determines the reaction based on the rate at which the error has been changing and the weighted sum of these three actions is used to adjust the process via the final control element.

The block diagram of a control system with unity feedback employing Soft computing PID control action is shown in Figure-1.

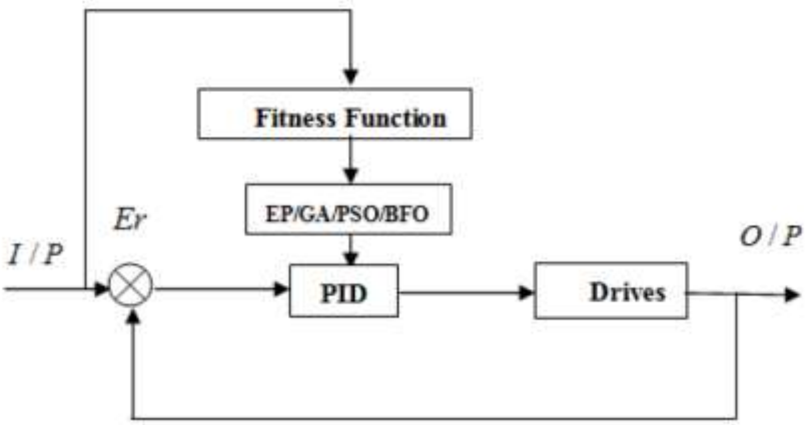


**Fig 1: PID Tuning**

The mathematical representation of PID control is given in (1).

$$u(t) = K_p\left[e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau + T_d\frac{de(t)}{dt}\right] \qquad (1)$$

A. *Classical Techniques*

Classical techniques make certain assumptions about the plant and the desired output and try to obtain analytically, or graphically some feature of the process that is then used to decide the controller settings. These techniques are computationally very fast and simple to implement, and are good as a first iteration. But due to the assumptions made, the controller settings usually do not give the desired results directly and further tuning is required. A few classical techniques have been reviewed in this paper.

B. *Computational or Optimization Techniques*

These are techniques which are usually used for data modeling and optimization of a cost function, and have been used in PID tuning. Few examples are neural networks (computational models to simulate complex systems), genetic algorithm and differential evolution. The optimization techniques require a cost function they try to minimize. There are four types of cost functions used commonly,

- Integral Absolute Error

$$\mathrm{IAE} = \int_0^{\tau} |e(t)|$$

- Integral Square Error

$$\mathrm{ISE} = \int_0^{\tau} |e(t)|^2$$

- Integral Time Absolute Error

$$\mathrm{ITAE} = \int_0^{\tau} t|e(t)|$$

- Integral Time Square Error

$$\mathrm{ITSE} = \int_0^{\tau} t|e(t)|^2$$

Computational models are used for self tuning or auto tuning of PID controllers. Self tuning of PID controllers essentially sets the PID parameters and also models the process by using some computational model and compares the outputs to see if there are any process variations, in which case the PID parameters are reset to give the desired response. The existent types of adaptive techniques are classified based on the fact that if the process dynamics are varying [3], then the controller should compensate these variations by adapting its parameters. There are two types of process dynamics variations, predictable and unpredictable. The predictable ones are typically caused by nonlinearities and can be handled using a gain schedule, which means that the controller parameters are found for different operating conditions with an auto-tuning procedure that is employed thereafter to build a schedule. Different techniques have been used to replace the gain schedule mentioned above.

## III. Bacterial Foraging Algorithm (BFA)

Recently, bacterial foraging algorithm (BFA) has emerged as a powerful technique for the solving optimization problems. BFA mimics the foraging strategy of *E. coli* bacteria which try to maximize the energy intake per unit time.

The E. coli bacterium has a plasma membrane, cell wall, and capsule that contains the cytoplasm and nucleoid (Figure 1). The pili (singular, pilus) are used for a type of gene transfer to other E. coli bacteria, and flagella (singular, flagellum) are used for locomotion . The cell is about 1 μmin diameter and 2 μm in length. The E. coli cell only weighs about 1 picogram and is about 70% water. Salmonella typhimurium is a similar type of bacterium .

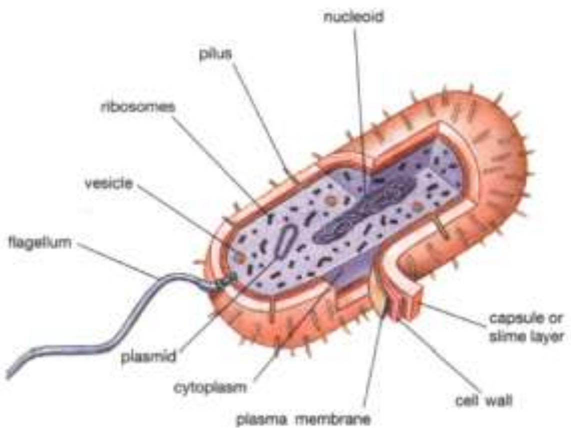


**Fig 2: Bacterial foraging E.Coli**

From the very early days it has drawn attention of researchers due to its effectiveness in the optimization domain. So as to improve its performance, a large number of modifications have already been undertaken. The bacterial foraging system consists of four principal namely chemotaxis, swarming, reproduction and elimination-dispersal. A brief description of each of these processes along with the pseudo-code of the complete algorithm is described below.

**Chemotaxis**: This process simulates the movement of an *E.coli* cell through swimming and tumbling via flagella. Biologically an *E.coli* bacterium can move in two different ways. It can swim for a period of time in the same direction or it may tumble, and alternate between these two modes of operation for the entire lifetime. Suppose $\theta^i(j,k,l)$represents $i^{th}$ bacterium at $j^{th}$chemotactic, $k^{th}$ reproductive and $1^{th}$ elimination-dispersal step. C(i) is the size of the step taken in the random direction specified by the tumble (run length unit).Then in computational chemotaxis the movement of the bacterium may be represented by

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

Where $\Delta$ indicates a vector in the random direction whose elements lie in [-1, 1].

**Swarming**: An interesting group behavior has been observed where a group of *E.coli* cells arrange themselves in a traveling ring by moving up the nutrient gradient when placed amidst a semisolid matrix with a single nutrient chemoeffecter. The cells, when stimulated by a high level of *succinate*, release an attractant *aspertate*, which helps them to aggregate into groups and thus move as concentric patterns of swarms with high bacterial density. The cell-to-cell signaling in *E. coli* swarm may be represented by the following function.

$$J_{CC}(\theta, P(j,k,l)) = \sum_{i=1}^{S} J_{CC}\left(\theta, \theta^i(j,k,l)\right)$$

$$J_{CC} = \sum_{i=1}^{S}\left[-d_{attractant}\, e^{\left(-w_{attractant}\sum_{m=1}^{p}\left(\theta_m-\theta_m^i\right)^2\right)}\right] + \sum_{i=1}^{S}\left[h_{repellant}\, e^{\left(-w_{repellant}\sum_{m=1}^{p}\left(\theta_m-\theta_m^i\right)^2\right)}\right] \qquad (2)$$

where $J_{CC}\left(\theta, P(j,k,l)\right)$ is the objective function value to be added to the actual objective function (to be minimized)to present a time varying objective function, $S$ is the total number of bacteria, $p$ is the number of variables to be optimized, which are present in each bacterium and $\theta = \left(\theta_1, \theta_2, \cdots \theta_p\right)^t$is a point in the $p$ dimensional search domain.

**Reproduction:** The least healthy bacteria eventually die while each of the healthier bacteria (those yielding lower value of the objective function) asexually split into two bacteria, which are then placed in the same location. This keeps the swarm size constant.

**Elimination and Dispersal**: Gradual or sudden changes in the local environment where a bacterium population lives may occur due to various reasons e.g. a significant local rise of temperature may kill a group of bacteria that are currently in a region with a high concentration of nutrient gradients. Events can take place in such a fashion that all the bacteria in a region are killed or a group is dispersed into a new location. Some guidelines for BFA Parameter Choices are

**Size of population 'S':** Increasing *S* can significantly increase the computational complexity of the algorithm. However, for larger values of *S*, it is more likely at least some bacteria near an optimum point should be started, and over time, it is then more likely that many bacterium will be in that region, due to either chemotaxis or reproduction.

**Length of chemotactic step** 'C(i)'**:** If C(i) are too large, then if the optimum value lies in a valley with steep edges, the search will tend to jump out of the valley, or it may simply miss possible local minima by swimming through them without stopping. On the other hand, if C(i) are too small, convergence can be slow, but if the search finds a local minimum it will typically not deviate too far from it. *c*(*i*) is a sort of a "step size" for the algorithm

**Chemotactic step '***Nc***':** If the size of *Nc* is chosen to be too short, the algorithm will generally rely more on luck and reproduction, and in some cases, it could more easily get trapped in a local minimum (premature convergence). *Ns* creates a bias in the random walk (which would not occur if Ns = 0), with large values tending to bias the walk more in the direction of climbing down the hill.

**Reproduction number '***Nre***':** If *Nre* is too small, the algorithm may converge prematurely; however, larger values of *Nre* clearly increase computational complexity.

**Elimination and dispersal number '***Ned***':** A low value for *Ned* dictates that the algorithm will not rely on random elimination-dispersal events to try to find favorable regions. A high value increases computational complexity but allows the bacteria to look in more regions to find good nutrient concentrations. Clearly, if *ped* is large, the algorithm can degrade to random exhaustive search. If, however, it is chosen appropriately, it can help the algorithm jump out of local optima and into a global optimum.

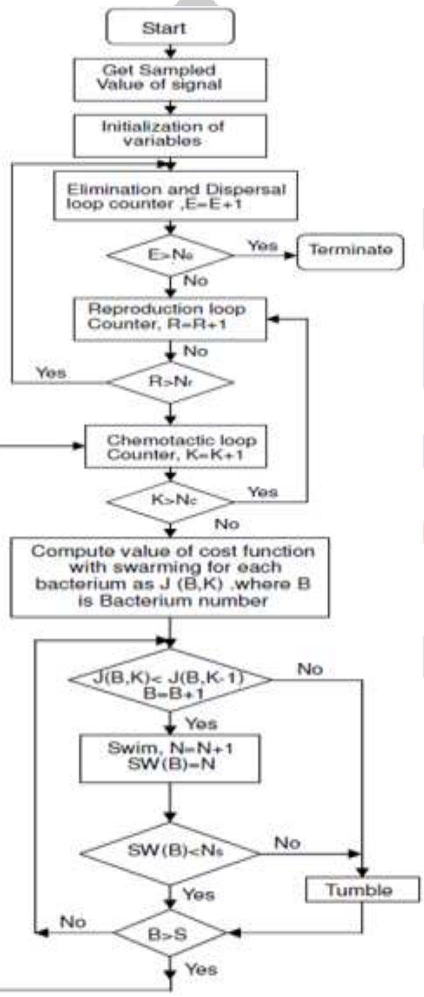


**Fig. 3 Flowchart of BFA**

**Parameters defining cell-to-cell attractant functions '$J_{cc}$':**

If the attractant width is high and very deep, the cells will have a strong tendency to swarm (they may even avoid going after nutrients and favor swarming). On the other hand, if the attractant width is small and the depth shallow, there will be little tendency to swarm and each cell will search on its own Social versus independent foraging is then dictated by the balance between the strengths of the cell-to-cell attractant signals and nutrient concentrations.

## IV. PROPOSED ALGORITHM FOR THE IMPLEMENTATION OF THE BFOA

We have used bacterial forging optimization algorithm for segmention of the image. This algorithm shows how the bacterial based theory is used to detect the image. Natural selection tends to eliminate animals with poor "foraging strategies" (methods for locating, handling, and ingesting food) and favor the propagation of genes of those animals that have successful foraging strategies since they are more likely to enjoy reproductive success (they obtain enough food to enable them to reproduce). After many generations, poor foraging strategies are either eliminated or shaped into good ones (redesigned). Logically, such evolutionary principles have led scientists in the field of "foraging theory" to hypothesize that it is appropriate to model the activity of foraging as an optimization process:

Steps:

initialize Parameters: p, S, Nc, Ns, Nre and C (i), i = 1, 2… S
Where,
p = Dimension of search space
S = Number of bacteria in the population
Nc = Number of chemotaxis steps
Ns= Number of swimming steps
Nre = Number of reproduction steps
C (i) = Step size taken in the random direction specified by the tumble
J (i, j, k) = Fitness value or cost of i-th bacteria in the j-th chemotaxis and k-th reproduction steps
θ (i, j, k)= Position vector of i-th bacterium in j-th chemotactic step and k-th reproduction steps
Jbest (j, k) = Fitness of best position in the j-th chemotaxis and k-th reproduction steps
Jglobal= Fitness value or cost of the global best position in the entire search space

**Step 1:** Update the following parameters: J (i, j, k), Jbest (j, k) and Jglobal= Jbest (j, k)

**Step 2:** Reproduction Loop: k = k+1

**Step 3:** Chemotaxis loop: j = j+1
a) Compute fitness function J (i, j, k) for i= 1, 2, 3…S
b) Update Jbest (j, k).
c) Tumble: Generate a random vector Δ(i) ∈Rp with each
element Δm (i) m = 1, 2…p, a random
d) Compute q for i = 1, 2 ………..S
e) Swim
i) Let m = 0 (counter for swim length)
ii) While m < Ns (if have not climbed down too long.

- Let m = m+1
- Compute fitness function J (i, j+1, k) for i = 1, 2… S
- Update Jbest (j+1, k)
- If Jbest (j+1,k)< Jbest (j, k) (if doing better), Jbest (j, k) = Jbest (j+1, k) Compute θ for i = 1, 2…S [Synchronous position updation] Use this θ (i, j +1, k) to compute the new J (i, j+1, k)
- Else, let m = Ns. This is the end of the while statement Sub Step f) Mutation Operator Compute θ for i = 1, 2….S [Synchronous position updation by mutation operator]

**Step 4:** If j < Nc, go to step 3. In this case, continue chemotaxis since the life of bacteria is not over.

**Step 5:** Reproductions: The Sr=S/2 bacteria with the highest cost function values die and other Sr bacteria with the best values split. Update Jglobal from Jbest (j, k).

**Step 6:** If k < Nre, go to Step 2 otherwise end.

## V. APPLICATION AREAS OF BFOA

1. Automatic circle detection using BFOA.
2. Automatic generation control using BFO.
3. Calculate resonant frequency using BFO
4. Economic emission load dispatch through fuzzy BFOA.
5. Image compression using BFOA based ANN.

## VI. RESULT AND DISCUSSION

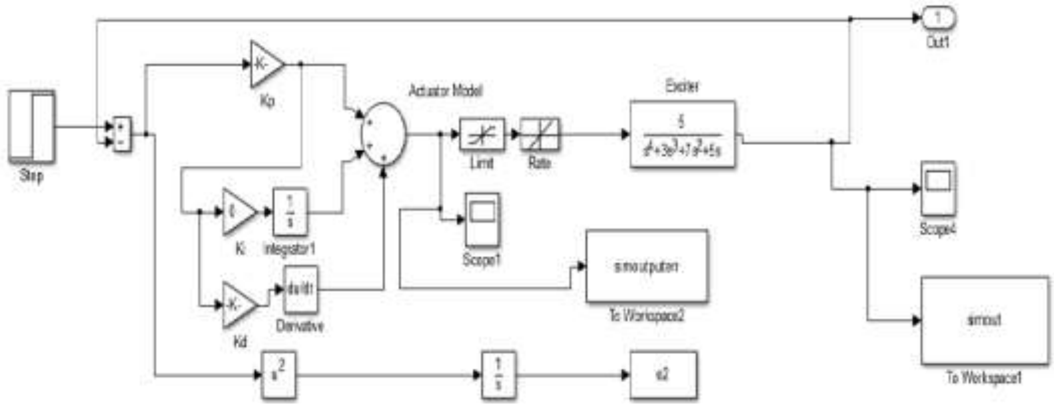


Fig. 4 : value of gain by BFO using PID Technique

As the figure 4 shows above that manipulating the values of gain by BFO PID Technique it gives the required results. In BFO there are two important factor i.e. alpha and beta which is the base of the gain values. Alpha and Beta are the two constant terms which provides the system stability. After the execution of algorithm the required values of proportional gain , integral gain and derivative gain are saved in workspace and from the work space the required values are taken p by the PID controller.

The response in Fig.5 shows that in BFO when the number of iteration is 50 and manipulating the value of beta and by keeping the value of alpha at 10 i.e. constant from that all the factors like maximum overshoot, rise time, settling time will be determined.

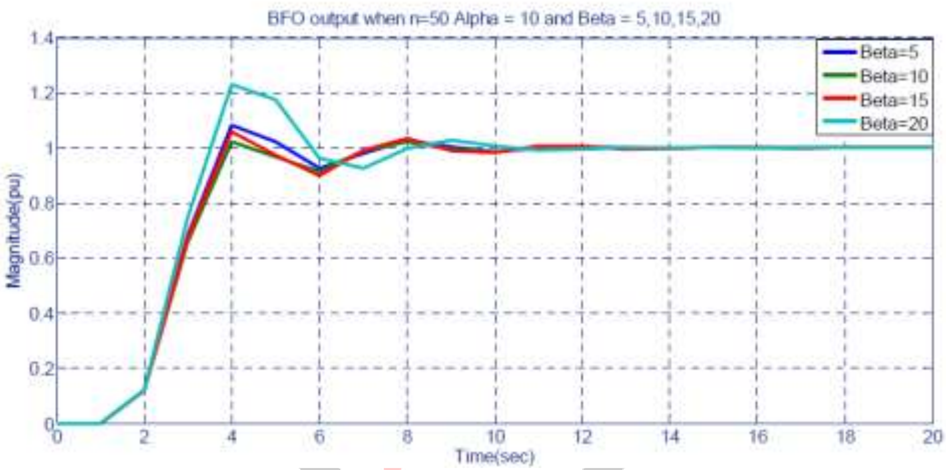


**Fig 5 : For BFO when n =50 alpha = 10 and Beta = 5,10,15,20**

The above figure shows that when n=50 and keeping the value of Alpha constant i.e. 10 and by varying the value of Beta from 5 to 20 (in 4 interval) and by applying PSO algorithm it will gives the value of Kp= 0.8809, 0.8403 0.8747 and 0.9762 and Kd = 0.6724, 0.7790, 0.8018 and 0.4032. It concludes that when Beta = 20 then the maximum overshoot as well as settling time is greater than allthe other values of Beta, and the rise time when Beta= 15 is greater than all the values of Beta.

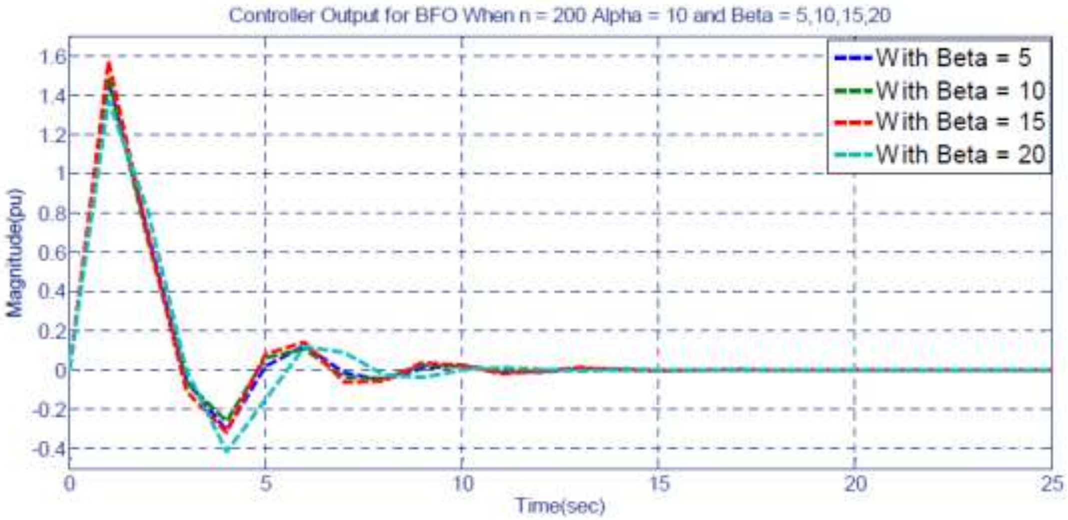


**Fig 6 : Controller output response for BfO when n =50 alpha = 10 and Beta = 5,10,15,20** The above figure shows that controller output when n=50 and applying BFO algorithm, by keeping the value of alpha constant at 10 and by varying the value of beta from 5 to 20 (in 4 interval) and by applying BFO algorithm it will gives the value of Kp and Kd and the fluctuation in the output of controller is totally depends upon the value of Alpha and Beta The Table shows that by maintaining the value of beta and regulating the value of alpha it gives the required result of Kp, Ki and Kd

| S.No | Generation | Alpha | Beta | Kp | Ki | Kd | Mp% | Ts | Tr |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 200 | 10 | 5 | 0.1694 | 0 | 0.5522 | 0 | 21.74 | 11.27 |
| 2 | 200 | 10 | 10 | 0.812 | 0 | 0.7876 | 10.4471 | 7.10 | 1.76 |
| 3 | 200 | 10 | 15 | 0.5489 | 0 | 0.7454 | 4.9755 | 8.195 | 2.396 |
| 4 | 200 | 10 | 20 | 0.920 | 0 | 0.9178 | 0 | 42.75 | 22.95 |

## VII. Conclusion

The designe of PID BFA technique that was explored as a candidate to optimally tune the PID gains represents an optimization technique mimic to the search of food for the bacteria. This is a promising technique that can be used in complex problems. From the simulation results, the system, when driven by such BFA-tuned PIDs, shows a good performance during regulation, reference tracking and parameters change tests.PID controllers using BFO based optimization have less overshoot

## References

[1] Jhankal, N.K., Adhyaru, D., “Bacterial foraging optimization algorithm: A derivative free technique,” presented at the 2nd International Conference on current in technology, Ahmedabad, India, 8-10, Dec.2011.
[2] G. Kevin Passino,”Biomimicry of Bacterial Foraging for Distributed Optimization and Control,” IEEE Control Systems Magazine, June 2002.
[3] V. Fabian, “Simulated annealing simulated,” Computers math Applic. Vol. 33, no. 1/2, pp. 81-94, 1997.
[4] Stephens and J. Krebs, Foraging Theory. Princeton, NJ: Princeton univ. Press, 1986.
[5] W. O’Brien, H. Browman, and B. Evans, “Search strategies of foraging animals,” Amer. Scientist,” vol. 78, pp. 152-160, Mar /Apr. 1990.
[6] H. Berg, Random Walks in Biology. Princeton, NJ: Princeton Univ. Press, 1993.
[7] D. De Rosier, “The turn of the screw: The bacterial flagellar motor,” Cell, vol. 93, pp. 17-20, 1998.
[8] G. Lowe, M. Meister, and H. Berg, “Rapid rotation of flagellar bundles in swimming bacteria,” Nature, vol. 325, pp. 637-640, Oct. 1987.
[9] Swagatam Das, Sambarta Dasgupta, Arijit Biswas, Ajith Abraham,” On Stability of the Chemotactic Dynamics in Bacterial-Foraging Optimization Algorithm, ’’ IEEE Trans. Syst.,Man, Cybernet-Part A: systems and humans, vol. 39, no. 3, may 2009.
[10] B. Sumanbabu, S. Mishra, B.K. Panigrahi, and G.K. Venayagamoorthy, “Robust tuning of modern power system stabilizers using bacterial foraging algorithm,” IEEE Congress on Evolutionary Computation, CEC 2007.
[11] 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, pp. 3918-3937, 2007.
[12] S. Dasgupta, S. Das, A. Abraham, and A. Biswas, “Adaptive computational chemotaxis in bacterial foraging optimization: an analysis,” IEEE Transactions on Evolutionary Computing, (in press), 2008.

[13] R. Allamneni , “Bacterial foraging based channel equalizers,” Master of Technology, Dept. Electronics and Communication Eng., National Institute of Technology, Rourkela, Orissa, Spain, May-2006.
[14] Mangraj B B, Misra IS and Barisal AK, “Optimizing Included Angle of symmetrical V-Dipoles for Higher Directivity Using Bacteria Foraging Optimization Algorithm”, PIER B, Vol. 3, 2008, pp. , 295-314

[15] Sastri GSVR, Pattnaik SS, Bajpai OP, Devi S, Sagar CV, Patra PK and Bakwad KM, “Bacterial Foraging Optimization Technique to Calculate Resonant Frequency of Rectangular Microstrip Antenna”, Int. J.

RF Microwave Computer Aided Eng., Vol. 18, 2008, pp. , 383-388.

[16] Sastri GSVR, Pattnaik SS, Bajpai OP, Devi S and Bakwad KM , “Velocity Modulated Bacterial Foraging Optimization Technique (VMBFO)”, Applied Soft Computing Journal, Vol. 11 (1), January 2011, pp. 154-165.

[17] Sastri GSVR, Pattnaik SS, Bajpai OP, Devi S, Bakwad KM and Patra PK “Intelligent Bacterial Foraging Optimization Technique to Calculate Resonant Frequency of RMA”, IJMOT, Vol 4(2), March 2009, pp.67-75.

[18] Chen H, Zu Y, and Hu K, “Adaptive Bacterial Foraging Optimization”, Abstracts and Applied Analysis, Vol. 2011, 2011, pp. 1-27.

[19] Mahmoud KR, “Design Optimization of A Bow-Tie Antenna For 2.45 GHz RFID Readers Using A Hybrid BSO-NM Algorithm”, PIER 100, 2010, pp. 105-117

[20] Bakwad KM, Pattnaik SS, Sohi BS, Devi S, Panigrahi BK, Sastri GSVR, “Bacterial Foraging Optimization Technique Cascaded with Adaptive Filter to Enhance Peak Signal to Noise Ratio from Single Image”, IETE Journal of Research, Vol 55 (4), Jul-Aug 2009.

[21] Passino KM, “Biomimicry of Bacterial Foraging”, IEEE Control System Magzine, Vol. 22, 2002, pp. 52-67.

[22] Datta T, Misra IS, Mangraj BB, Imtiaj S, “Improved Adaptive Bacteria Foraging algorithm in Optimization of Antenna Array for Faster Convergence”, PIER C, Vol. 1, 2008, pp. 143-157.

[23] Liu W, Chen H, Chen H, and Chen M, “RFID Network Scheduling Using an Adaptive Bacteria Foraging Algorithm”, Journal of Computer Information Systems (JCIS), Vol. 7(4), 2011, pp. 1238-1245.

[24] Korani W, “ Bacterial Foraging Oriented By Particle Swarm Optimization Strategy for PID Tuning”, GECOO 2008, 12-16 July, 2008, Atlanta, USA, pp. 1823-1826.

[25] Zhang Y, Wu L and Wang S, “ Bacterial Foraging Optimization Based Neural Network for Short term Load Forecasting.

[26] Singh, Mahesh, R. N. Patel, and Rajkumar Jhapte. "Performance comparison of optimized controller tuning techniques for voltage stability."2016 IEEE First International Conference on Control, Measurement and Instrumentation (CMI). IEEE, 2016.