International Journal of Information & Computation Technology.
ISSN 0974-2239 Volume 4, Number 13 (2014), pp. 1275-1289

http://www. irphouse.com

Bacterial Foraging Algorithm Based Adaptive Control of Water Bath System

Bharat Bhushan[1] and Sakshi Praliya[2]

[1, 2]Electrical Engineering
Delhi Technological University Delhi, India

Abstract

This paper introduces a bacterial foraging algorithm (BFA) based high performance adaptive control of temperature of water bath system. The temperature of the water bath system is being made to follow an arbitrary selected trajectory. The unknown nonlinear dynamics of the temperature system are captured by BFA. The trained BFA identifier is used with a desired reference model to achieve temperature trajectory control of water bath system. Simulation study on proposed system has been carried out in MATLAB. System nonlinearities alpha and beta have been estimated using BFA and compared with actual nonlinearities of dynamical system. In tracking of temperature of dynamical system using BFA based controller the performance of the dynamical system have been observed and compared with reference one. Performance study of plant has been carried out through genetic algorithm (GA) also. A comparison of performance analysis using BFA controller and that of GA controller for trajectory tracking of temperature also have been carried out.

Keywords: bacterial foraging algorithm; identification; adaptive control; genetic algorithm; water bath system.

Introduction

There are many techniques available in the literature for identification and control of the system like fuzzy logic [2, 8, 9], neural networks [12-17] and search algorithms. To tackle several complex search problems of real world, scientists have been looking into the nature for years- both as model and as metaphor- for inspiration. Optimization is at the heart of many natural processes like group behaviour of social insects, birds and the foraging strategy of other microbial creatures.

Many bio-inspired computational methodologies such as Genetic Algorithm (GA) [1, 10 and 11], Particle Swarm Optimization (PSO) [24, 26], Ant Colony [25] and Artificial Fish Swarm Algorithm (AFSA) [23] have been intensively studied and applied to various optimization problems. In recent years, a new and rapidly growing subject- Bacterial Foraging Algorithm (BFA) [22] proposed by Prof. Passion in 2002 is a novel modern search algorithm. BFA has been tested on many unconstrained global optimization functions like Sphere function, Rosenbrock function, Rastrigin function, Ackley function and Griewank function etc. A fast bacterial swarming algorithm [27] was tested on high-dimensional function optimization. Self-adaptive bacterial foraging optimization [28], employing the adaptive search strategy significantly improve the performance of the original algorithm was achieved by enabling BFA to adjust the run-length unit parameter dynamically during evolution to balance the exploration/exploitation tradeoff. A variable denoting the overall best value [29] is incorporated to guide the bacterial swarm to move to the global optima and replace the role of interaction behavior between bacteria in classic BFA which is complicated and time-consuming. Micro-bacterial foraging algorithm [30] was proposed, which evolves with a very small population compared to its classical version. To accelerate the convergence speed of the bacterial colony near global optima, two cooperative approaches have been applied to BFA [31] that resulted in a significant improvement in the performance of the original algorithm in terms of convergence speed, accuracy and robustness. BFA aiming for optimization in dynamic environments [32] called DBFA (Dynamic bacterial foraging algorithm), is studied. Analysis of Reproduction Operator [33] has been studied in Bacterial Foraging Optimization Algorithm. BFA has been applied to many kinds of real world optimization problems like antenna array [35], nonlinear systems [34], image segmentation[37] etc. High performance electric drives [18, 21] are essential in applications such as robotics [6-7], actuation and guided manipulation where precise movements are required. In high performance drives traditional approach is not satisfactory, as it results in poor speed and position tracking when sudden changes in load results in continuous acceleration or deceleration of the machines. It is desired to have an adaptive control system [3]-[5] which should response satisfactorily under the dynamic changes of the system. Many hybrid algorithms have been developed using BFA [20]. In this paper we have implemented BFA on water bath temperature control system [36].

Bacterial Foraging Algorithm

Bacterial Foraging Algorithm (BFA) [22] proposed by Prof. Passion in 2002 is a novel modern search algorithm based on the behaviour of E. coli bacteria During the lifetime of E. coli, it shows the behaviour of chemo-tactic action, which makes the E. coli swimming up a nutrient gradient or out of noxious substance. The E. coli bacterium communicates each other and at the same time competes for food. Beside the chemo-tactic action and information communication, other complex stages such as reproduction, elimination and dispersal stages are included in E. coli foraging behaviours. The optimization in BFA comprises the following processes: chemotaxis, swarming, reproduction, elimination and dispersal.

A. Chemotaxis:

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.

B. Swarming:

A group of E. coli cells arrange themselves in a travelling ring by moving up the nutrient gradient when placed amidst a semisolid matrix with a single nutrient chemo-effecter. The cells when stimulated by high level of succinate release an attractant aspartate, which helps them to aggregate into groups and thus move as concentric patterns of swarms of high bacterial density.

C. *Reproduction*:

After a certain number of complete swims, the least healthy bacteria eventually die while each of the healthier bacteria (those yielding higher value of fitness function) asexually split into two bacteria which are placed in the same location. This keeps the swarmsize constant.

D. Elimination- dispersal:

The bacteria are eliminated and dispersed to random positions in the optimization domain according to the elimination-dispersal probability. This elimination-dispersal event helps the bacterium avoid being trapped into local optima.

Discrete Time Water Bath Model

Discrete time model of a water bath system [36] is used for verifying the performance of BFA technique in identification and control. Using this model of water bath a detailed simulation and analysis of the drive is presented. From a control systems point of view, the water bath system can be considered as a SISO plant, thereby eliminating the complications associated with multi-input systems. The temperature of water in a stirred tank is described by the equation

$$C\frac{dy(t)}{dt} = \frac{Y_o - y(t)}{R} + h(t) \tag{1}$$

Where $y(t)$ is the temperature of water in the tank (°C), $h(t)$ is the heat flowing into the tank through the base heater (watts), Y_o is the temperature of the surroundings, C denotes the tank thermal capacity (Joules/°C), and R is thermal resistance between tank borders and surroundings. R and C are assumed constant. Thermal system is represented as shown in equation (1).

$$x(t) \triangleq y(t) - Y_0 \tag{2}$$

$$\dot{x}(t) = -\frac{1}{RC}x(t) + \frac{1}{C}h(t) = -\alpha x(t) + \beta h(t) \tag{3}$$

The discrete time state equation for thermal system is

$$x(k+1) = Fx(k) + gh(k) \tag{4}$$

$$F = e^{-\alpha t}; g = \beta \int_0^T e^{-\alpha t} d\tau = \frac{\beta}{\alpha}[1 - e^{-\alpha t}] \tag{5}$$

Saturation nonlinearity is included in model so that the water temperature cannot exceed some limitation. The nonlinear plant is described by

$$y(k+1) = Fy(k) + \frac{g}{1+\exp[0.5y(k)-40]} u(k) + (1-F)Y_0 \tag{6}$$

Sampling time 0. 03 sec and surrounding temperature Y_0 is assumed to be 25℃. u is the input, limited between 0 and 5 volts. Here task is to control the plant in order to follow a specified reference trajectory. The various physical parameters of the system were used in the simulation performed in this paper are Y_0=25°C, T=30 m sec, $\alpha = 1.00151 * 10^{-4}$ and $\beta = 8.67973 * 10^{-4}$

Adaptive Control Using Bfa

Bacterial foraging algorithm is applied for implementation of adaptive control [22] for a water bath temperature control system. An indirect adaptive control has been used to learn a plant model during the operation of the system. Learning rate is viewed as foraging for good model information. Multiple identifier models and social foraging are being used. The BFA search in the parameter space corresponds to getting low identification errors between the model and the plant. According to the sum of the squared identifier errors, at each time instant the model that is the best is being used in a standard certainty equivalence approach to specify a controller. Each identifier model is an affine mapping to match plant nonlinearities. The identifier model parameters represent the foragers' position. The cost function for each forager, which defines the nutrient profile, is defined to be the sum of squares of past identifier error values for each identifier model. For parameter adjustment, a foraging algorithm that is based on E. coli chemotactic behaviour is used. Here a plant model is tuned in order to specify the controller parameters. A set (population) of approximators is used to tune and optimize the set in bacterial foraging optimization algorithm. Figure 1 shows the adaptive control using BFA.

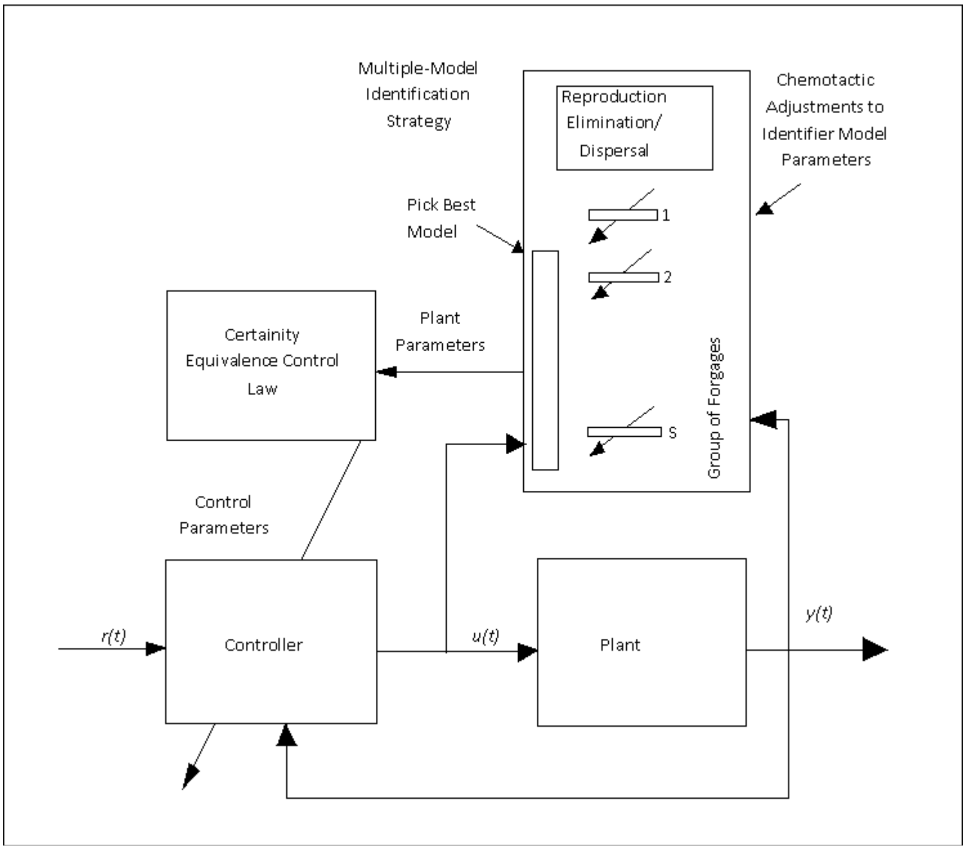

Figure1. BFA based adaptive control

The water bath system is specified as a plant and the output of the plant used is represented by:

$$y(k+d) = \alpha\big(x(k)\big) + \beta\big(x(k)\big)u(k) \tag{7}$$

Output of the system is $y(k)$.
Then state $x(k) = y(k)$
The discrete model of water bath system is:

$$y(k+1) = Fy(k) + \frac{g}{1+\exp[0.5y(k)-40]}u(k) + (1-F)Y_0 \tag{8}$$

where

$$\alpha\big(y(k)\big) = Fy(k) + (1-F)Y_0 \tag{9}$$

$$\beta\big(y(k)\big) = \frac{g}{1+\exp[0.5y(k)-40]} \tag{10}$$

$\alpha\big(y(k)\big)$ *and* $\beta\big(y(k)\big)$ are unknown smooth functions of the state $y(k)$ while $y(k+d)$ is a nonlinear function of past values of u. $\beta\big(y(k)\big)$ is requiring to be bounded away from zero. $d \geq 1$ is the delay between the input and output. For d =1,

$$y(k+1) = \alpha\big(y(k)\big) + \beta\big(y(k)\big)u(k) \tag{11}$$
$$y(k+d) = \alpha\big(y(k)\big) + \beta\big(y(k)\big)u(k) \tag{12}$$
$$y(k+d) = \alpha_u\big(y(k)\big) + \alpha_k(y(k)) + \beta_u\big(y(k)\big) + \beta_k\big(y(k)\big)u(k) \tag{13}$$

Functions $\alpha_u\big(y(k)\big)$ and $\beta_u\big(y(k)\big)$ represent the unknown nonlinear dynamics of the plant. It is these functions which require to be estimated for specifying a controller. $\alpha_k(y(k))$ and $\beta_k(y(k))$ are defined to be as known parts of the plant dynamics, these can be set to zero. $\beta\big(y(k)\big)$ is assumed to satisfy $0 < \beta_0 < \beta\big(y(k)\big)$ for some known $\beta_0 > 0$ for all $y(k)$. Estimation of an unknown ideal controller: An ideal controller is described by

$$u^*(k) = \frac{-\alpha\big(y(k)\big)+r(k+d)}{\beta(y(k))} \tag{14}$$

This linearizes the dynamics of equation (13) such that $y(k) \to r(k)$. Substituting $u(k) = u^*(k)$ in equation (13) $y(k+d) = r(k+d)$ so that tracking of reference input have been achieved within d steps. Since $\alpha\big(y(k)\big)$ or $\beta\big(y(k)\big)$ are unknown, an estimator is developed for these plant nonlinearities and used them to form an approximation to $\mathbf{u}^*(\mathbf{k})$. Using a "Certainty equivalence controller", the control input is described as

$$u(k) = \frac{-\tilde{\boldsymbol{\alpha}}\big(\mathbf{y}(\mathbf{k})\big)+\mathbf{r}(\mathbf{k}+\mathbf{d})}{\tilde{\boldsymbol{\beta}}(\mathbf{y}(\mathbf{k}))} \tag{15}$$

$\tilde{\alpha}\big(y(k)\big)$ and $\tilde{\beta}\big(y(k)\big)$ are estimates of $\alpha\big(y(k)\big)$ and $\beta(y(k))$ respectively.

The certainty equivalence controller is described with the following estimates

$$\tilde{\alpha}\big(y(k)\big) = F_\alpha\big(y(k), \theta_\alpha(k)\big) \tag{16}$$
$$\tilde{\beta}\big(y(k)\big) = F_\beta\Big(y(k), \theta_\beta(k)\Big) \tag{17}$$

Error $e(k) = r(k) - y(k)$ is not a linear function of the parameters. $e(k) = \tilde{y}(k) - y(k)$, where $\tilde{y}(k)$ is estimate of $y(k)$.

A set of approximators for α and β where the i^{th} ones are denoted by $F_\alpha(y, \theta_\alpha^i)$ and $F_\beta(y, \theta_\beta^i)$ for i=1, 2,...., S. From a foraging perspective, θ^i is viewed as the location of the i^{th} foragers in the environment. In foraging method position of the forager $\boldsymbol{\theta^i}$ is used to minimize the fitness function $\boldsymbol{J(\theta^i)}$. The $\boldsymbol{i^{th}}$ estimate of the output and identification error be

$$\tilde{y}^i(k+1) = F_\alpha\Big(y(k), {\theta^i}_\alpha(k)\Big) + F_\beta\Big(y(k), {\theta^i}_\beta(k)\Big)u(k) \tag{18}$$

and

$$e^i(k) = \tilde{y}^i(k) - y(k) \text{ for i=1, 2..., S.} \tag{19}$$

i^{th} Individual (bacteria) at time k is described as

$$\theta^i(k) = \Big[{\theta^i}_\alpha^T(k), {\theta^i}_\beta^T(k)\Big], \text{ i=1, 2,..., S.} \tag{20}$$

Fitness function is described as

$$J\left(\theta^i(k-1)\right) = \left(e^i(k)\right)^2 \tag{21}$$

$$= \left(\tilde{y}^i(k) - y(k)\right)^2. \tag{22}$$

which measures the size of the estimation error for the i^{th} estimate. It is required to minimize $J\left(\theta^i(k-1)\right)$. Forager's position in one dimension is given by θ_α and in the other dimension by θ_β so that forager's position is $\theta^i = [\theta^i_\alpha, \theta^i_\beta]^T, i = 1, 2.., S$. S is the population size of the bacteria.

Simulation Results And Discussion

Simulation have been performed in MATLAB 7. 4 [19] and scratches have been developed using Intel(R) Core(TM) 2 Duo CPU T 6400@ 2. 00GHz 1. 20GHz, 1. 99 GB of RAM. Eighteen independent runs of BFA were carried out. In the case of BFOA algorithm the best suited set of parameters were chosen after a series of hand tuning experiments. The step size $C(i, k)$ is set to be 0. 05 for all bacteria. The maximum number of steps along a good direction is N_S =4, and $\theta^i_\alpha(0)$=2, $\theta^i_\beta(0)$=0. 5, i=1, 2... S.

The Parameters used for simulation with BFA algorithms are

- *Water Bath Temperature Control System*: $p = 2, S = 1000, N_S = 4, N_C = 1000$, runlengthunit=0. 05, beta0=0. 5, beta1=0. 9, input control $0 < u(k) < 5$.

The Parameters used for simulation with GA algorithms are

- *Water Bath Temperature Control System*: $p = 2, S = 1000$, probability of crossover P_c=0. 9, mutation probability Pm=0. 05, Initial population element $\theta^i_\alpha(0)$=2, Initial population element $\theta^i_\beta(0)$=0. 5, input control $0 < u(k) < 5$.

Simulation results of water bath temperature control system using adaptive control with GA and BFA are shown in figures 2 to 10. The tracking performance of Water Bath with reference trajectory and GA based adaptive controller is shown in figure 2 which shows that after poor initialisation adaptive controller tracks the reference trajectory.

Figure 3 shows the tracking performance of Water Bath system with BFA based adaptive controller. It is observed that the tracking performance with BFA matches with the reference trajectory in steady state except some transients at t= 10 sec.

Figure 4 and 5 show the actual plant nonlinearities and its estimates. Plant nonlinearities beta estimated by GA and BFA adaptive controller is observed to be same as that of actual beta plant nonlinearities. While it has been observed that estimated alpha plant nonlinearity of BFA based adaptive control having slightly more fluctuations at high temperature as compare to that of estimated alpha plant nonlinearity using GA based adaptive control.

Figures 6 and 7 show the actual temperature and estimated value of temperature obtained with GA and BFA based adaptive controller respectively. It has been observed that estimated value of temperature using both the GA as well as BFA adaptive controller is same as that of actual temperature.

The best costs of the foragers are shown in figures 8 and 9 for GA and BFA based adaptive control of water bath system respectively. It takes time before the controller adapts, but that as the cost index decreases over time, the fitness function value decreases which indicates healthier bacteria are available for reproduction, elimination and dispersal. It has been observed that healthier bacterias are available early in GA based adaptive control as compare to BFA based adaptive control since fitness value decreases early in GA based adaptive control as compare to BFA based adaptive control.

Figure 10 shows the error between actual temperature and reference trajectory of temperature of water bath system using GA and BFA based adaptive control. Error is initially high up to 10 sec and after this the error diminishes quickly. Error is less with BFA based adaptive control as compare to GA based adaptive control. A comparison of error transient clearly shows that transients are relatively higher in magnitude for BFA based adaptive control as compared to that of GA based adaptive control.

From Table 1 it have been observed that a lower time to run the simulation imply that the BFA have a lower computational complexity than GA, so BFA can make the work with less resources.

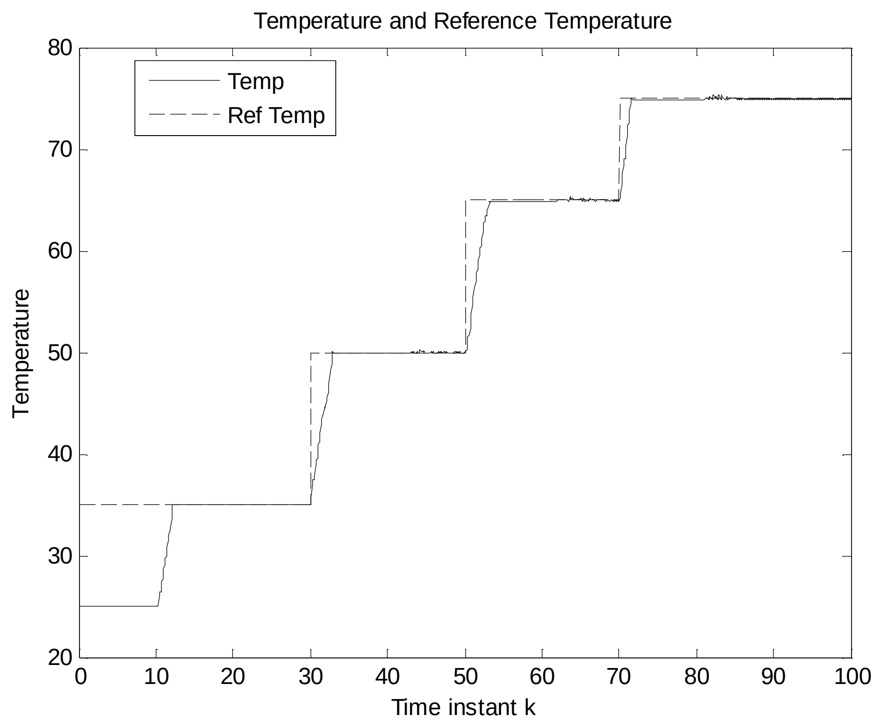

Figure2. Tracking performance of Water Bath System with GA based adaptive controller

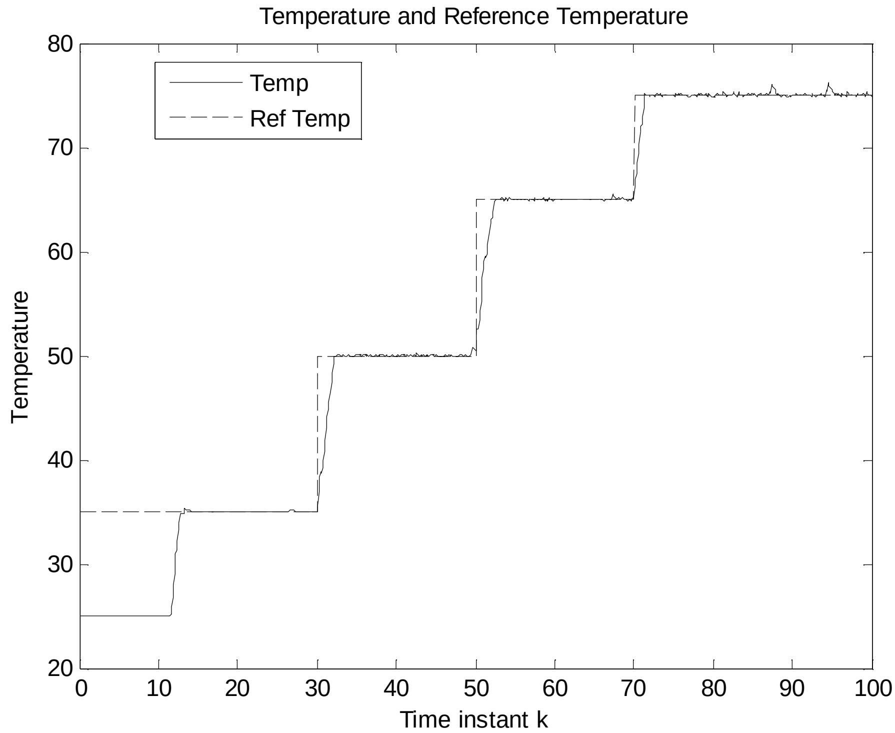

Figure3. Tracking performance of Water Bath System with BFA based adaptive controller

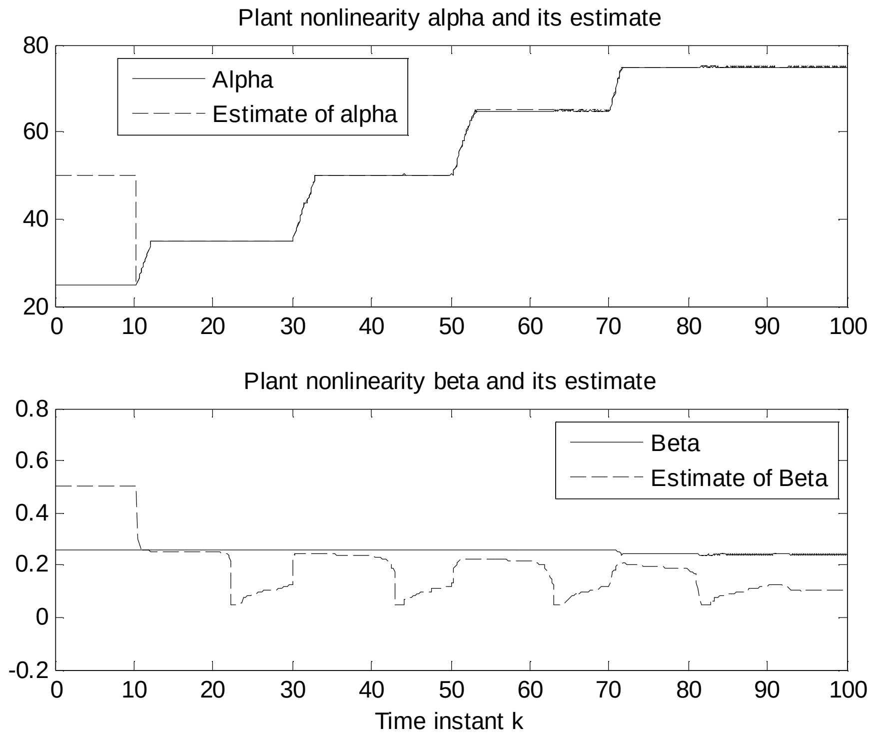

Figure4. Plant nonlinearities alpha and beta with GA based adaptive control

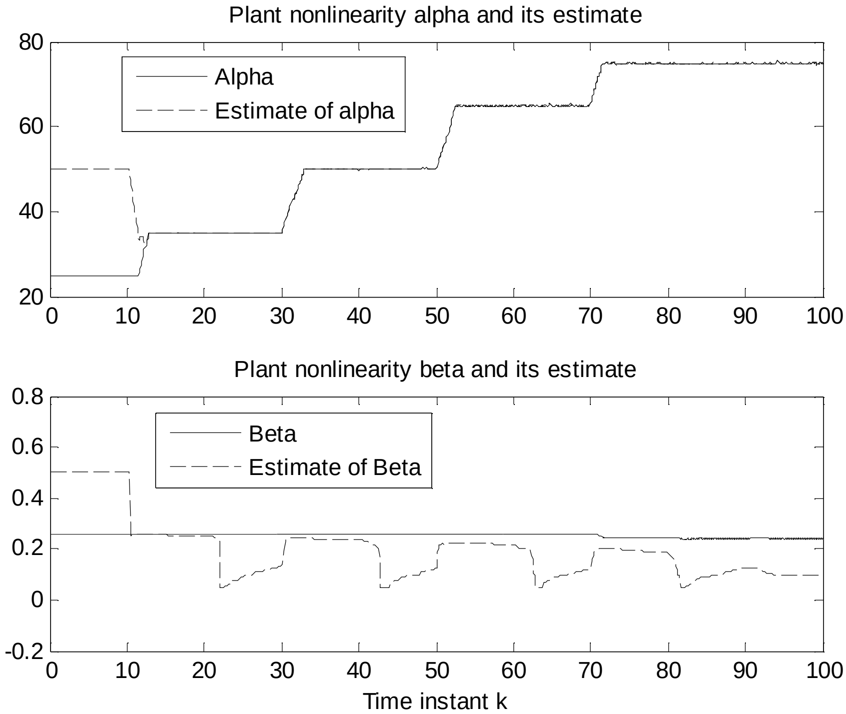

Figure5. Plant nonlinearities alpha and beta with BFA based adaptive control

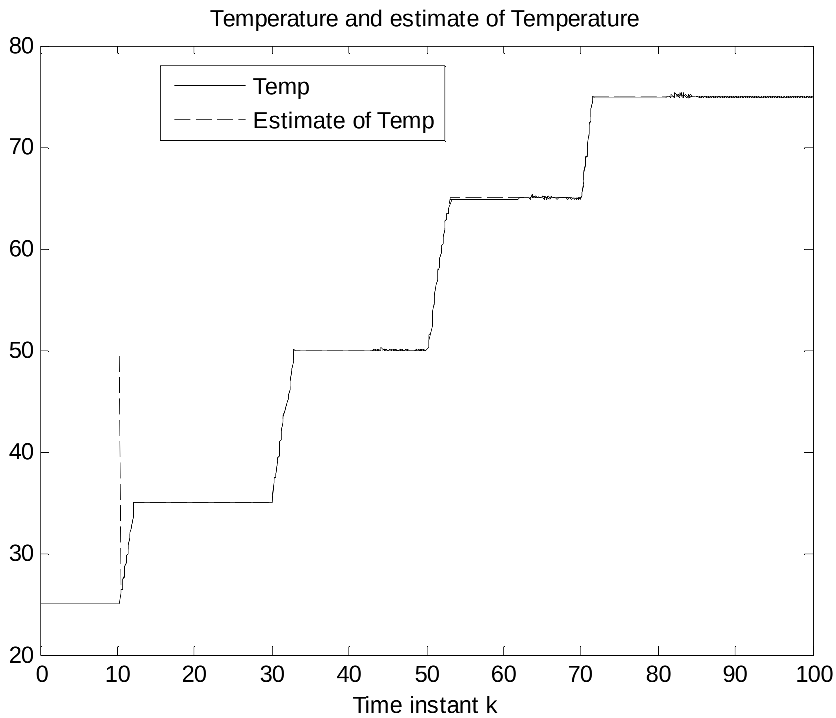

Figure6. Temperature and estimated temperature with GA based adaptive control

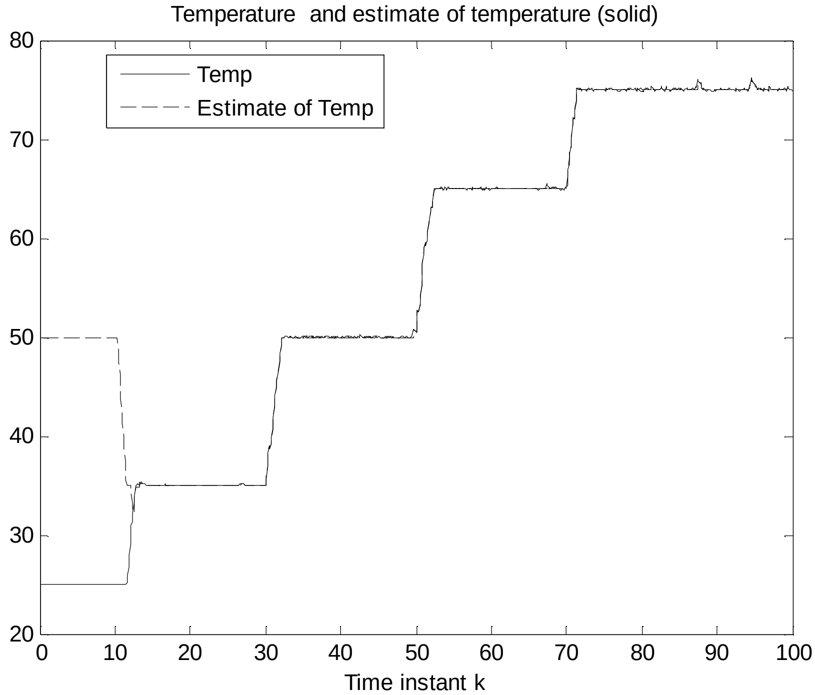

Figure7. Temperature and estimated temperature with BFA based adaptive control

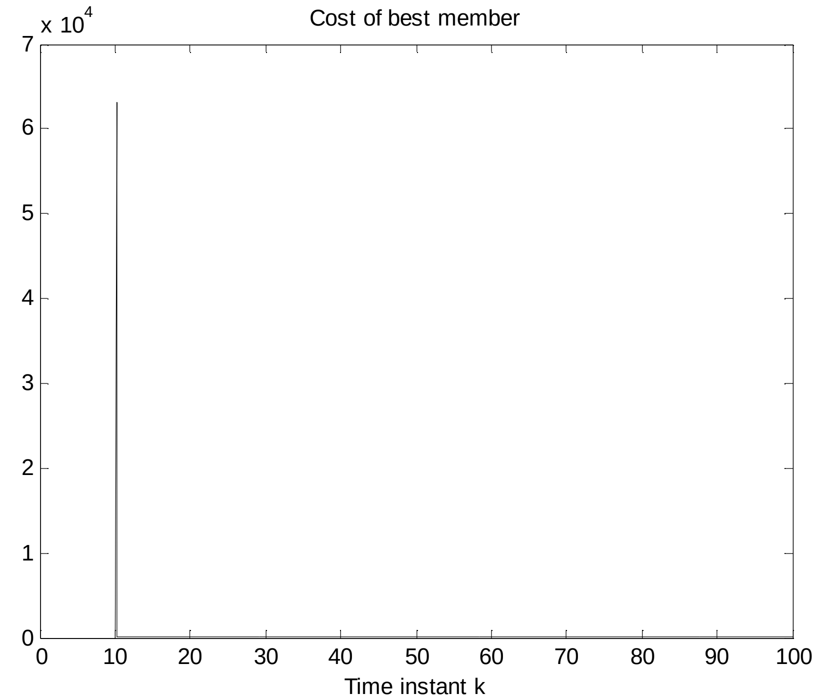

Figure8. Fitness values of best member in water bath temperature control system with GA based adaptive control

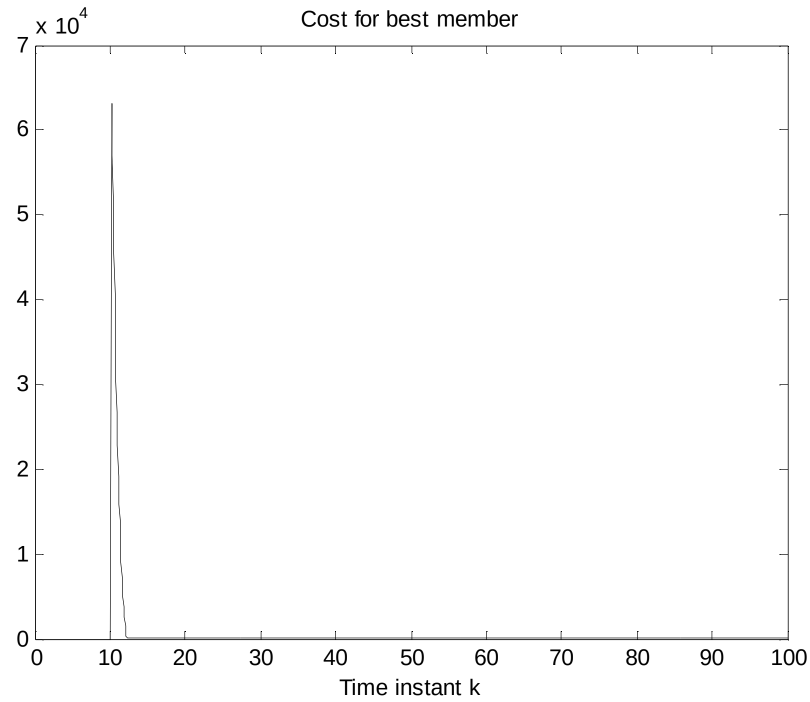

Figure9. Fitness values of best member in water bath temperature control system with BFA based adaptive control

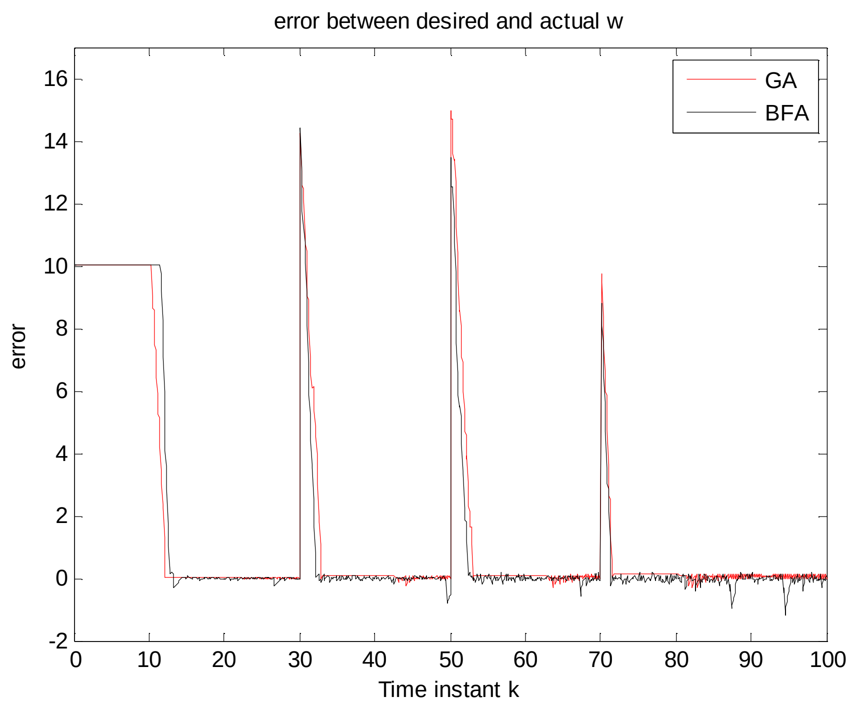

Figure10.Error between actual temperature and estimated temperature with GA and BFA based adaptive control

Table 1 Elapsed time in simulation

Non linear system	GA	BFA
Water Bath Temperature control system	88. 82 sec	42. 5630 sec

Conclusion

Bacterial foraging algorithm has been implemented on water bath temperature control system. BFA is used for identification and control of water bath system. Simulations have been carried out using MATLAB. It has been observed that BFA based adaptive controller tracks the reference trajectory of water bath system after ten seconds. Responses of water bath system have been obtained over a specified period of time and it is observed that after an initial transient period a reasonably good tracking of the reference input is obtained. It has been observed that an estimate value of the temperature of water bath system exactly matches with actual temperature. The error between actual value and estimated value of temperature reduces to zero value in steady state. Simulations for water bath system have been performed by GA also. In water bath system transient in error is less in GA adaptive control compare to BFA adaptive control. Elapsed time of simulation is less in BFA based adaptive control as compare to GA based adaptive control for the water bath system. The digital implementation of BFA based adaptive controller has fast dynamic performance to that of a GA based adaptive controller. BFA have a lower computational complexity than GA, so BFA can make the work with fewer resources.

References

[1] M. G. Song, N. C. Park, K. S. Park and Y. P. Park, “Design of a Leaf Spring Using a Genetic Algorithm, ” IEEE Trasaction on MAGNETICS, vol. 47, No. 3, pp. 590-593, March 2011.
[2] J. Jantzen, “Foundations of Fuzzy Control, ” John Wiley & Sons Ltd., 2007.
[3] K. J. Astrom and B. Wittenmark, “ Adaptive Control, ” Addison Wesley, 1989.
[4] K. D. Young, “Variable structure tracking of DC motor for high performance applications, ” IEEE Trans. Energy Conversion, vol. 4, pp 643-650.
[5] K. D. Young, “Design of variable structure model following control system, ” IEEE Trans. Automatic Contr., vol. AC -23, pp. 1079-1084, Dec. 1978.
[6] S. K. Saha, “ Introduction to Robotics, ” Tata Mc-Graw Hill, 2008.
[7] L. Sciavicco and B. Siciliano, “Modelling and Control of Robot Manipulators, ” Springer, 2000.
[8] L. A. Zadeh, ”fuzzy theory, ” university of California, Berkely, 1965
[9] K. M. Passino and S. Yurkovich, “Fuzzy Control, ” Addison Wesley, 1998.
[10] S. N. Sivanandam and S. N. Deepa, “Introduction to Genetic Algorithms, ” Springer, 2008.
[11] D. E. Goldberg, “Genetic Algorithms in search, optimization, and machine learning, ” Addison-Wesley Publishing Corporation, Inc., 1989.

[12] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamic systems using neural networks, " *IEEE Trans. Neural Networks*, vol. 1, pp. 4-27, Jan. 1990.
[13] J. S. R. Jang, C. T. Sun and E. Mizutani, “Neuro-Fuzzy and Soft Computing, ” Pearson Education, 2004.
[14] S. E. Lyshevski, “Control Systems Theory with Engineering Applications, ” Jaico Publishing House, 2003.
[15] P. Melin and O. Castillo, “Modelling, Simulation and Control of Nonlinear Dynamical Systems, ” Taylor & Francis, 2002.
[16] O. Omidvar and D. L. Elliott, “Neural Systems for Control, ” Academic Press, 1997.
[17] C. T. Lin and C. S. G. Lee, “Neural Fuzzy Systems, ” Prentice Hall P T R, 1996.
[18] C. M. Ong, “Dynamic Simulation of Electric Machinery, ” Prentice Hall P T R, 1998.
[19] S. Nakamura, “Numerical Analysis and Graphic Visualization with MATLAB, ” Prentice Hall PTR, 1996.
[20] Xiongfa MAI, Ling LI, “Bacterial Foraging Algorithm Based on Quantum-behaved Particle Swarm Optimization and Opposition-based Learning”, Journal of Computational Information Systems: 1157-1165, 10 March 2012.
[21] Z. Liu, X. Zhuang and S. Wang, “Speed Control of a DC Motor using BP Neural Networks, ” Proc. Of IEEE Conf. pp 832-835, 2003.
[22] K. M. Passino, “Biomimicry of Bacterial Foraging for Distributed Optimization and Control, ” IEEE Control System Magazine, 52-67, 2002.
[23] J. W. Ma, G. L. Zhang and H. Xie, “The optimization of Feed-Forward neural networks based on artificial fish-swarm algorithm, ” Computer Applications, vol. 24, pp. 21-23, Pct. 2004.
[24] W. M. Korani, H. T. Dorrah and H. M. Emara, “Bacterial Foraging Oriented by particle Swarm Optimization Strategy for PID Tuning”, in proc. of the 10th Annual conference on Genetic and Evolutionary Computation, USA, pp. 1823-1826, July 2008.
[25] M. Dorigo, G. D. Caro, and L. M. Gambardella, “Ant algorithms for Discrete Optimization”, Artificial Life, vol. 3, pp. 137-172, 1999
[26] J. Kennedy and R. C. Eberhart, “Particle Swarm Optimization, ” in proc. of the IEEE Int. Conf. on Neural Networks, pp. 1942-1948, 1995.
[27] Y. Chu, H. Mi, H. Liao, Z. Ji and Q. H. Wu, “A Fast Bacterial Swarming Algorithm for High-dimensional Function Optimization, ” in proc. of IEEE World Congress on Computational Intelligence, pp. 3135-3140, 2008.
[28] H. Chen, Y. Zhu and K. Hu, “Self-Adaptation in Bacterial Foraging Optimization Algorithm, ” in proc. of IEEE Int. Conf. on Intelligent System and Knowledge Engineering, pp. 1026-1031, 2008.
[29] Y. Chen and W. Lin, “An Improved Bacterial Foraging Optimization, ” in Proc. of IEEE Int. Conf. on Robotics and Biomimetics, pp. 2057-2062, Dec. 19-23, 2009

[30] S. Dasgupta, A. Biswas, S. Das, B. K. Panigrahi andA. Abrahm, “A Micro-Bacterial Foraging Algorithm for High Dimensional Optimization, ” in proc. of IEEE Int. Congress on Evolutionary Computation, pp: 785-792, 2009.
[31] H. Chen, Y. Zhu and K. Hu, “Cooperative Bacterial Foraging Algorithm for Global Optimization, ” in proc. of IEEE Int. Conf. on, pp. 3896-3901, 2009
[32] W. J. Tang, Q. H. Wu and J. R. Saunders, “Bacterial Foraging Algorithm for Dynamic Environments, ” in proc. of IEEE Int. Conf. on Evolution Computation, pp. 1324-1330, 2006.
[33] A. Abrahm, A. Biswas, S. Dasgupta and S. Das, “Analysis of Reproduction Operator in Bacterial Foraging Optimization Algorithm, ” in proc. of IEEE Int. Conf. on, pp. 14761483, 2008.
[34] U. Fang, Y. Liu and J. Liu, “A Novel Simplified Foraging Optimization Algorithm for Parameter Identification of Nonlinear System Model, ” in proc. of IEEE Int. Conf. on Automation and Logistics, pp. 798-802, 2007.
[35] T. Datta and I. S. Misra, “Improved Adaptive Bacteria Foraging Algorithm in Optimization of Antenna Arry for Faster Convergence, ” Progress in Electromagnetic Research C, vol. 1, 143-157, 2008.
[36] M. Gopal, “Digital Control and State Variable Methods, ” Tata McGraw-Hill Publishing Company Limited New Delhi, 2003.
[37] Navneet Kaur, Ramandeep Singh, Dr. R. K. Tulli, “Bacteria Foraging Based Image Segmentation”, Research Cell: An International Journal of Engineering Sciences ISSN: 2229-6913 Issue July 2012, Vol. 6.