International Journal of Control Theory and Applications

ISSN : 0974-5572

Volume 10 • Number 31 • 2017

Optimum Tuning of PI Controller Parameter using Optimization Techniques

T. Geetha[a] and S. Somasundaram[a]

[a]*Department of Electronics and Instrumentation Engineering Annamalai University Chidambaram, India*

Abstract : A simple method for finding the optimal value of PI controller parameters among several stable PI parameters obtained from the global stability region using the stability boundary locus approach. A new optimization algorithm is introduced in this paper to compute the stabilized value of PI controller. The GSO based PI Controller satisfies the conditions of the stable PI controller. The best value of the PI controller parameters will have the shortest settling time, rise time and reduced overshoot. Besides stability, the following requirements such as Robust performance, Set point tracking is included in the control system design. The simulation results for the class of stable FOPTD process model of the proposed optimization algorithm is then compared with the other optimization algorithms.

Keywords: *Controller; Setpoint; Optimization; Tracking; Stability.*

1. INTRODUCTION

For many decades, many researchers have contributed their findings towards the design and development of PID control design. The PID controller is widely used in process control industries due to its simplicity, robustness and its principle is easier to understand than many other advanced controllers and makes it easy to regulate the process output. The general performance of PID controller is satisfactory in many applications and it is an attractive research area. The majority of controllers used in industry are PI/PID type. The controller designs are usually performed based on an approximate model, the nature of the process and the performance requirements. Many important results have been recently shown on computation of all stabilizing P, PI controllers followed by the publication of work by Ho et al[1][2]. An alternative fast approach to this problem is based on the Nyquist plot has been given in [3][4].A new approach is shown to compute the stabilizing PI controllers in the parameter plane, (Kp, Ki) plane.The controller design will satisfy the time domain specification, stability and robustness. For a well designed control system the following requirements needed besides stability are disturbance atttenuation, set point tracking and robust performance. It computes the stable PI controller parameter for a given SISO control system with reduced computational time. All the set of stabilizing values of the parameters of PI controller are obtained in the (Kp, Ki) plane using the stability boundary locus approach. The GSO based PI Controller finds the optimal Kp, Ki value which satisfies the requirements to obtain the shortest settling time, fastest rise time and reduced overshoot. The tuning rules utilized for the comparison are Firefly Optimization Algorithm (FA) and Bacterial Foraging Optimization Algorithm (BF).

2. DESIGN OF THE CONTROLLER

2.1. Computation of stabilizing PI controller parameters

Consider the Single-Input Single-Ouput (SISO) control system shown in fig 1 where

$$G(s) = (X(S))/(Y(S)) \tag{1}$$

G(s) is the plant being controlled and C(s) is a PI controller given in the form

$$\begin{aligned} C(s) &= \{K_P + K_i / (s)\} \\ &= [K_p S + K_i] / S \end{aligned} \tag{2}$$

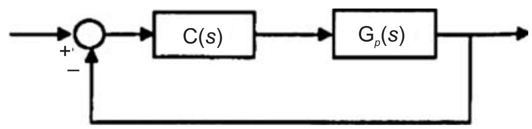

Figure 1: A SISO control system

The stabilizing PI controllers are computed by Tan's method published in [5] [6]. In this method stability boundary locus is plotted by substituting $s = j\omega$ in eqn (1) and decomposing the numerator and denominator polynomials into their even and odd parts which gives

$$G(j\omega) = \frac{X_e(-\omega^2) + j\omega X_0(-\omega^2)}{Y_e(-\omega^2) + j\omega Y_0(-\omega^2)} \tag{3}$$

The expression of closed loop characteristic polynomial is obtained and equate the real and imaginary parts to zero for getting the proportional gain and integral gain.

$$K_p = \frac{Z_5(\omega)\, Z_4(\omega) - Z_6(\omega)\, Z_2(\omega)}{Z_1(\omega)\, Z_2(\omega) - Z_2(\omega)\, Z_3(\omega)} \tag{4}$$

$$K_i = \frac{Z_6(\omega)\, Z_1(\omega) - Z_5(\omega)\, Z_3(\omega)}{Z_1(\omega)\, Z_4(\omega) - Z_2(\omega)\, Z_3(\omega)} \tag{5}$$

Where

$$\begin{aligned} Z_1(\omega) &= -\omega_2 X_0(-\omega^2) \\ Z_2(\omega) &= X_e(-\omega^2) \\ Z_3(\omega) &= \omega X_e(-\omega^2) \\ Z_4(\omega) &= \omega X_0(-\omega^2) \\ Z_5(\omega) &= \omega^2 Y_0(-\omega^2) \\ Z_6(\omega) &= -\omega Y_e(-\omega^2) \end{aligned} \tag{6}$$

The stability boundary locus is plotted by solving the equations (4) & (5) simultaneously. Making Ki = 0 the (Kp, Ki) plane is divided into stable and unstable regions. The stabilizing Kp & Ki parameters will be available in the stable region and finding the best PI controller parameter is carried out using the optimization techniques.

Example 1: Consider a first order plant with time delay given by transfer function

$$G(s) = \frac{6.6}{42s+1} e^{-3s} \tag{7}$$

The time delay is approximated by pade approximation technique

$$= \frac{6.6}{42s+1} \times \frac{2-3s}{2+3s}$$

The stabilized PI Controllers are computed by the Tan's method by decomposing the even and odd parts in (3)

$$K_p = \frac{-3385.8\omega^2 + 26.4}{-392.04\omega^2 - 174.24} \tag{8}$$

$$K_i = \frac{2494\omega^4 - 1188\omega^2}{-392.04\omega^2 - 174.24} \tag{9}$$

$$0 = \frac{2494\omega^4 - 1188\omega^2}{-392.04\omega^2 - 174.24} \tag{10}$$

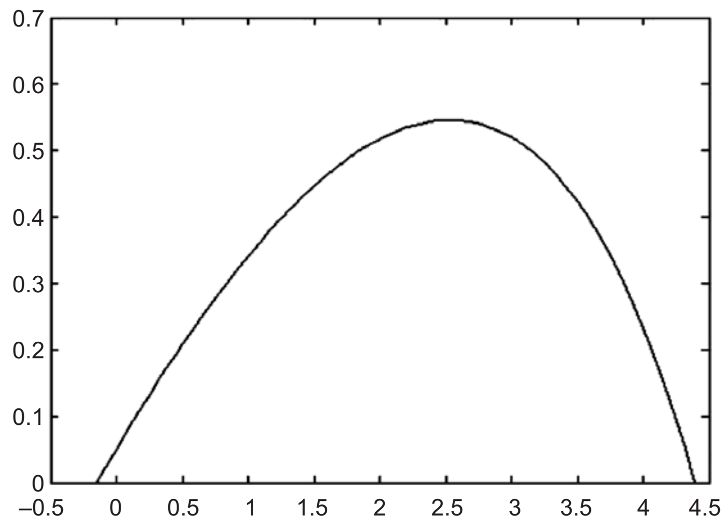

Figure 2: Stability Boundary Locus

Making = 0, ω value will be calculated. Substituting ω value in (8) & (9) solving these simultaneously the stability boundary locus will be easily plotted as shown in Fig (2). The set of K_p and K_i values are obtained and group search optimization algorithm is proposed to find the optimal value to achieve reduced overshoot, shortest settling time and rise time.

2.2. Bacterial Foraging Optimization Algorithm

Bacterial Foraging Optimization Algorithm (BFOA) is proposed by Kim, Abraham and cho (2007) [7]. Bacteria find for nutrients to maximize energy obtained per unit time. Individual bacterium too communicates with others by sending signals [8]. A bacterium takes foraging decisions by considering two factors. The process, in which a bacterium moves by taking small steps while searching for nutrients, is called chemotaxis. The main idea of BFOA is imitating the chemotactic movement of virtual bacteria in the problem search space [9]. Bacterial Foraging optimization theory is explained by Chemotaxis, Swarming, Reproduction & Eliminational-Dispersal.

Chemotaxis process personate the movement of an E.coli cell by swimming and plunging 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 between these two modes of operation for the entire lifetime.

Swarming is an process in which an interesting group behavior has been noticed for several motile species of bacteria containing E.coli and S. Typhimurium, where intricate and stable spatio-temporal patterns (swarms) are produced in semisolid nutrient medium[10]. A group of E.coli cells arrange themselves in a travelling ring by moving up the nutrient gradient and infuse a semisolid matrix with a single nutrient chemo-effecter. The cells are stimulated by a high level succinate, release an attractant aspertate, which helps them to aggregate into groups and move as concentric patterns of swarms with high bacterial density.

Reproduction involves the least healthy bacteria eventually die when each of the healthier bacteria (which yields 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.

Eliminational-Dispersal process involves Gradual or sudden changes in the local environment make the bacterium population to live may occur due to various reasons. Events can occur such that all the bacteria in a region are killed or a group is dispersed into a new part of the environment. For example, a significant local rise of temperature may kill a group of bacteria that are currently in a region with 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 new location. During the long time period, these events spread various types of bacteria into all parts of the environment from our intestines to hot springs and underground environments. To stimulate this phenomenon in BFOA some bacteria are selected at random with a very small probability while the new replacements are randomly initialized over the search space. Elimination and dispersal events have the ability of possibly destroying chemotactic progress also they assist the chemotaxis process and dispersal may place the bacteria near good food sources. From a broad perspective, elimination and dispersal are parts of the population-level long-distance motile behavior.

2.3. FireFly Optimization Algorithm

Firefly is an insect that mostly produces short and rhythmic flashes that produced by a process of bioluminescence(11). The function of the flashing light is to attract partners (communication) or attract the prey and also a protective warning towards the predator. This intensity of light is the factor of the other fireflies to move towards the other firefly. The light intensity is varied at the distance from the eyes of the beholder[12]. It is safe to say that the light intensity gets decreased when the distance increase. The light intensity have the influence towards air absorbed by the surroundings leads to the less intensity as the distance increases (Yang, 2010[13]). Firefly algorithm follows three idealize rules, 1) Fireflies have attraction towards each other regardless of gender. 2) The attractiveness of the firefly is correlated with the brightness of the firefly makes the less attractive firefly tend to move forward to the more attractive firefly. 3) The brightness of firefly depends on the objective function (Yang, 2010).

2.4. GSO based Stable PI Controller

GSO is a resource searching inspired by group living theory which was developed by (S.He et al., 2009). The inhabitant of the GSO algorithm is called group and the singular number in the inhabitant is called a member. The head angle φ_i^K where *i* is the member and *k* is the searching iteration (drum) in the search space calculates the search direction. While starting with GSO, certain parameters need to be initialized. It includes lower bound value, upper bound value, population size, number of producers, number of scroungers, maximum turning angle, maximum pursuit angle and number of iterations. The population size is the number of K*p* and K*i* parameters and the number of dimension is two. All possible sets of controller parameter values are adjusted so as to minimize the objective function, it is the error criterion. The number of such criteria are available and here controller's performance is evaluated in terms of Integral Square Error(ISE) error criteria. The global best solution was selected having the minimum error.L = [min(*kc*) min(*ki*)] U = [max(*kc*) max(*ki*)] Where L is the lower bound value and U is the upper bound value containing minimum K*c*, K*i* values and maximum K*c*, K*i* values.

The initial head angle $\varphi°$ of apiece inhabitant is set to be $\pi/4$. The maximum pursuit angle θ_{max} is fit to be $\pi/(a^2)$. The maximum turning angle α_{max} is fit to be $\theta_{max}/2$. The maximum pursuit distance l_{max} is derived by

It is possible to obtain more accurate results by fine tuning the φ°, θ_{max} , l_{max} and α_{max}. In GSO, a group contains three forms of members they are producer, scroungers and rangers. Producer and scroungers whose procedures are based on the PS model and dispersed members who support random walk motions. For simplicity a single producer is assumed at every searching drum and the rest are scroungers and dispersed members. All scroungers use the resources found by the producer. In the search process if the scrounger finds better resource then it will switch to be a producer and all the scroungers and the producer in the previous searching drum will perform scrounging strategies. Producer and scroungers don't vary in their relevant phenotypic characteristics so they can switch between the two roles [14] [15]. At each iteration (drum) a group member present in the propitious area and the best fitness value is selected as the producer. At each iteration (drum) a group member present in the propitious area and the best fitness value is selected as the producer. In GSO, vision is the main scanning mechanism employed by producer and scanning strategies were characterized by maximum pursuit angle and maximum pursuit distance. First the producer will scan at zero degree and then first point at zero degree[16][17].

$$X_2 = X^k_p + r_1 l_{max} D^k_p(\varphi^k) \tag{11}$$

The second point in the right hand side of the producer.

$$X_r = X^k_p + r_1 l_{max} D^k_p(\varphi^k + r_2 \theta_{max} / 2) \tag{12}$$

The third point in the left hand side of the producer.

$$X_1 = X^k_p + r_1 l_{max} D^k_p(\varphi^k - r_2 \theta_{max} / 2) \tag{13}$$

Where $r_1 \in R^1$ is a normally distributed random number with mean 0 and standard deviation 1, $r_2 \in R^{n-1}$ is a uniformly distributed random sequence in the range (0,1) θ_{max} is the maximum pursuit angle and l_{max} is the maximum puruit distance. Again producer will search for the better resource if it finds the best fitness value it will move to that point otherwise it will stay in the existing position and turn its head to the new angle

$$\varphi^{k+1} = \varphi^k + r_2 \alpha_{max} \tag{14}$$

Where $\alpha_{max} \in R^1$ is maximum turning angle.

If the producer is unable to find the better area after a iterations then it will turn its head to zero degree.

$$\varphi^{k+a} = \varphi^k$$

Where a $\in R^1$ is a constant. During every searching drum, a number of group members are selected as scroungers. The scroungers will search for opportunities to join the resources found by the producer.

During the K^{th} iteration resource searching behaviour of the i^{th} scrounger walk towards the producer.

$$l_i = a.\, r_1\, l_{max}$$

And move to the new point $$x_i^{k+1} = x_i^k + l_i D^k_i(\varphi^{k+1})$$

The flowchart of the GSO algorithm is shown in Fig.3. A generic social foraging model, *e.g.*, PS model, was employed as the framework to derive GSO. The producer of GSO is quite similar to the global best particle of PSO, the major difference is that the producer performs producing which differs from the strategies performed by the scroungers and the dispersed member. while, in PSO each individual performs the same searching strategy. In GSO the producer remembers its head angle when it starts producing. In GSO, the search is simply conducted by turning the head to a new angle. The scanning procedure of GSO is like a simplified direct search method. Various strategies are adopted to restrict their search and GSO algorithm employs bounded search space[18]. In this strategy when a member is outside the search space it will turn back to search space by setting the variables that violated the boundary criteria. Optimization is a process of seeking optima in a search space, is analogous to the resource searching process of animals in nature As long as the limit of time delay is from 0.1 to 50, GSO works well for all the FOPTD models.

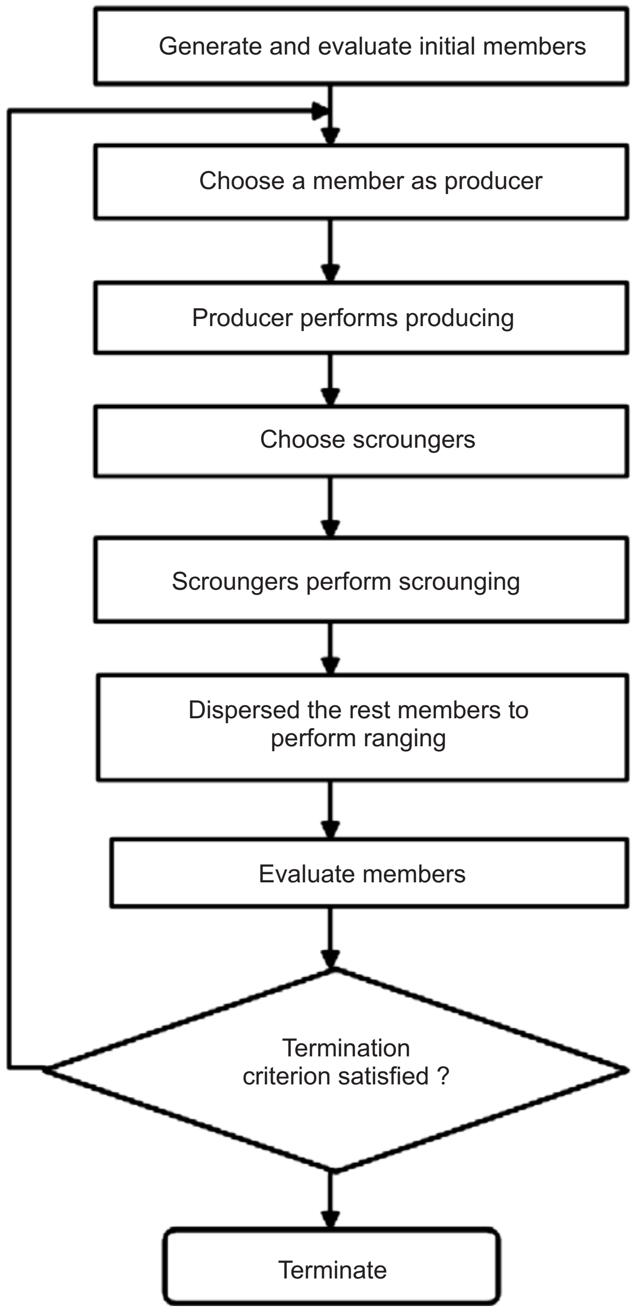

Figure 3: Flowchart of the GSO Algorithm

3. SIMULATION RESULTS

The PI values obtained by the GSO are compared with the results derived from FA and BF algorithm in various perspectives, viz Servo Regulatory responses and Robust performances. All the simulations were implemented using MATLAB.

1. $G(S) = \frac{1}{s+1} e^{-10s}$

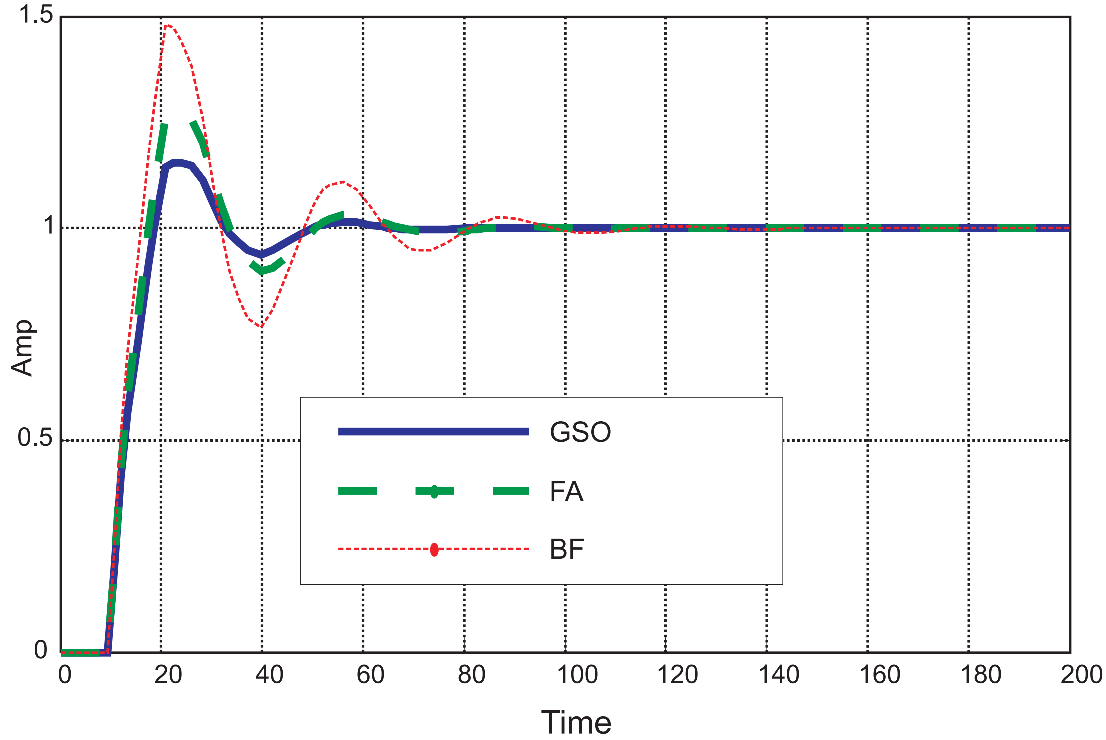

Figure 4: Response of the best K_p K_i values for t_d = 10

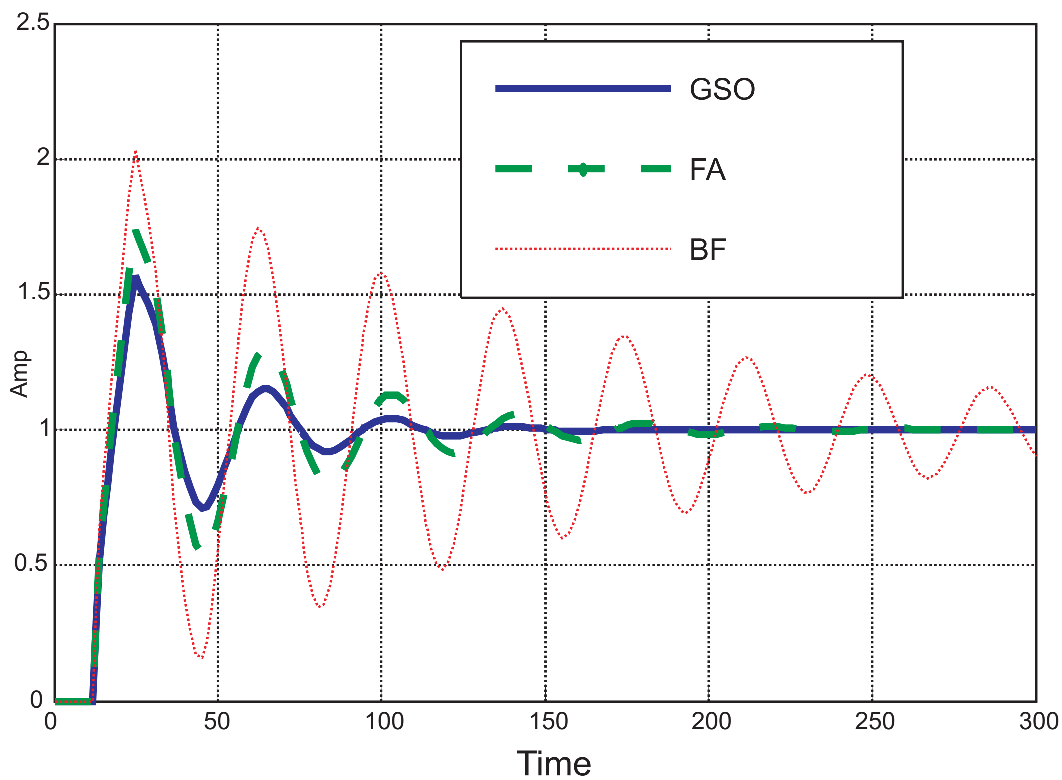

Figure 5: Response of the Robustness for + 20% in K_p T_d and -20% in τ_p

Table 1
Comparison of Performance Index and Time Domain Specifications of GSO tuned, FA tuned and BF tuned PI Controllers

S.No	*Tuning Rule*	*ISE*	*IAE*	t_r	t_s	*O.S*
1.	GSO	12.05	15.06	7	60	0.18
2.	FA	12.4	17.42	6.5	80	0.23
3.	BF	13.64	21.68	6	120	0.49

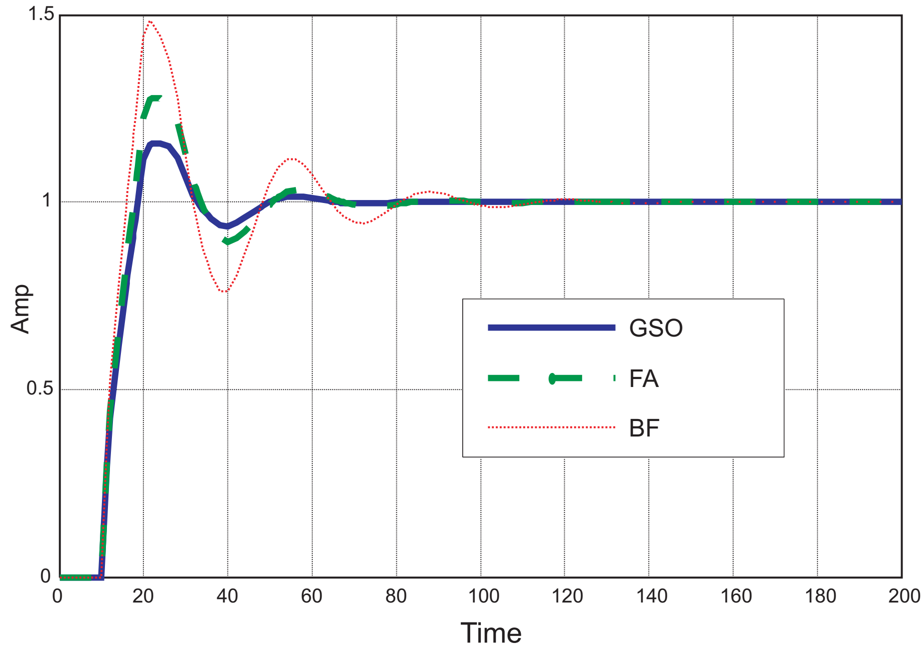

Figure 6: Servo Response of the GSO tuned, FA tuned and BF tuned PI Controllers

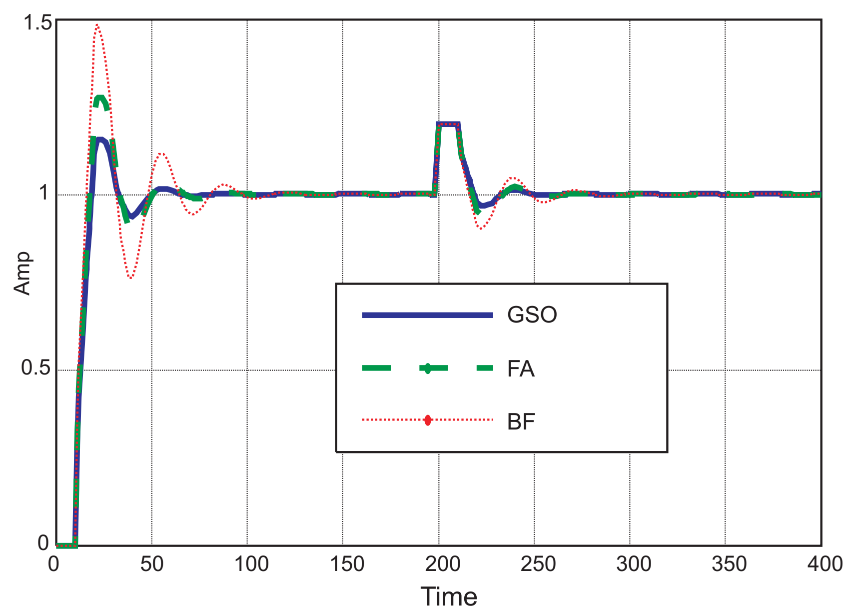

Figure 7: Regulatory response of the GSO tuned, FA tuned and BF tuned PI Controllers

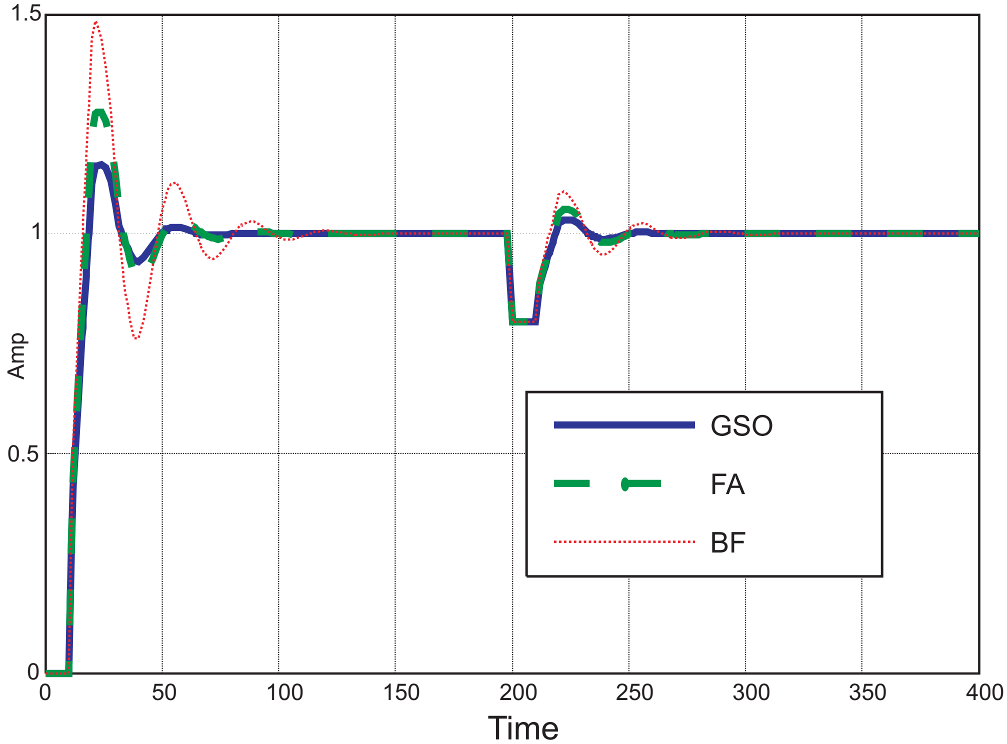

Figure 8: Regulatory response of the GSO tuned, FA tuned and BF tuned PI Controllers

2. $$G(S) = \frac{2s+1}{(10s+1)(0.5s+1)}e^{-s}$$

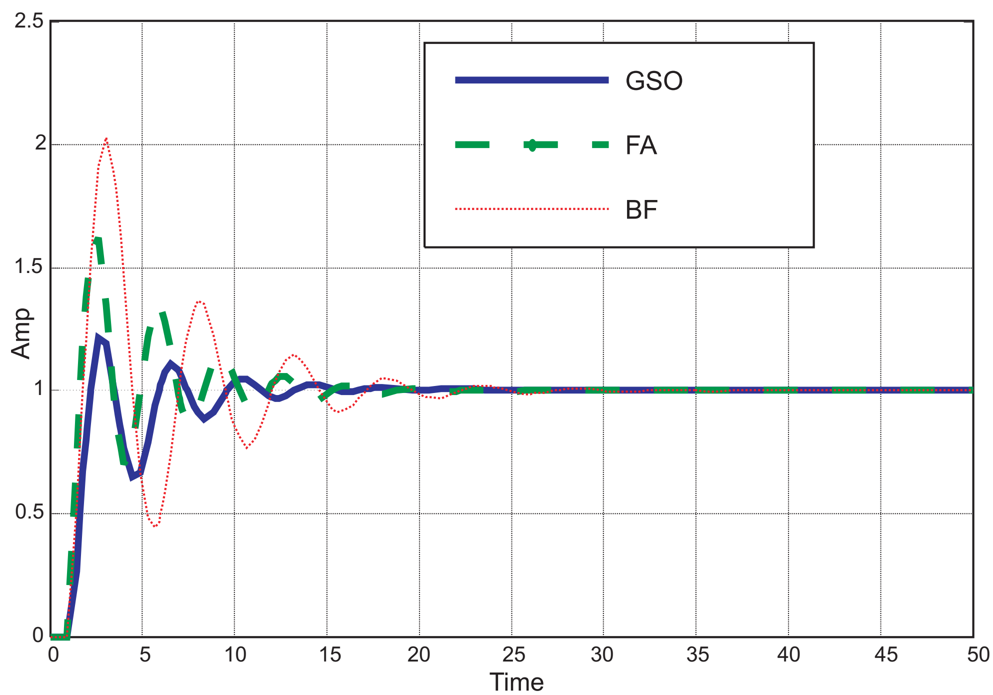

Figure 9: Response of the best K_pK_i values for $t_d = 1$

Table 2
Comparison of Performance Index and Time Domain Specifications of GSO tuned, FA tuned and BF tuned PI Controllers

S.No	*Tuning Rule*	*ISE*	*IAE*	t_r	t_s	*O.S*
1.	GSO	1.701	2.833	1	20	0.2
2.	FA	1.768	3.282	0.6	25	0.6
3.	BF	3.336	5.578	0.7	30	1

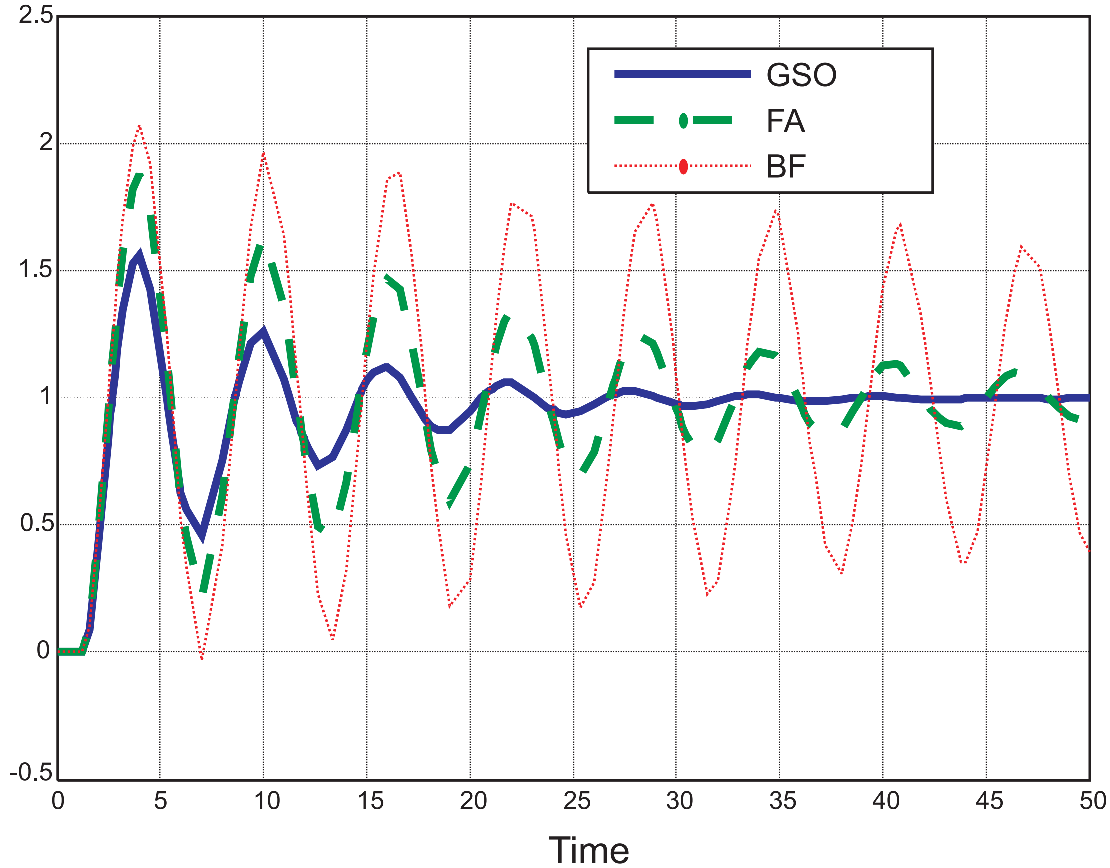

Figure 10: Response of the Robustness for + 20% in $K_p T_d$ and -20% in τ_p

4. CONCLUSION

In the present work the design and implementation of GSO based PI controller PSO based PI controller and BF based PI controller for FOPTD models have been presented. Based on the simulation results it is concluded that, the performance of the GSO controller is most superior compared to the PSO based PI controller and BF based PI controller. The GSO based controller works well for FOPTD models having time delay upto 50. The simulation responses reflect the effectiveness of the GSO based controller in terms of time domain specifications. The performance index under the various error criterions of the GSO based controller is always less than the FA and BF based PI controller.

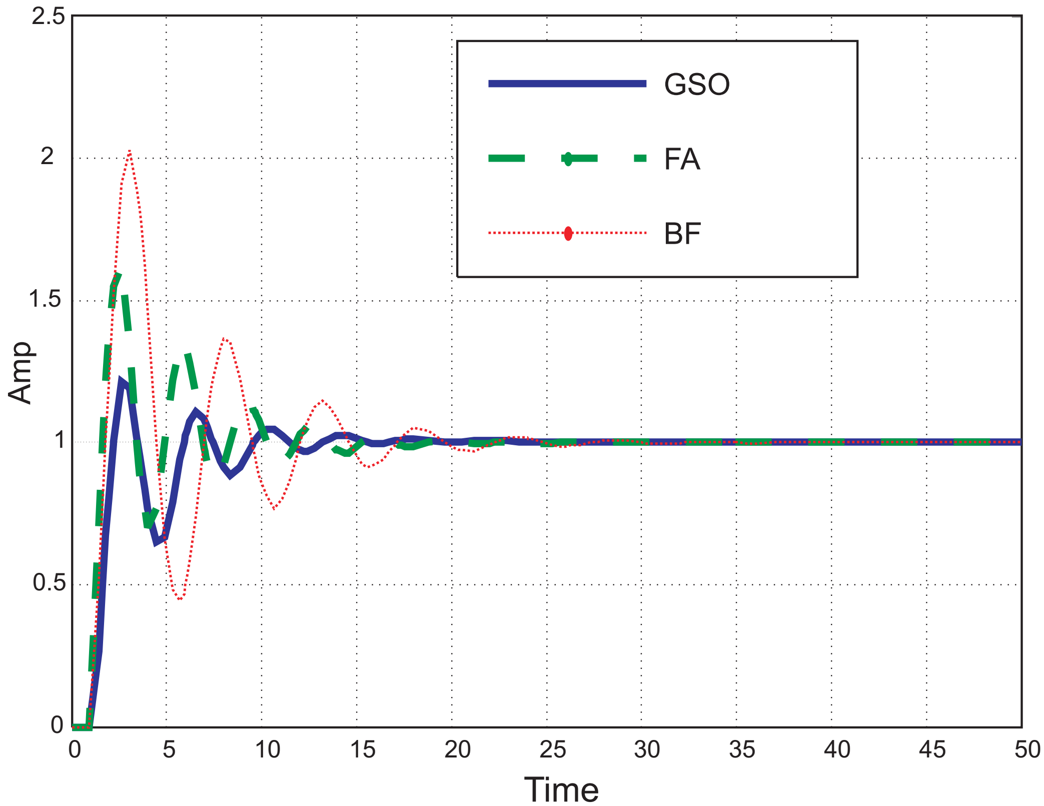

Figure 11: Servo Response of the GSO tuned, FA tuned and BF tuned PI Controllers

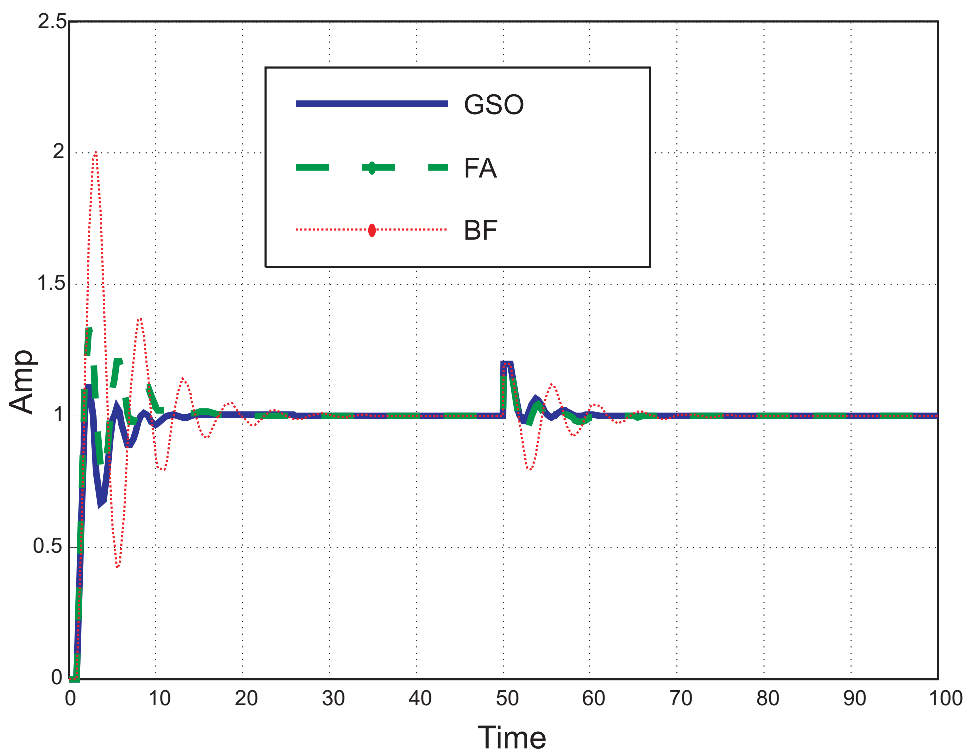

Figure 12: Regulatory response of the GSO tuned, FA tuned and BF tuned PI Controllers

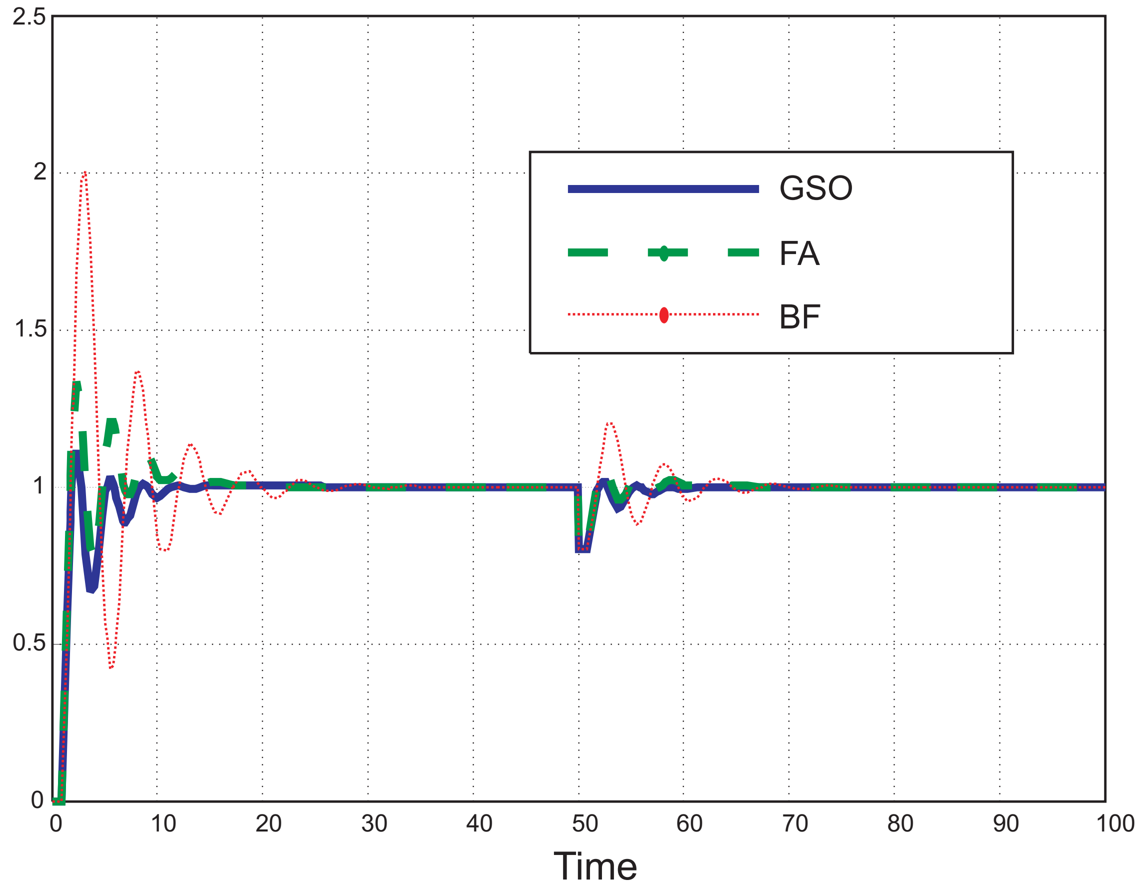

Figure 13: Regulatory response of the GSO tuned, FA tuned and BF tuned PI Controllers

REFERENCES

[1] M.T. Ho, A. Datta, and S.P. Bhattacharya, "A new approach to feedback stabilization ," Proc. of the 35th CDC ., pp.4643-4648,1996.

[2] M .T. Ho, A. Datta, and S.P. Bhattacharya, "A new approach to feedback design part I: PI and PID controllers," Dept of Electrical Eng.,Texas A&M Univ., College Station TX, Tech. Report TAMU-ECE97-001-A,1997

[3] N.Munro and M.T. Soylemez, " Fast calculation of Stabilizing PID controllers for uncertain parameter systems," Proc. Symposium on Robust Control, Prague,2000

[4] M. T. Soylemez, N.Munro, and H.Baki, " Fast calculation of stabilizing of PID controllers," Automatica , Vol. 39, pp. 121-126,2003.

[5] N. Tan, and I. Kaya, "Computation of stabilizing PI controllers for interval systems" presented at the 11th Mediterranean Conference on Control and Automation, Rhodes, Greece, 2003.

[6] N. Tan, I. Kaya, C. Yeroglu, and D. P. Atherton, "Computation of stabilizing PI and PID controllers using the stability boundary locus" *Energy Conversion and Management*, vol. 47, no. 18-19, pp. 3045-3058,2006.

[7] Alam, M., Claviez, M., Oesterhelt, D., & Kessel, M. (1984), Flagella and motility behaviour of square bacteria. The EMBO Journal,13(12).2899 2903.

[8] Berg, H. C. (1993). Random Walks in Biology , New Expanded Edition. Princeton, NJ: Princeton University press.

[9] Berg, H. C., & Brown, D. A. (1972). Chemotaxis in Escherichia Coli Analysed by Three Dimensional Tracking. Nature, 239, 500 504.

[10] Blat, Y., & Eisenbach, M.(1995). Tar-Dependent and Independent Pattern Formation by Salamonella typhimurium Journal of Bacteriology,117, 1683 1691.

[11] Yang X. 2010. Nature-Inspired Metaheuristic Algorithms. 2nd ed. Frome: Luniver Press.

[12] Gandomi A., Yang X. and Alavi A. 2011. Mixed variable structural optimization using firefly algorithm. Computers and Structures. 89(23): 2325-2336

[13] H.banati and M. Bajaj, (2011),"Firefly based feature selection approach", IJCSI International Journal of Computer Science Issues, vol. 8,Issue 4, no. 2,pp. 473- 480

[14] E. Bonabeau, M. Dorigo, and G. Theraulza, "Inspiration for optimization from social insect Behavior," *Nature*, vol. 406, pp. 39–42, Jul. 2000.

[15] J. Najemnik and W. S. Geisler, "Optimal eye movement strategies in visual search," *Nature*, vol. 434, pp. 387–391, Mar. 2005.

[16] I. D. Couzin, J. Krause, N. R. Franks, and S. A. Levin, "Effective leadership and decision-making in animal groups on the move," *Nature*,vol. 434, pp. 513–516, Feb. 2005..

[17] M. Locatelli. A note on the griewank test function. *Journal of Global optimization*, 25(2):169–174, 2003..

[18] M. Clerc and J. Kennedy. The particle swarm: Explosion, stability,and convergence in a multi-dimensional complex space. *IEEE Trans.on Evolutionary Computation*, 6(1):58–73, 2002.