COMPARATIVE STUDY IN OPTIMISING PID CONTROLLER USING BEES ALGORITHM AND FIREFLY ALGORITHM

N.S.S. Hameed, W.A.F.W. Othman, A.A.A. Wahab, S.S.N. Alhady

School of Electrical & Electronics Engineering,
Universiti Sains Malaysia, 14300 Nibong Tebal, Penang, Malaysia.

Corresponding Author's Email: wafw_othman@usm.my, aeizaal@usm.my, sahal@usm.my

ABSTRACT: The robustness, efficiency and effectiveness of Bees Algorithm and Firefly Algorithm were compared by analyzing the minimization of the objective function and the step responses of the closed-loop systems with Proportional-Integral-Differential (PID) controller. An objective function comprises peak overshoot, M_p, steady-state error, e_{ss}, rise time, T_r and settling time, T_s compares acceptable solutions and selects the best one with respect to an optimized design. The parameters of PID controller *i.e.* K_p, K_i and K_d must be properly selected as the selection affects the transient response of a system. The best combination of parameters reduces problems such as nonlinearities encountered by industrial plants. Therefore, PID parameters were determined by analyzing the average and standard deviation of cost returned by the cost function. After analyzing the minimization of cost function and the dynamic performance specifications of the closed-loop systems, both algorithms had good performance in general. Nevertheless, Firefly Algorithm outperformed Bees Algorithm in terms of fast convergence rate. The good performance of Firefly Algorithm reflected on the importance of its parameters; brightness, β, and attractiveness, I which dictated the search for the best solutions in a short time.

KEYWORDS: *PID Controller; Bees Algorithm; Firefly Algorithm; Optimization; Swarm Intelligence*

1.0 INTRODUCTION

Algorithms were developed in earlier years to carry out the optimization procedures. Most of optimization problems produce many local optima solutions. Therefore, selecting good optimization technique is important because it ensures that the method does not only searches in the neighborhood of the best solution as to avoid mislead in search process. Therefore, optimization algorithm is the procedure to find the best solution from the set of all feasible solutions.

Basically, the aim of optimization algorithm is to minimize or to maximize the value of an objective function. The algorithm will continuously search for the best solutions until a stopping criterion is satisfied. Despite that, the optimization algorithm should have a mechanism to balance between global and local search [1]. A suitable optimization algorithm would involve in finding peaks in a fitness landscape or valleys in the cost landscape.

In many real-life applications, the optimization functions may not behave well mathematically, and it is a well-known challenge in searching for a global optimal solution [2]. Although many optimization methods such as Genetic Algorithm, Simulated Annealing and some other algorithms have been developed, but these algorithms have some weaknesses such as getting trapped to local optima and slower execution time.

In short, due to the computational drawbacks of mathematical techniques and methods such as complex derivatives, sensitivity to initial values, and the large amount of enumeration memory required, researchers relied on meta-heuristic algorithms based on simulations and some degree of randomness to solve optimization problems [3]. But these meta-heuristic approaches are not very accurate, and they do not always return the optimal solution, in most cases they give a near optimal solution with less effort and time than the mathematical methods [4].

2.0 METHODOLOGY

In this study, two different plant transfer functions are chosen from research papers. Based on research paper by Pareek *et al.* in [5], G_{p1} (namely *System A*) is used and another plant transfer function, G_{p2} (namely *System B*) from a research paper by Wadhwani and Verma in [6]. Eq. (1) and Eq. (2) represent the plant transfer functions used in this work, *System A* and *System B*, respectively.

$$G_{p1}(s) = \frac{1}{s^4 + 6s^3 + 11s^2 + 6s} \tag{1}$$

$$G_{p2}(s) = \frac{0.01}{0.005\,s^3 + 0.06\,s^2 + 0.1001\,s} \tag{2}$$

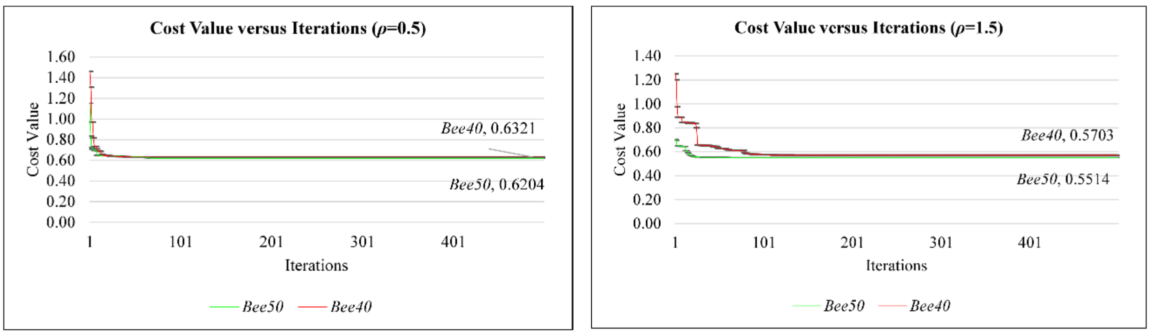
A cost function (Eq. 3) to evaluate the response of the system is selected from [7-9]. The cost function is chosen because it is defined such a way that it selects best PID parameters with good dynamic performance in terms of the four criteria in the function.

$$F = \left(1 \quad e^{\rho}\right)\left(M_p + e_{ss}\right) + \left(e^{\rho}\right)\left(T_s \quad T_r\right) \tag{3}$$

Two sets of control parameters are selected and compared for each algorithm. Each algorithm are then executed for 200 iterations for 20 runtimes. Graphs of average cost, standard deviation of cost are plotted. Then the closed-loop systems are tested with the best PID parameters with the lowest cost. Finally, the dynamic performance specifications of both systems with both algorithms are compared.

3.0 RESULTS

Figure 1(a) and 1(b) show graph of cost value plotted against iterations for *Bee50* and *Bee40*, using ρ = 0.5 and ρ = 1.5 respectively. **Table 1** shows the lowest cost value obtained by *Bee50* and *Bee40* for ρ = 0.5 and ρ = 1.5. The set of control parameters which returned the best cost is highlighted and hence, the set was chosen for the comparison study.



(a) (b)

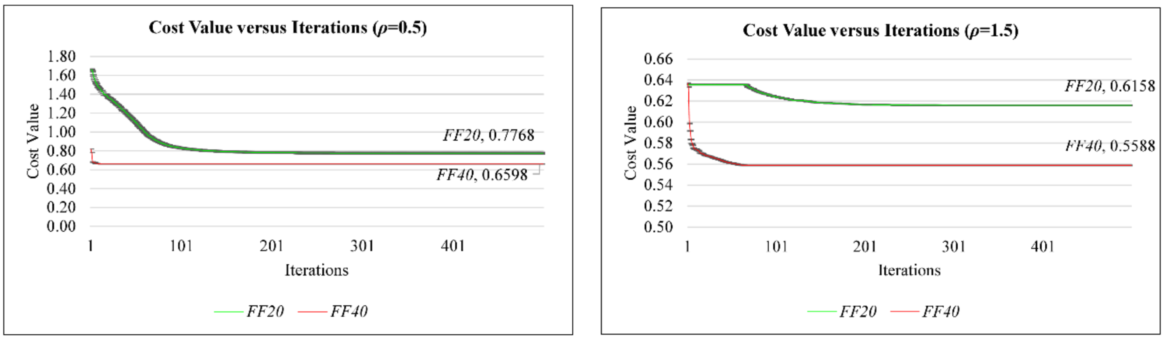
Fig. 1: Plot of cost value versus iterations for Bee Algorithm, *Bee50* and *Bee40* using (a) ρ=0.5 and (b) ρ=1.5

Table 1: Lowest cost value for Bee50 and Bee40 (ρ=0.5 and ρ=1.5)

ρ value	*Bee50*	*Bee40*
ρ=0.5	**0.6204**	0.6321
ρ=1.5	**0.5514**	0.5703

Figure 2(a) and 2(b) show graph of cost value plotted versus iterations for *FF20* and *FF40* (refer **Table 2**) using ρ = 0.5 and ρ = 1.5 respectively. **Table 2** shows the lowest cost value obtained by

FF20 and *FF40* for ρ =0.5 and ρ = 1.5. The set of control parameters which returned the best cost was highlighted and hence, the set was chosen for the comparison study.



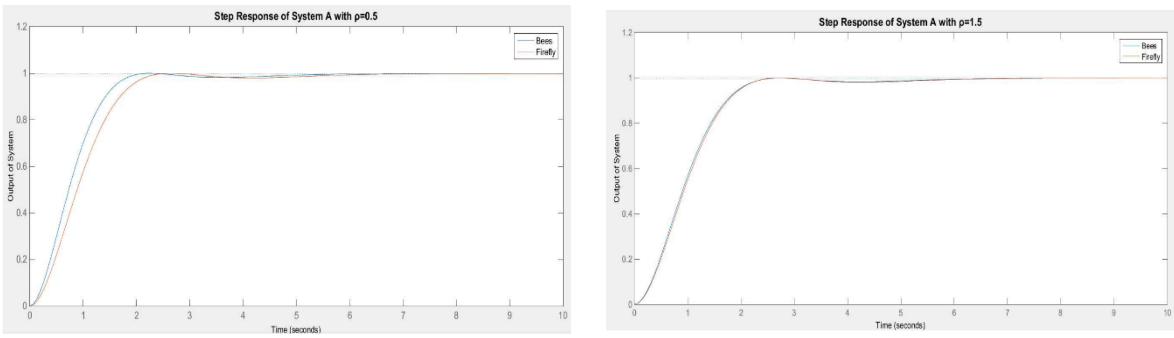
(a) (b)

Fig. 2: Plot of cost value versus iterations for Firefly algorithm, *FF20* and *FF40* using (a) ρ=0.5 and (b) ρ=1.5

Table 2: Lowest cost value for *FF20* and *FF40* (ρ=0.5 and ρ=1.5)

ϱ value	*FF20*	*FF40*
ϱ=*0.5*	0.7768	**0.6598**
ϱ=*1.5*	0.6158	**0.5588**

Figure 3 depicts corresponding step response of System A using PID gains tuned using Bees Algorithm and Firefly Algorithm with ϱ=0.5 and ϱ=1.5 respectively. While, **Table 3** shows the corresponding dynamic performance specifications of System A for (a) ϱ=0.5 and (b) ϱ=1.5.



(a) (b)

Fig. 3: Step response of System *A* with Bees Algorithm and Firefly Algorithm with (a) ρ=0.5 and (b) ρ=1.5

Table 3: Dynamic performance specifications of System *A* for (a) ϱ=0.5 and (b) ϱ=1.5

	ϱ=0.5		ϱ=1.5	
	Bees Algorithm	Firefly Algorithm	Bees Algorithm	Firefly Algorithm
M_p (%)	0.000241	0	0	0
T_s (s)	1.8107	2.1442	2.2085	2.2153
T_r (s)	1.1645	1.3808	1.4142	1.4257
e_{ss} (s)	0.5000	0.5000	0.5000	0.5000
Peaktime (s)	2.2262	2.6279	10.1319	2.7280

3.0 CONCLUSION

Based on the step responses and dynamic performance specifications, Bees Algorithm and Firefly Algorithm have equally good dynamic performance specifications with slight variation in cost values. It can be shown that both algorithms managed to find nearly optimal solutions which returned low cost values.

4.0 ACKNOWLEDGEMENT
All authors have disclosed no conflicts of interest, and authors would like to thank the Ministry of Education Malaysia and Universiti Sains Malaysia for supported the work by Fundamental Research Grant Scheme (Grant number: USM/PELECT/6071239).

5.0 REFERENCES
[1] Said, G. A. E. N. A., Mahmoud, A. M., & El-Horbaty, E. S. M. (2014). A comparative study of meta-heuristic algorithms for solving quadratic assignment problem. arXiv preprint arXiv:1407.4863.

[2] Tang, R., Fong, S., Yang, X. S., & Deb, S. (2012, August). Wolf search algorithm with ephemeral memory. In Seventh International Conference on Digital Information Management (ICDIM 2012) (pp. 165-172). IEEE.

[3] Lee, K. S., & Geem, Z. W. (2004). A new structural optimization method based on the harmony search algorithm. Computers & structures, 82(9-10), 781-798.

[4] Lee, K. S., & Geem, Z. W. (2005). A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. Computer methods in applied mechanics and engineering, 194(36-38), 3902-3933.

[5] Pareek, S., Kishnani, M., & Gupta, R. (2014, August). Optimal tuning of PID controller using meta heuristic algorithms. In 2014 International Conference on Advances in Engineering & Technology Research (ICAETR-2014) (pp. 1-5). IEEE.

[6] Wadhwani, S., & Verma, V. (2013). Evolutionary computation techniques based optimal PID controller tuning. International Journal of Engineering Trends and Technology, 4(6), 2529-34.

[7] Gaing, Z. L. (2004). A particle swarm optimization approach for optimum design of PID controller in AVR system. IEEE transactions on energy conversion, 19(2), 384-391.

[8] Wong, C. C., Li, S. A., & Wang, H. Y. (2009). Optimal PID controller design for AVR system. Tamkang Journal of Science and Engineering, 12(3), pp. 259-270.

[9] Danaei, H., & Khajezadeh, A. (2015). Optimal Design of PID Controller Using New Version of Bee’s Algorithm for Quarter-Car Active Suspension System. Academie Royale Des Sciences D Outre-Mer Bulletin Des Seances, 4(4), 119-125.