

# Optimal Tuning of PD controllers using Modified Artificial Bee Colony Algorithm

M. R. Hashim<sup>1,2</sup>, Hyreil A. K.<sup>1</sup> & M. O. Tokhi<sup>3</sup>

<sup>1</sup>Universiti Teknikal Malaysia Melaka, Melaka, Malaysia

<sup>2</sup>University of Sheffield, Sheffield, United Kingdom

<sup>3</sup>London South Bank University, London, United Kingdom  
ruzaini@utem.edu.my

**Abstract**—This paper presents an investigation of PD controller tuning using a modified artificial bee colony algorithm (MABC). The main purpose of this work is to apply and investigate the performance of MABC in tuning the PD controller of single link manipulator system (SLMS) in comparison with the original ABC. The objective of the MABC algorithm is to minimize the error by using mean square error (MSE) as an objective function. The proposed algorithm has also been tested in three benchmark functions with different dimensions to checked the robustness of the algorithm in different problems surface. The result shows that the MABC able to tune the controller to their best optimum value.

**Index Terms**— Artificial Bee Colony; Local Search; Single Link Manipulator System, PD Controller.

## I. INTRODUCTION

In recent year, Optimization techniques based on the adaptation of nature phenomenon and biological behavior are called evolutionary algorithm become popular among researchers. These algorithms types are free gradient algorithm where it does not need differentiation operation to perform the best search strategies. Thus, it can use in many optimization problems and able to give a good quality of solution in each iteration. Nature phenomenon and biological behavior adaptation optimization algorithm can be categorized as a metaheuristic algorithm. the definition of metaheuristic algorithm is a way to solve general problems [1] using a structure in heuristic optimization strategies, and it may create the efficient way to perform the searching in the search space.

In metaheuristic algorithm, the search of an unknown optimal point in search space (feasible area) is in random and some parameter needs to be set in initial stages by the user in order to make it work properly. The most important factor that gives the metaheuristic optimization merit in solving problems is a balanced strategy between exploration and exploitations. The exploration in the optimization context is the phase that the searching of the new solution in the unexplored area within the feasible region. It can increase the chance to find a good solution (not guaranteed). While in exploitation, is the process to focus the searching in a small area within a good solution found in the previous exploration. Too much exploration can increase the convergence speed, but it may result in it trapped at local optimum and may not improve the solution. Searching a solution based on the previously discovered solution during exploitation phases can increase the chance to find a much better solution within it, but if too much exploitation it will cause slower convergence

toward the global optimum. In reality, there is no guarantee that the balanced strategies between exploration and exploitation can produce a really good solution and fast convergence speed because there will be a trade-off between both of them [2] [3].

As mention before, the Evolutionary algorithm (EA) is part of the metaheuristic algorithm widely used by the researcher to find the optimum solution in many problems (e.g., single objective and multi-objective problems). Evolutionary algorithms are based on the adaptation of biological system (bio-inspired algorithmic) or natural phenomenon (natural-inspired algorithms). A bio-inspired algorithmic is associate based on the behavior of living organisms (e.g., animal behavior) whereas a nature-inspired is based on natural phenomena (e.g., a pattern of lightning). EA are optimization heuristic, and it used stochastic search based on the evolution theory. EA starts with randomly distribute the possible solution in the search space, after that it will perform the exploitation with the good solution found before and do the exploration in the uncharted region toward the global optimum. The good about EA, they have a good strategies in self-adapting and self-organizing and also have strong capability of collaboration during evolution and behavior [4], have a different type of character in the algorithm structure (modularity), good improvement mechanism and able to find a good optimal solution [5][6]. Other factor may increase the performance is deterministic, and randomization approaches through the searching process [7].

There is well known EA called swarm intelligent algorithm. Swarm intelligent algorithm basically based on the interaction behavior between members in a population of the swarm. Example of Swarm intelligent algorithm is a Genetic algorithm (GA) adopted from the process of genetic evolution, particle swarm optimization (PSO) that mimics the communication behavior of fish and bird flocking [8], Artificial bee colony algorithm (ABC) based on the behavior honey bee foraging the food source [9]. Bat algorithm (BA) is based is the echolocation behavior of bats [10]. Inspired from Escherichia coli bacteria foraging pattern, [11] proposed bacteria foraging algorithm (BFA), Ant Colony Optimization (ACO) proposed by Dorigo based on trails pattern by ants [12], Firefly Algorithm (FA) [13]. The example of natural-inspired optimization algorithms are lightning search algorithm (LSA) based on the natural phenomenon of lightning [14], inspired by the improvisation characteristics of a musician, Geem et al developed harmony search algorithm [15][16]. Galaxy-based search algorithm [17]. Spiral dynamic algorithm (SDA)[18] inspired from a natural spiral pattern such as a spiral galaxy, tornado and DNA

molecule. SDA has also been modified to an enhanced version to increase the capability of original SDA [19][20][21]. All of the mentioned algorithm widely used to solve different field of problems such as in engineering, medical, finance, science and many more. These algorithms also have been tested in nonlinear systems; complex fitness landscape constrained problems or multi-objective problems and the output from this algorithm showing promising to solve those problems.

In this paper, the original ABC has been modified and tested in single link manipulator system (SLMS). The algorithms have also been tested in 3 benchmark functions. The enhanced version of ABC is called modified artificial bee colony (MABC)

## II. ARTIFICIAL BEE COLONY ALGORITHM

The artificial bee colony (ABC) algorithm was developed by D. Karaboga [22] in 2007 based on the behavior of real honey bee foraging the food (flower nectar). To search for the best optimal value, ABC using the bee as an agent to search the best food (solution) in the search space. These agents are divided into three (3) group; employed bee (EB), Onlooker bee (OB) and scout bee (SB). EB has a role to share the information about the quality of food source with OB and their only foraging 1 specific food source until it depleted. While OB is the bee that is waiting for the information gather by EB inside the hive at the dance floor. They will choose which the food source they want to exploit based on EB information. The EB will change to the SB when their food sources are depleted. SB role is to explore and find a new food source. The EB population number is half of the colony while OB population number is another half. The number of EB is equal to the number of a possible solution (food source).

Initially, the initial population  $P(\text{Iter}=0)$  are randomly distributed in the search area. Where  $x_i (i = 1, 2, \dots, nFS)$  is a possible solution and this solution is a vector of D-dimensional [23]. Where nFS is a number of food source and D is optimization parameter numbers. Next, the EB produced a new position based on the local information. Then, it will test the fitness value of that solution. In this stage, they will use a greedy selection method. If the new position can provide a better solution, EB will keep the new position and discard the old one. Otherwise EB will keep the previous food position. The position modification is followed Equation 1.

$$v_{ij} = x_{ij} + rand(-1,1)(x_{ij} - x_{kj}) \quad (1)$$

From (1); when the difference between  $x_{ij}$  and  $x_{kj}$  become smaller, the perturbation on the current position also decreased. This mean the step size is reduced the search approaches to the optimal point.

After all, EB completes the search. They will go back to the hive and share the local information with OB. This information contains the quality of solution (food source). The OB will choose the food source based on the probability associated with the food source (FS). They tend to choose the foods that have the highest recruiting probability. The calculation of the probability of food source selection by OB can be calculated by the following equations:

$$Fitness(i) = \begin{cases} \frac{1}{1 + f(\bar{S}_i)}, & f(\bar{S}_i) \geq 0 \\ 1 + abs[f(\bar{S}_i)], & f(\bar{S}_i) < 0 \end{cases} \quad (2)$$

$$probability(i) = \frac{Fitness(i)}{\sum_{i=1}^{FS/2} Fitness(i)} \quad (3)$$

After choosing the food source. OB will modify the current position using Equation 1. Same with EB, they will compare the fitness of the previous position and modification position. OB will choose the best solution based in their fitness and keep the best position in their memory and forget the un-improved solution. The food source will be abandoned when the solution cannot be improved anymore after the predetermined cycle limit. When this happened, the EB will change role into SB and finds new food source as defined by Equation 4:

$$x_i^j = x_{min}^j + rand(0,1)(x_{max}^j - x_{min}^j) \quad (4)$$

The searching for the best optimum solution is continued until it reaches maximum predetermine iteration number.

## III. MODIFIED ARTIFICIAL BEE COLONY ALGORITHM

In the proposed algorithm, the step size was modified to ensure the searching movements for the best solution are more dynamic. Thus, the exponential constant,  $\partial_w$  has been introduced in Equation 1.  $\partial_w$  able to control the maximum step size. The exponential characteristic can lead the searching toward to the optimal point. The mathematical equation for MABC is given as: -

$$v_{ij} = x_{ij} + rand(-1,1)(x_{ij} - x_{kj})\partial_w + rand(0,1)(x_j - x_{kj})\partial_w \quad (5)$$

$$\partial_w = \frac{a_1 - a_2}{1 + \frac{b_2}{exp(b_1|fitness(i) - minF|)}} + a_2 \quad (6)$$

where  $\partial_w$  is exponential constant,  $b_1$  and  $b_2$  are constant to be tune heuristically.  $a_1$  is the minimum step size and  $a_2$  is the maximum step size where the value must be chosen between 0 – 1. Tunable  $b_1$  and  $b_2$  have a role to ensure the  $\partial_w$  not to big and not to small. To further improvement, the best fitness,  $x_j$  was introduced to fine tune the step size and to ensure the algorithm not trapped at local optimum. The other process is followed the original ABC.

## IV. BENCHMARK VALIDATION

Three benchmark functions have been used to validate the performance of the proposed algorithm. The benchmark functions involved are Sphere, Ackley, and grienwark. The results collected are compared with the original ABC. The test is run 30 times with different value of dimension (10,30,50 and 70). The ABC and MABC parameter are shown in Table 1:

Table 1  
Algorithm Parameter Setting

Parameter	value
Number of Foods (NF)	40
Number of Bee limit	80
iteration	100
	1600

The statistical performance of ABC and MABC are shown in Table 2. From the result, the MABC outperformed ABC in all benchmark functions. The result also shown the deviation from the average value for all benchmark function showed that MABC output are stable.

Table 2  
Statistical Performance Results of the ABC and MABC

Function	Dim	ABC	MABC	
Sphere	10	Mean	8.8E-17	<b>1.9E-195</b>
		SD	1.61E-17	<b>0</b>
		Worse	1.04E-16	1.9E-194
		Best	6.12E-17	0
	30	Mean	9E-16	<b>4.53E-23</b>
		SD	1.57E-16	<b>9.04E-23</b>
		Worse	1.18E-15	2.44E-22
		Best	7.04E-16	4.84E-31
	50	Mean	6.06E-13	<b>4.27E-19</b>
		SD	6.17E-13	<b>5.02E-19</b>
		Worse	1.7E-12	1.47E-18
		Best	1.1E-13	2.29E-20
70	Mean	1.51E-08	<b>2.6E-18</b>	
	SD	1.05E-08	<b>2.02E-18</b>	
	Worse	3.89E-08	6.73E-18	
	Best	3.04E-09	4.52E-22	
Ackley	10	Mean	9.06E-15	<b>8.88E-16</b>
		SD	1.72E-15	<b>0</b>
		Worse	1.15E-14	8.88E-16
		Best	7.99E-15	8.88E-16
	30	Mean	2.08E-10	<b>8.88E-16</b>
		SD	1.45E-10	<b>0</b>
		Worse	4.84E-10	8.88E-16
		Best	8.11E-11	8.88E-16
	50	Mean	2.76E-05	<b>4.8E-15</b>
		SD	1.49E-05	<b>3.91E-15</b>
		Worse	5.36E-05	1.51E-14
		Best	1.18E-05	8.88E-16
70	Mean	2.61E-03	<b>2.93E-14</b>	
	SD	9.69E-04	<b>2.77E-14</b>	
	Worse	4.47E-03	8.62E-14	
	Best	1.46E-03	4.44E-15	
Grienwank	10	Mean	5.92E-03	<b>0</b>
		SD	6.68E-03	<b>0</b>
		Worse	1.72E-02	0
		Best	6.78E-13	0
	30	Mean	7.48E-04	<b>0</b>
		SD	2.36E-03	<b>0</b>
		Worse	7.48E-03	0
		Best	1.11E-16	0
	50	Mean	9.37E-04	<b>0</b>
		SD	2.36E-03	<b>0</b>
		Worse	7.42E-03	0
		Best	1.66E-10	0
70	Mean	1.98E-03	<b>0</b>	
	SD	4.99E-03	<b>0</b>	
	Worse	1.60E-02	0	
	Best	1.42E-06	0	

## V. SINGLE LINK MANIPULATOR SYSTEM

For further testing, MABC was used to tune the PD controller of a single link manipulator system (SLMS). In this

paper, the controller used to test the SLMS model is based on controller design by Supriyono et al. [24]. The PD controller for the SLFMS can be formulated as: -

$$u(t) = K_p e(t) - K_v \frac{d\theta(t)}{dt} \quad (7)$$

where  $u(t)$  is the controller output,  $K_p$  is proportional gain,  $K_v$  is derivative gain,  $e(t)$  is the error  $\theta(t)$  is angular Displacement. The schematic diagram of the controller is show in Figure 1.

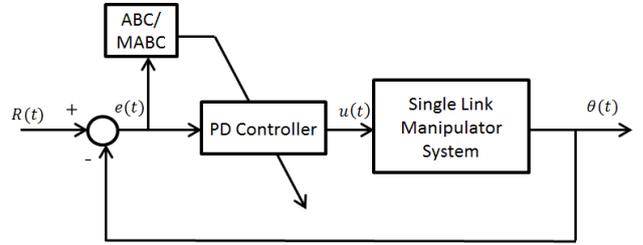


Figure 1 SLMS controller schematic diagram

The Objective of the proposed algorithm is to tune the PD gain until it gets the minimum hub angle different between actual and set point hub angle. The objective function in this test will use Mean square error (MSE). The objective function can be formulated as: -

$$J = \left( \frac{1}{N} \sum_{k=1}^N (e(t))^2 \right) \quad (8)$$

where, J is an objective function, the error  $e(t)$  is the difference angle between the reference hub angle and actual hub angle. The smallest value of objective function shows the hub angle of the SLMS is close to the reference hub angle value.

## VI. RESULTS

In this section, the performance of MABC in tuning PID controller of SLMS will be discussed. The reference hub angle used in this test is 30 degrees. The parameter setting in this test are shown in Table 3;

Table 3  
Algorithm Parameter Setting

Parameter	value
Number of Foods (NF)	20
Number of Bee limit	40
	100

The hub angle response of the SLFMS show in figure 2 and Table 4 indicates that MABC able to tune SLFMS to the reference point with low overshoot and reasonable rise time. In other hand, ABC able to reach the reference point faster than MABC but the overshoot quite high. There are no steady state error occurred in both algorithms. Therefore, this result shows the capability of MABC in solving engineering problems are good.

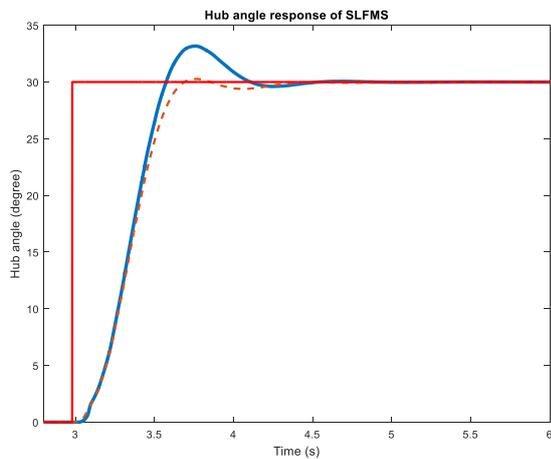


Figure 2 Hub angle response of SLMS

Table 4  
Numerical Result of Controller Parameter

Measurement	ABC	MABC
Kp	1.6997	1.7561
Kd	0.5140	0.6076
Overshoot	10.55%	0.51%
Rise time,s	360.67ms	403.19ms

## VII. CONCLUSION

The modification on the step size during searching for a new position in EB and OB stage has been proposed in this paper. Validations with three benchmark functions have been carried out. The result shows the MABC able to navigate the solution toward the optimal point faster than ABC. Furthermore, the performances of MABC to tune the PD controller of SLMS have shown it outperformed the original ABC with small overshoot and reasonable rise time.

## REFERENCES

- [1] T. Weise, "Global Optimization Algorithms—Theory and Application," URL <http://www.it-weise.de>, Abrufdatum, vol. 1, p. 820, 2009.
- [2] L. J. Eshelman, R. A. Caruana, and J. D. Schaffer, "Biases in the crossover landscape," *Third Int. Conf. Genet. Algorithms*, no. November 2014, pp. 10–19, 1989.
- [3] N. Muttill and S.-Y. Liong, "Superior Exploration–Exploitation Balance in Shuffled Complex Evolution," *J. Hydraul. Eng.*, vol. 130, no. 12, pp. 1202–1205, 2004.
- [4] T. Feng, Q. Xie, H. Hu, L. Song, C. Cui, and X. Zhang, "Bean Optimization Algorithm Based on Negative Binomial Distribution," in *Advances in Swarm and Computational Intelligence: 6th International Conference, ICSI 2015, held in conjunction with the Second BRICS Congress, CCI 2015, Beijing, China, June 25-28, 2015, Proceedings, Part I*, Y. Tan, Y. Shi, F. Buarque, A. Gelbukh, S. Das, and A. Engelbrecht, Eds. Cham: Springer International Publishing,

- 2015, pp. 82–88.
- [5] I. Kassabalidis, M. a. El-Sharkawi, R. J. . I. Marks, P. Arabshahi, and A. A. Gray, "Swarm intelligence for routing in communication networks," *GLOBECOM'01. IEEE Glob. Telecommun. Conf. (Cat. No.01CH37270)*, vol. 6, pp. 3613–3617, 2001.
- [6] H. Garg, "Solving structural engineering design optimization problems using an artificial bee colony algorithm," *J. Ind. Manag. Optim.*, vol. 10, no. 3, pp. 777–794, 2013.
- [7] X. Yang, "Review of Meta&#45;Heuristics and Generalised Evolutionary Walk Algorithm," *Int. J. Bio-Inspired Comput.*, vol. 3, no. 2, pp. 77–84, Apr. 2011.
- [8] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," *MHS'95. Proc. Sixth Int. Symp. Micro Mach. Hum. Sci.*, pp. 39–43, 1995.
- [9] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: artificial bee colony (ABC) algorithm and applications," *Artif. Intell. Rev.*, vol. 42, no. 1, pp. 21–57, Mar. 2012.
- [10] X. Yang and A. H. Gandomi, "Bat algorithm: a novel approach for global engineering optimization," *Eng. Comput.*, vol. 29, no. 5, pp. 464–483, 2012.
- [11] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *Control Syst. IEEE*, vol. 22, no. 3, pp. 52–67, 2002.
- [12] M. Dorigo, V. Maniezzo, and a Colorni, "The ant systems: optimization by a colony of cooperative agents," *IEEE Trans. Man, Mach. Cybern. B*, vol. 26, no. 1, 1996.
- [13] X. Yang, *Nature-Inspired Metaheuristic Algorithms Second Edition*. 2010.
- [14] H. Shareef, M. M. Islam, A. A. Ibrahim, and A. H. Mutlag, "A Nature Inspired Heuristic Optimization Algorithm Based on Lightning," *2015 3rd Int. Conf. Artif. Intell. Model. Simul.*, pp. 9–14, 2015.
- [15] Z. W. . Geem, J. H. . Kim, and G. V. . Loganathan, "A New Heuristic Optimization Algorithm: Harmony Search," *Simulation*, vol. 76, no. 2, pp. 60–68, 2001.
- [16] Z. W. Geem, X. S. Yang, and C. L. Tseng, "Harmony search and nature-inspired algorithms for engineering optimization," *J. Appl. Math.*, vol. 2013, pp. 2–4, 2013.
- [17] H. Shah&#45;Hosseini, "Principal Components Analysis by the Galaxy&#45;Based Search Algorithm&#58; a Novel Metaheuristic for Continuous Optimisation," *Int. J. Comput. Sci. Eng.*, vol. 6, no. 1/2, pp. 132–140, Jul. 2011.
- [18] K. Tamura and K. Yasuda, "Spiral dynamics inspired optimization," *J. Adv. Comput. Intell. Informatics*, vol. 15, no. 8, pp. 1116–1122, 2011.
- [19] M. R. Hashim and M. O. Tokhi, "Enhanced chaotic spiral dynamic algorithm with application to controller design," in *2016 IEEE International Conference on Power and Energy (PECon)*, 2016, pp. 752–756.
- [20] M. R. Hashim and M. . O. Tokhi, "Greedy spiral dynamic algorithm with application to controller design," in *2016 IEEE Conference on Systems, Process and Control (ICSPC)*, 2016, no. December, pp. 29–32.
- [21] M. O. T. M.R. Hashim, "CHAOTIC SPIRAL DYNAMICS OPTIMIZATION ALGORITHM," *Adv. Coop. Robot. Proc. 19th Int. Conf. Clawar 2016*, pp. 551–558, 2016.
- [22] karoboga, "basic idea on artificial bee colony algorithm," 2005.
- [23] G. Li, P. Niu, and X. Xiao, "Development and investigation of efficient artificial bee colony algorithm for numerical function optimization," *Appl. Soft Comput. J.*, vol. 12, no. 1, pp. 320–332, 2012.
- [24] H. Supriyono, M. O. Tokhi, and B. a M. Zain, "Control of a single-link flexible manipulator using improved bacterial foraging algorithm," *2010 IEEE Conf. Open Syst. (ICOS 2010)*, no. Icos, pp. 68–73, Dec. 2010.