Variable Population Size Artificial Bee Colony Algorithm Based Tuning of Optimal PID Controller

Nasr A. Elkhateeb*
Electrical Communications Department
Faculty of Engineering, Modern University for Technology and Information, Cairo, Egypt
*nasrantar@gmail.com

Ragia I. Badr
Electronics and Communications Department
Faculty of Engineering, Cairo University
Giza, Egypt
ragiabadr@yahoo.com

***Abstract*— Swarm intelligence has proven its superiority in solving problems that cannot be easily dealt with classical mathematical techniques. The foraging behavior of honey bees produces an intelligent social behavior and falls in the category of swarm intelligence. Artificial Bee Colony (ABC) algorithm is a foraging model established by Karaboga in 2005. But ABC algorithm has limitations due to its stochastic searching characteristic and complex computation that result in slow convergence to the global optimum solution. Variable Population Size ABC (VPS-ABC) algorithm based on global best re-initialization strategy is introduced to overcome the impact of the effect of initial population and improve the convergence rate of classical ABC. The VPS-ABC main idea is based on reducing the number of food sources gradually and moving the bees towards the global best food source in each re-initialization process. This work compares the performance of VPS-ABC algorithm with the classical ABC, PSO, and GA algorithms in tuning the proportional-integral-derivative (PID) controllers for the ball and hoop system which represents a system of complex industrial processes, known to be non-linear and time variant. Simulation results show that VPS-ABC algorithm is highly competitive, often outperforming PSO and GA algorithms and achieve convergence rate better than the classical ABC.**

Keywords*— Swarm intelligence; Evolutionary optimization; Artificial bee colon; PID controller.

I. INTRODUCTION

Evolutionary computation has become an important search and optimization technique for many researchers. These evolutionary computation algorithms (EA) are stochastic optimization methods that imitate biologic processes or natural phenomena [1]. Population-based parallel computation, collective learning process, self-adaptation, and robustness are some of the key features of evolutionary algorithms (EAs). The capability to find a global optimum, without being trapped in local optima, and the possibility to well face nonlinear and discontinuous problems with great numbers of variables, are some advantages of these techniques. Besides these methods does not need to compute any derivatives in order to optimize the objective function and this fact allows managing more complex fitness function. A significant improvement in performance can be achieved when the appropriate search distribution is applied. Population distribution can be improved by adding new particles in each generation to enhance the global solution. Recently, new optimization techniques have been developed to satisfy the optimization requirements (i.e. speed convergence, global solution, and reduction of computing effort). Current research activities are inspired by the behavior of bee life. Bee colony has been presented to guarantee the above requirements. It is based on the honey bee swarms and applied to solve optimization problems. Yang presented a virtual bee algorithm to solve the numerical optimization problems [2]. In 2005, Karaboga [3] has described a bee swarm algorithm called artificial bee colony (ABC) algorithm, which is different from the virtual bee algorithm. In ABC, the solution candidates are modeled as food sources and their corresponding objective functions as the quality (nectar amount) of the food source. In the first step, the artificial employed bees are randomly scattered in the search domain producing initial solutions. The initial solution represents the number of employed or onlooker bees which are considered equal until the end of the algorithm. However like most of the population based algorithms, ABC also has some non-appreciable drawbacks. One of the most important points is the impact of the initial population distribution which has affected the overall performance of artificial bee colony (i.e. the global solution, the convergence rate, and explorations for the global solution). As pointed by [4]-[5], the structure of ABC is such that it supports global exploration more in comparison to the exploitation. However, the performance of an algorithm depends on both exploration and exploitation phases for a required feasible solution. More recent, new modifications have been applied on the classical artificial bee colony such as one-position inheritance ABC [6], faster convergence ABC [7], and weighted sum ABC [8].

In this work, Variable Population Size Artificial Bee Colony (VPS-ABC) based on global best re-initialization strategy is proposed. This technique differs from the above techniques by using variable food sources for both employed and onlookers with global best guiding strategy instead of using fixed size food sources without guiding strategy. The main idea of VPS-ABC is to divide the time generations into periods. In each period the number of food sources starts with an initial number then decreases gradually. At the beginning of the next period randomly generated individuals around the global food source reinstall the population. In each period T new energetic food source individuals around the global solution start the race to search the neighborhood for the global best particles, resulting in more exploration of the search space and increasing the diversity of the population by incorporating dynamic initial weight in the basic equation of the food sources generation. These modifications enable VPS-ABC to escape from the local optima. Moreover, the overall performance is improved and the computation time is reduced. To show the fitness of the proposed algorithm; the new developed algorithm VPS-ABC is carried out to design optimal PID controller parameters for the ball and hoop. The ball and hoop is a system of complex industrial processes, known to be non-linear and time variant. Experimental studies show that VPS-ABC performs better in comparison to the classical ABC, PSO, GA search techniques and the obtained results have higher fitness as well as faster convergence rate.

The rest of the article is organized as follows. In section II, the basic concepts and the most common variants of ABC are explained. Section III describes VPS-ABC and the global best re-initialization strategy in details. An overview of evolutionary PID controllers, performance indices, problem under consideration are presented in section IV. Simulation results and the comparison between VPS-ABC algorithms and PSO as well as GA are provided in Section V. Finally, concluding remarks appear in section VI.

II. ARTIFICIAL BEE COLONY (ABC)

An intelligent behavior of honey bee colony which search new food sources around their hive was considered to compose bee colony algorithm. Although there are several models based on honeybees, this work is based on the artificial bee colony (ABC) algorithm which is derived from the behavior of honeybee. This model was initially proposed by Karaboga [3] and then lately formally introduced by Basturk and Karaboga [9]. ABC belongs to the group of algorithm which simulates foraging behavior. Tereshko developed a model of foraging behavior of a honeybee colony based on reaction [10]. The foraging model and searching behaviors will be explained in the following.

A. Foraging Model

This model that leads to the emergence of collective intelligence of honeybee swarms consists of three essential components: food sources, employed foragers, and unemployed foragers. It defines two leading modes of the honeybee colony behavior: recruitment to a food source and abandonment of a source. In ABC algorithm, the colony of artificial bees consists of three groups of bees called employed bees, onlookers, and scouts. While half of the colony consists of the employed artificial bees, the other half includes the onlookers. There is only one employed bee for every food source. That is, the number of employed bees is equal to the number of food sources around the hive. ABC algorithm has been applied successfully to a large number of various optimization problems [11]. In each cycle, food sources are mutated with their neighbors to produce new solutions and then evaluated based on the fitness function. A food source that does not produce improvement in solutions is assumed abandoned source.

B. Basics of ABC

In ABC algorithm, the colony of artificial bees consists of three groups of bees called employed bees, onlookers, and scouts. While a half of the colony consists of the employed artificial bees, the other half includes the onlookers. There is only one employed bee for every food source. That is, the number of employed bees is equal to the number of food sources around the hive. In each cycle, food sources are mutated with their neighbors to produce new solutions and then evaluated based on the fitness function. A food source that does not produce improvement in solutions is assumed abandoned source. In the ABC algorithm, each cycle of the search consists of three phases:

1) Employed Bee Phase:

In this phase, ABC send employed bees onto the food sources and then measures the source quality (i.e. quantity, richness, and closeness….etc.). Employed bees carry this information to the hive and then throughout the dancing area (area of information exchange) share it with the onlookers. Employed bee memorizes her food source and then either continues in the same food source or selects a new one. An artificial bee produces a new solution with formula [9].

$$v_{i,j} = x_{i,j} + \Phi_{ij}(x_{i,j} - x_{k,j}) \tag{1}$$

$v_{i,j}$ is a new solution that comes from food source $x_{i,j}$ and its neighbor $x_{k,j}$, where j and k are two random indices, Φ is randomly produced number in the range [-1,1].

2) Onlooker Bee Phase

Onlookers decide and select the source food depending on the nectar information. The probability of selecting certain source food increases as the information received from the dancing area mean large amount of nectar exist and high quality Onlookers choose a food source with a probability calculated using different schemes. In this work an artificial onlooker bee chooses a food source with probability p_i expressed as:

$$p_i = a * \frac{fit_i}{\sum_i^k fit} + b \tag{2}$$

where a and b are two arbitrary numbers in range [0, 1], fit_i is the fitness value of solution i, and k is the number of employed bees. Basturk et al [9] has expressed the fitness function as

$$fit_i = \begin{Bmatrix} \frac{1}{1+F_i} & if\ F_i \geq 0 \\ 1 + abs(F_i) & if\ F_i < 0 \end{Bmatrix} \tag{3}$$

F_i is the objective function to be optimized

3) Scout Phase:

When an employed bee decides to leave her food source she becomes a scout. In ABC, if a food source position cannot be improved further through a predetermined number of cycles, then that food source is discarded. The value of predetermined number of cycles is an important control parameter of the ABC algorithm, which is called *limit* for abandonment. Scout phase plays a good role in adding new comers to the solution space (i.e. by generating new random solutions instead of abandonment food sources). Moreover, the number of scouts is limited by the colony size and dimension of the problem.

III. Variable Population Size Artificial Bee Colony (VPS-ABC)

In ABC, the system is initialized with a population of random solutions and searches for the optima by updating the population in the succeeding generations. However, reaching the global solution and faster convergence rate are the two basic advantages in any optimization algorithm; so that improving the convergence rate and the computational effort is the motivation to this work:

A. Variable Food Source

Varying the population size in the proposed algorithm depends on reducing, periodically, the number of the food sources available for both employed bees and onlooker bees phase for a population period time T. The period time T is suggested to energize the available food sources in each period by using re-initialization strategy. This modification is promising, because the most evolutionary algorithms (i.e. ABC, GA, PSO...) suffer from the impact of the initial population on the new produced solution, which reflect the difficulty to reach the global solution and settled to a local one. Moreover, VPS-ABC algorithm starts with a bigger food sources at the beginning which provides a better initial signal for the ABC evolution process and guides the algorithm to the region of global solution; whereas, a smaller population size is adequate at the end of the run, where the ABC converges to the optimum. The proposed food source number profile is suggested to decrease linearly with the generation time as depicted in the following figure.

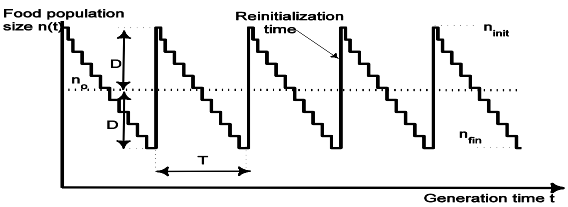

Fig. 1 Variable food sources size

In **Fig.1**, the number of food sources starts with maximum number n_{init} and linearly decreases to n_{fin}. The average number of population size is n_o where $n_{init} \leq n_o \leq n_{fin}$ for a population period time T. This process is then repeated along the generation time. The suggested function that produces the food source number is introduced by [14] to generate the population size in (Saw-Tooth GA) scheme as follow:

$$n(t) = int(n_o + D - \frac{2D}{T-1}\left[t - T * int\left(\frac{t-1}{T} - 1\right)\right]) \quad (4)$$

where T is the population period time, t the generation time, D the maximum deviation around the average population size (i.e. $n_o - D \leq n(t) \leq n_o + D$), int refers to $integer$. If $D = 0$, VPS-ABC returns to constant food source artificial bee colony algorithm.

B. Global Best Re-initialization Strategy

The effect of population re-initialization is in a sense similar to the mutation operator in Genetic Algorithms (GAs). Such operator introduces random changes in the population to increase diversity in the suggested population space and achieve better exploration of the search space. This effect is favorable when the GA population prematurely converges to a certain point or local optimum and further improvement is not likely. Therefore, population re-initialization represents a good strategy especially for the case of multimodal problems [12].

In this work, the global best re-initialization strategy is introduced to improve the overall performance of the classical ABC. This strategy is based on guiding the bees around the most recent best food source to reach the best food source; this variant is inspired from the particle swarm optimization (PSO) algorithm. The global best re-initialization strategy is executed at the end of the population period T, the re-initialization strategy is applied by replacing the low fitness probability food sources with new food sources around the global solution equation (5). This strategy helps the (ABC) algorithm to redistribute the expected solutions around the region of the global minimum in each population period time T which has a good impact on the convergence rate and the overall performance of (ABC). The re-initialization equation is proposed to be:

$$v_{ij} = x_{ij} + \Phi_{ij}(x_{global} - x_{ij}) \quad (5)$$

v_{ij} is a new solution comes from food source x_{ij}around $x_{global}$the global best food source found position among all food sources, Φ is a randomly produced number in range [0,1].

C. Dynamic Inertia Weight

Moreover, the performance of the bee colony can be improved by controlling the impact of the initial population on the new produced food sources. It was shown in [4] that, introducing dynamic inertia weight parameter in the basic (ABC) equation (1) can play a good role in controlling the impact of the previous foods on the new expected one as follow:

$$v_{ij} = \omega_i x_{ij} + \Phi_{ij}\left(x_{global} - x_{ij}\right) \quad (6)$$

The inertia weight ω_i is employed to manipulate the impact of the previous history of velocities on the current velocity. Therefore, ω_i resolves the tradeoffs between the global (wide ranging) and local (nearby) exploration ability of the swarm [13]. A large inertia weight encourages global exploration (moving to previously not encountered areas of the search space), while a small one promotes local exploration, i.e., fine-tuning the current search area. Suitable value for ω_i provides the desired balance between the global and local exploration ability of the swarm and, consequently, improves the effectiveness of the algorithm [13]. A linearly-increasing time-dependent inertia weight is implemented according to the following updated equation:

$$\omega_i = \left(\omega_{init} - \omega_{fin}\right)\left(\frac{N-i}{N}\right) + \omega_{fin} \quad (7)$$

where ω_{init} is the initial inertia weight, ω_{fin} is the final inertia weight, N is the maximum iteration value and i is the variable iteration index. Note here that the inertia weight ω plays an important role in the convergence of the ABC algorithm to the global optimal solution and hence has an influence on the time taken for a simulation run. The framework of the variable population size artificial bee colony is given as follows:

Algorithm Variable Population Size **VPS-ABC** algorithm

Input: Number of Employed Bees, Onlooker Bees**,** iterations Population period, Optimization parameters, Optimization function, and Upper and lower parameters limits

Output: Optimum parameters, global solution

1. Generate random initial population
2. Apply Fitness evaluation of all individuals
3. Store global parameters
4. **Loop (iterations)**
5. Check optimization criteria
6. Update food sources $n(t)$ **(4)**, and $\omega_i(7)$
7. Employed Bee Phase **(5)** ➔ for all employed bees
   Fitness evaluation and store local parameters
8. Calculate probability **(2)**
9. Onlookers Bee Phase **(5)** ➔ for all onlookers bees
   Fitness evaluation and store local parameters
10. Scout Phase
11. Store global parameters
12. **If** the population period T is ended
    Replacing the low fitness probability food sources with new food sources around the global solution **(6)**
    Else return to **step 5**
13. **End**
14. **End Loop**

Fig. 2 Variable population size artificial bee colony framework

IV. Experimental Case Study

In this section the proposed Variable Population Size Artificial Bee Colony (VPS-ABC) optimization algorithm is tested with tuning the proportional-integral-derivative PID controllers for the ball and hoop system then the obtained results are compared with PSO and GA results:

A. *Evolutionary PID Controller*

Popularity of PID controllers makes it widely used in processes and motion control system in industry because it is simple and robust in practical applications. The most critical step in application of PID controller is parameters' tuning. There are many tuning rules for PID controller parameters based on the EAs such as Genetic Algorithms (GA) [14] and Particle Swarm Optimization (PSO) [15]. The parameter settings of a PID controller for optimal control of a plant depend on the plant's behavior. A general body of a PID control system is shown in (8), where it can be seen that in a PID controller, the error signal $e(t)$ is used to generate the proportional, integral, and derivative actions, with the resulting signals weighted and summed to form the control signal $u(t)$ applied to the plant model. A mathematical description of the PID controller is:

$$u(t) = K_p e(t) + K_i \int e(t)\ dt + K_d \frac{de(t)}{dt} \qquad (8)$$

The goal is to tune proper coefficients K_p, K_i and K_d so that the output has some desired characteristics. Usually in time domain, these characteristics are given in terms of maximum overshoot, rise time, settling time and steady state error.

B. *Performance Indices*

The most common performance indices have been suggested to measure the performance of the optimum PID controller are the integral absolute error (IAE), the integral square error (ISE), the integral time absolute error (ITAE), and the integral time square error (ITSE). These indices are normally calculated based on the step response.

1-*Integral Absolute Error (IAE)*

$$I_{IAE} = \int_0^\infty |r(t) - y(t)|\ dt = \int_0^\infty |e(t)|\ dt$$

2-*Integral Time Absolute Error (ITAE):* $I_{ITAE} = \int_0^\infty |te(t)|\ dt$

3-*Integral Square Error (ISE):* $I_{ISE} = \int_0^\infty e(t)^2\ dt$

4-*Integral Time Square Error (ITSE):* $I_{ITSE} = \int_0^\infty te(t)^2\ dt$

The transient response parameters such as, maximum overshoot M_p, settling time T_s, rise time T_r are normally considered significant where the benefits of faster systems necessitates minimum possible values for them [16]. In this work, two performance indices are suggested to evaluate the performance of the optimum PID controller:

$$IAE\ Cost\ Function = \int_0^\infty e(t)\ dt + 0.5\,T_s + 0.1\,M_p \qquad (9)$$

$$ISE\ Cost\ Function = \int_0^\infty e(t)^2\ dt + 0.5\,T_s + 0.1\,M_p \qquad (10)$$

The tuning of VPS-ABC based PID controller's results will be compared to those obtained from the classical ABC, GA, and PSO. The following figure describes the auto PID-tuning diagram.

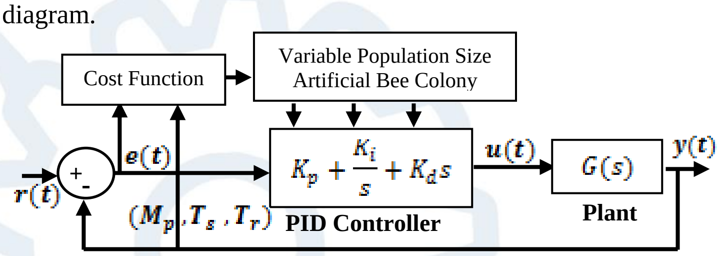

Fig. 3 Schematic diagram of VPS-ABC based optimizer for PID controller

C. Ball and Hoop system

The Ball and Hoop system is an analogue of the liquid slop/slosh problem that occurs when fluids are transported, either as a cargo or a fuel [17] as depicted in **Fig.4**. The movement of the ball in the inner periphery of the hoop reproduces the oscillations of fluids in tanks and tankers during transportation. The ball and hoop is also a good system for demonstrating some basic concepts in control and systems dynamics, such as system zeros and non-minimum phase systems. As illustrated in **Fig.4**, the angle θ is the hoop angular position and the position of the ball is given by:

1) y is the position of the ball on the hoop periphery with respect to a datum point.

2) ψ is the slosh angle which measures the deviation of the ball from its rest position.

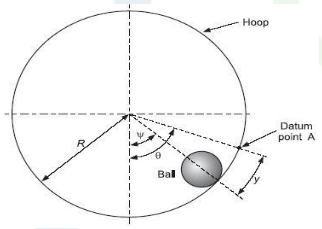

Fig. 4 The ball and hoop system model

A fourth order system was selected for the Ball and Hoop system with the following transfer function [17]

$$G(s) = \frac{1}{s^4 + 6s^3 + 11s^2 + 6s} \qquad (11)$$

The ball and hoop apparatus is difficult to control optimally using a PID controller because the system parameters are constantly. The PID controller will be tuned offline using VSP-ABC, GA, and PSO algorithms, where the proposed objective functions (9-10) were used to evaluate the performance of the problem as shown in **Fig.4**.

V. Simulation and Results

Numerical results of computer simulations have been performed to assess the capabilities of the proposed algorithm.

A. Settings

For VPS-ABC, the number of colony size is 30 (employed bees + onlooker bees), number of scouts is equal to one, food sources are equal to half colony size 15. The maximum deviation $D = 5$ (i.e.$5 \leq n(t) \leq 15$), the population period $T = 20$. ABC algorithm limit equal 10 (food source which could not be improved through "limit" trials is abandoned by its employed bee) [11]. PID parameters bounds are [0, 50]. Constants of probability (2) are chosen to be $a = 0.75$, and $b = 0.25$. The dynamic range of the inertia weight [0.6, 1] where$\omega_{init} = 0.6$, and $\omega_{fin} = 1$. PSO, all swarm particles start at a random position in the range [0, 50] for each dimension and dynamic inertia weight. The velocity of each particle is randomized to a small value to provide initial random impetus to the swarm and the velocities bounds $V_{max} = 25$, and $V_{min} = -25$, and the swarm size was limited to 20 particles. For GA, the population size of 20, the search range [0, 50], mutation rate is equal to 20% and with other default parameters. Finally, the maximum number of cycles is equal to $N = 100$. Any set of PID parameters that gave unstable system performance had their weighted error value set really high so that they would not be chosen for selection in the ABC learning process.

B. Results

Simulation results comparing between the four techniques (GA, PSO, Classical ABC, and VPS-ABC) are shown , where the IAE cost function (9) have been used in **Fig.5** while the ISE cost function (10) have been used in **Fig.7**. In each case the step response for the ball and hoop system represented by the transfer function (11) is shown as well as the convergence history during the tuning process. To show the performance of VPS-ABC, the optimal PID parameter tuning history is given under optimization of (9) and (10) respectively.

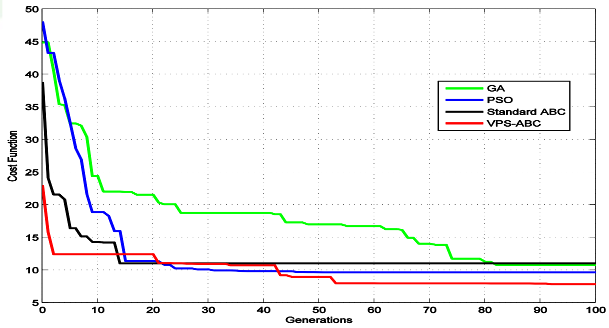

Fig. 5 Convergence history of IAE during tuning process G(s)

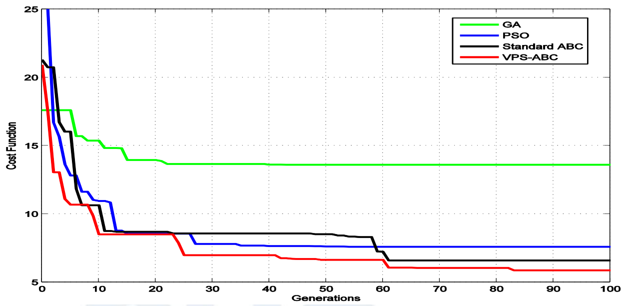

Fig. 6 Convergence history of ISE during tuning process G(s)

Moreover, table's I-II gives a comparative study between the different techniques with respect to the optimal PID parameters, rise time, maximum peak, settling time, and cost function. Simulation results demonstrate the superiority of VPS-ABC based on global best re-initialization strategy comparing to the other algorithms. In terms of overshot (peak value) has lower overshot by 8.324% than GA and 4.064% than classical ABC. For settling time T_s, VPS-ABC has a lower settling time by 50% than GA, PSO, and classical ABC. VPS-ABC has IAE cost function less than classical ABC by 40%, PSO by 22.9%, and GA by 37% as shown in **Fig.7**.

TABLE.I
TRANSIENT CHARACTERISTICS FOR IAE COST FUNCTION

Index	GA	PSO	Classical (ABC)	VPS-ABC
k_p	5.2105	3.9763	4.7165	3.7276
k_i	0.06789	0	0	0.0283
k_d	7.8852	7.3693	8.9605	5.2039
(secT_r)	1.3003	1.5552	1.2693	1.9566
T_s (sec)	5.5156	6.8429	6.0358	1.9566
T_p(sec)	2.6699	25.954	2.4699	14.97
M_p %	9.4652	0	5.2042	1.1364
IAE Cost Function	10.808	9.668	11.0342	7.8636

For ISE cost function, the results are as follow: the improvement is about 12.434% over classical ABC, 29.4% over PSO, 100% over GA. Moreover, the convergence rate of the VPS-ABC is very fast in comparing to classical ABC, PSO, and GA as shown in **Fig.8**.

TABLE.II
TRANSIENT CHARACTERISTICS FOR ISE COST FUNCTION

Index	GA	PSO	Classical (ABC)	VPS-ABC
k_p	9.0004	3.9716	4.5179	4.0359
k_i	2.7209	0	0.054771	0.042291
k_d	13.357	7.3711	6.7298	6.0582
T_r (sec)	0.82273	1.5554	1.5124	1.7191
T_s (sec)	7.722	6.8517	3.8466	2.7179
T_p(sec)	2.1054	26.002	3.2093	3.5259
M_p %	41.485	0	4.8607	1.4222
ISE Cost Function	13.606	7.5971	6.6011	5.58712

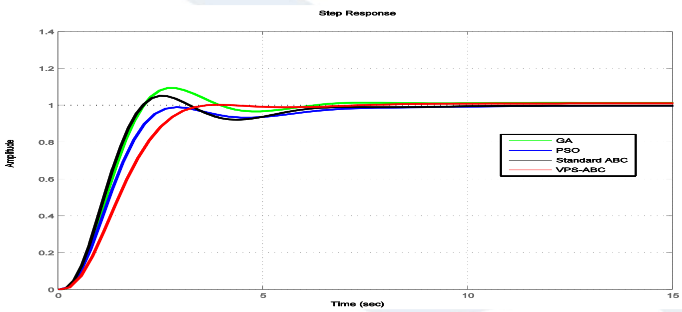

Fig. 7 Step response of G(s) having optimal PID under optimization of IAE

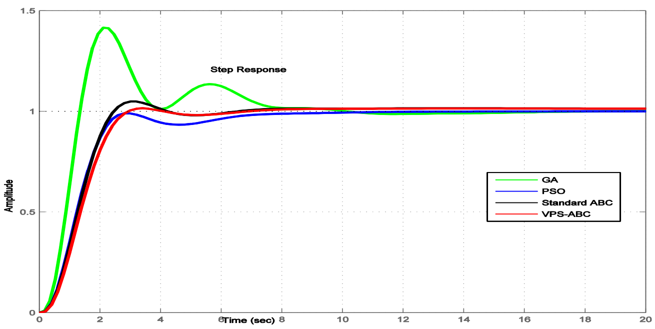

Fig. 8 Step response G(s) having optimal PID under optimization of ISE

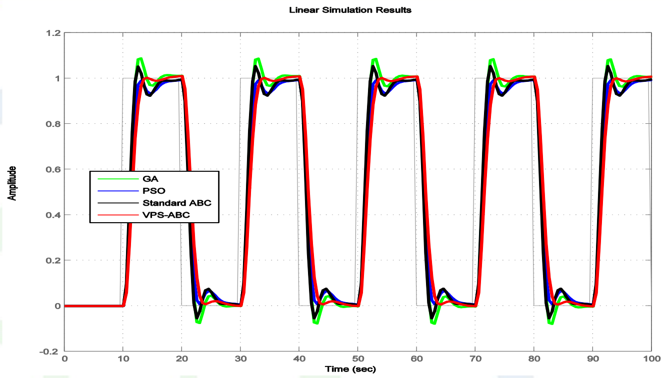

Fig. 9 Ball and hoop response due to square wave input under optimization of IAE

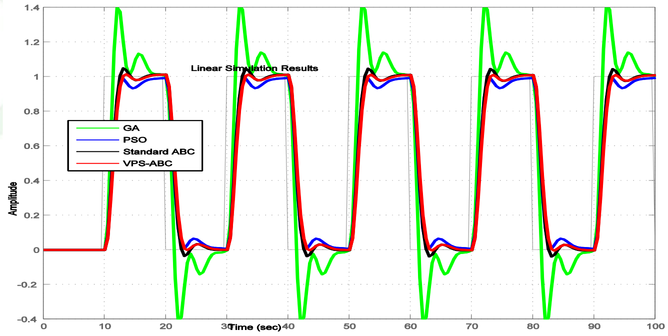

Fig. 10 Ball and hoop response due to square wave input under optimization of IAE

Moreover, the response of the ball and hoop system having optimal PID controller due to square input is depicted in **Fig.9** and **Fig.10**.

VI. CONCLUSION

In this paper a comparative study between GA, PSO, and a novel variable population size artificial bee colony (VPS-ABC) algorithm has been presented. The VPS-ABC strategy has been proposed to overcome the impact of the initial population, reduce the computational effort, and help the algorithm to explore the global solution quickly. The performance of VPS-ABC algorithm is compared to the classical ABC, GA, and PSO models in the tuning of PID controller in the ball and hoop system which represents a system of complex industrial processes. For the system under consideration, the simulation results with variable size ABC technique prove to be more effective than with GAs and PSO. It has been shown that the VPS-ABC algorithm is faster in convergence and the obtained solutions are of higher fitness than the classical ABC, GA, and PSO. Also, the time response characteristics of the processes demonstrate the superiority of VPS-ABC based on global best re-initialization strategy comparing to the other algorithms.

REFERENCE

[1]. T. Back," Evolutionary algorithms in theory and practice," Oxford University Press, London, UK, 1996.

[2]. X.S. Yang," Engineering optimizations via Nature-Inspired virtual bee algorithms," Lecture Notes in Computer Science, Springer-Verlag GmbH, vol. 3562, pp. 317- 323,2005.

[3]. D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Lecture Notes in Computer Science Technical Report-TR06,Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

[4]. N. Elkhateeb and R. Badr, "Employing artificial bee colony with dynamic inertia weight for optimal tuning of PID controller,"5th International Conference on Modeling, Identification and Control, Cairo, Egypt, 2013

[5]. G. Zhu, S. Kwong, "Gbest-guided artificial bee colony algorithm for numerical function optimization," Applied Mathematics and Computation, 217(7), pp.3166- 3173, 2010.

[6]. Xin Zhanga, Kwong Fai Fongb, and Shiu Yin Yuena , "A novel artificial bee colony algorithm for HVAC optimization problems",HVAC&R Research, Taylor&Francis,19(6), pp.715-731,2013.

[7]. Xiangyu Yu Jiaxin Zhang, Jiaru Fan and Tao Zhang," A Faster Convergence Artificial Bee Colony Algorithm in Sensor Deployment for Wireless Sensor Networks", International Journal of Distributed Sensor Networks, Article ID 497264, 9 pages, 2013.

[8]. Naidu,K.,Mokhlis,Bakar,A.H.A.," Multi-objective optimization using weighted sum Artificial Bee Colony algorithm for Load Frequency Control", Electrical Power and Energy Systems, vol.55,pp.657- 667,2014.

[9]. B. Bastruk and D. Karaboga," An artificial bee colony (ABC) algorithm for numeric function optimization," Applied Soft Computing, 8(1), pp. 687-697, 2008.

[10]. V. Tereshko and A. Loengarov, "Collective decision-making in honey bee foraging dynamics," Comput. Inf. Syst. J., 9(3) pp. 1-7, 2005.

[11]. T.K. Sharma, M. Pant, "Enhancing the food locations in an artificial bee colony algorithm," IEEE Swarm Intelligence Symposium (SIS), Paris, France, pp. 119-123, 2011.

[12]. Vlasis K. Koumousis and Christos P. Katsaras,"A saw-tooth genetic algorithm combining the effects of variable population size and reinitialization to enhance performance," IEEE Transactions on Evolutionary Computation,10(1), pp. 19-28,2006.

[13]. N. Elkhateeb and R. Badr, "Dynamic Inertia Weight Artificial Bee Colony versus GA and PSO for Optimal Tuning of PID Controller," Int. J. of Modeling, Identification and Control, Vol.22, No.4, pp.307-317, 2014.

[14]. S.N. Sivanandam, S.N. Deepa,"Introduction to Genetic Algorithms," Springer-Verlag, Berlin, Heidelberg, 2008.

[15]. Chang W.D.,"PID control for chaotic synchronization using particle swarm optimization," Chaos Solitons & Fractals, 39(2), pp. 910-917, 2009.

[16]. M. Abachizadeh, M. Yazdi,and A. Yousefi-Koma,"Optimal tuning of PID controllers using Artificial Bee Colony algorithm," IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Montréal, Canada, 2010.

[17]. Griffin, I.," On-line PID controller tuning using genetic algorithms," M.Sc. thesis School of Electronic Engineering, Dublin City University, 2003.