

Research Article

Application of Combinatorial Interaction Design for DC Servomotor PID Controller Tuning

Mouayad A. Sahib, Bestoun S. Ahmed, and Moayad Y. Potrus

Software Engineering Department, Salahaddin University-Hawler (SUH), Kirkuk Road, Erbil 44002, Kurdistan, Iraq

Correspondence should be addressed to Bestoun S. Ahmed; bestoon82@gmail.com

Received 22 January 2014; Revised 23 March 2014; Accepted 25 March 2014; Published 16 April 2014

Academic Editor: Petko Petkov

Copyright © 2014 Mouayad A. Sahib et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Combinatorial optimization has been used in different research areas. It has been employed successfully in software testing fields to construct minimum set of combinations (i.e., in terms of size) which in turn represents the minimum number of test cases. It was also found to be a successful approach that can be applied to solve other similar problems in different fields of research. In line with this approach, this paper presents a new application of the combinatorial optimization in the design of PID controller for DC servomotor. The design of PID controller involves the determination of three parameters. To find optimal initial PID parameters, different tuning methods have been proposed and designed in the literature. The combinatorial design is concerned with the arrangement of finite set of elements into combinatorial set that satisfies some given constraints. Consequently, the proposed method takes the interaction of the input parameters as a constraint for constructing this combinatorial set. The generated sets are then used in the proposed tuning method. The method proved its effectiveness within a set of experiments in a simulated environment.

1. Introduction

Direct current (DC) motors have been used intensively in various industrial applications. The significance of DC motors is related to their efficient characteristics such as preciseness, fast adaptation, smooth operation, and high torque capabilities. From the control point of view, DC motors exhibit linear speed-torque characteristics; therefore, proficient control aspects can be achieved [1]. During the recent decades, various controller structures have been proposed for controlling the DC motor. The most common controller used is the proportional plus integral plus derivative (PID) due to its simple structure and robust performance [2]. The design of the PID controller involves the determination of three parameters which are the proportional, integral, and derivative gains. Over the years, various tuning methods have been proposed to determine the PID gains. The first classical tuning rule method was proposed by Ziegler and Nichols [3] and also it is proposed by Cohen and Coon [4]. The advantage of these classical tuning methods is that the model of the system is not required to be known. However, the controller designed with these experimental methods can

have an acceptable but not optimum system response. Therefore, recently, many artificial intelligence (AI) techniques have been employed to determine the optimal parameters and hence improve the controller performances. Such AI techniques include differential evolution (DE) algorithm [5], multiobjective optimization [6], evolutionary algorithm [7], simulated annealing (SA) [8], artificial bee colony (ABC) [9], fuzzy systems [10], genetic algorithm (GA) [11], and many optimizing liaisons (MOL) [12].

In this paper, combinatorial design is used as a new approach to determine the optimal PID parameters. In combinatorial design, the finite set of elements is arranged into patterns (subsets, words, and arrays) according to specified rules [13]. As a part of this theory, the arrangement could be represented in the covering array (CA) form, which represents all the interaction combinations in one array. Thus, combination number can be reduced dramatically, and at the same time all of them are covered by the optimized set which is represented in the CA.

CAs have been used widely in different applications in the literature. They have been used mainly in software testing applications (e.g., [14–16]). They have also been used in other

applications such as hardware testing [17], gene expression regulation [18], advance material testing [19], performance evaluation of communication systems [20], and optimization of dynamic voltage scaling (DVS) in high performance processors [21]. However, CAs have not been used yet for PID controller design; thereby, it is proposed here to determine the optimal PID parameters.

The rest of this paper is organized as follows. Section 2 gives an overview of the combinatorial design and its notations. Section 3 describes the PID controller design. Section 4 gives the details of the combinatorial set construction and the procedure of its application on the system. Section 5 presents and discusses the evaluation results of the proposed approach followed by concluding remarks in Section 6.

2. Combinatorial Optimization Background

Combinatorial design is used to effectively search for the best solution among a finite set of feasible solutions based on the interactions. It is essential in this method to cover all of the combinations at least once. CA has been introduced as a mathematical object to represent all those combinations. A $CA\lambda(N; t, k, \nu)$ represents an $N \times k$ array with ν values such that every $N \times t$ subarray contains all ordered subsets from ν values of size t at least λ times [22], where k is the number of components (parameters). When the target is optimal combination-set, it is essential that all t -combinations occur at least once. In this case, the value of $\lambda = 1$ and the notation becomes $CA(N; t, k, \nu)$ [23]. As the target is the optimal set, the size of N , which is the size of the combination-set, has to be as minimum as possible.

T -way testing strategies have been used widely to construct CA and its variations such as mixed covering array (MCA) or variable strength covering array (VSCA) [24] for the purpose of software interaction testing. There are different mechanisms in the literature used with t -way strategies for generating the combinations. These methods are lying under the field of search based software engineering (SBSE) [25].

From the literature, there are two methods for generating the combinations in general: (1) algebraically, which is based on the generation of the combinations mathematically; (2) computationally, which is based on the generation of the combinations using computational and iterative methods. In practice, the second method is mostly used because of the flexibility since the former method is more restrictive for the combinations.

There are different tools and strategies in the literature for generating the combinations of different inputs. Each tool uses a special algorithm for generation. Among those implemented algorithms, the use of AI theories with the random generation proves the generation of optimal combination sizes. This approach has been used for combinatorial optimization in different researches (e.g., [14, 26]).

3. PID Controller Design

The objective of designing a controller for a DC servomotor is to make the system respond efficiently to the new desired

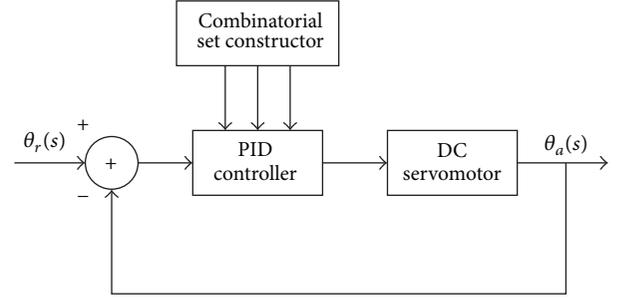


FIGURE 1: DC servomotor with PID controller tuned by combinatorial set constructor.

angular position. This can be achieved by comparing the output angular position with the desired (reference) position to calculate an error (actuating) signal. The controller manipulates the error signal and generates an appropriate DC voltage that will cause the motor to respond to the new position. Proportional-integral-derivative (PID) controller has been widely used for such application because of its simple structure and robust performance. Figure 1 shows a block diagram representation of a DC servomotor with PID controller.

The design of PID controller involves the computation of three parameters: proportional (K_p), integral (K_i), and derivative (K_d) constants. The general transfer function of the PID controller is given by

$$C(s) = K_p + \frac{K_i}{s} + sK_d. \quad (1)$$

The conventional method of determining the PID parameters is a heuristic method called the Ziegler-Nichols method [27]. This method is based on calculating the system critical gain at which the output of the control loop oscillates along with the corresponding period of oscillation. The critical gain and period of oscillation are used to set the P, I, and D gains. Later, methods based on several optimization techniques have been introduced, such as genetic algorithm [28], neural network [29], fuzzy based approach [30], and particle swarm optimization techniques [31]. Most systems arising in practice have time varying parameters which will affect the performance of the designed PID controller. In such cases a supervisory system can be used such that, when the performance of the controller drops below a prescribed level of acceptable performance index, the proposed optimization procedure must be reengaged. Therefore, the tuning process has to be sufficiently satisfactory in terms of convergence speed and this can be achieved when reducing the optimization searching space. This work proposes a new method for determining the PID controller parameters using covering array arrangement based on combinatorial design approach. As shown in the block diagram of Figure 1, the “combinatorial set constructor” feeds the PID controller with sets of parameters instead of the conventional techniques. This will reduce the amount of the search space which in turn reduces the required time for the system to reach the optimal value. The following section illustrates how this

combinatorial set is generated and how it has been applied to the model.

4. Applying the Generated Combinatorial Set

Figure 3 summarizes the process of constructing the combinatorial set and its application on the system. For the purpose of constructing the combinatorial set, we have used our previously developed strategy named PSTG [14, 16]. The strategy uses the particle swarm optimization (PSO) for constructing and optimizing the combinatorial set.

First of all, the strategy takes the factors involved in PID tuning process and then generates all the combinations of the input parameters (i.e., exhaustive set). During this process, an individual random set is generated initially and involved into an iterative path. The combinations are optimized during the iterative path of selection using the PSO strategy. Each row in the random generated set is compared with the complete set of combinations iteratively to know how many combinations it covers. Then, using the PSO mechanism for optimization, the strategy updates the random set of combinations according to an update rule. The strategy continues to iterate until it finds one combination in the random set that could cover more than one in the original set, which optimizes the original set to a smaller set of combinations. The strategy collects all the optimized combinations in one set and it stops until all the combinations are covered in the exhaustive set of combinations. Then, it applies the generated set to the PID controller (see Figure 1). Hence, this set will be the source of the PID parameters instead of using other sources which are coming by conventional techniques. As a result, the search space will be minimized and the system will reach the optimal value in a satisfactory time.

5. Simulation Results and Discussion

In this section, the performance of the proposed PID tuning method is evaluated in MATLAB and applied to control a DC servomotor. The DC motor system can be considered as a SISO system in which the input is the DC source voltage $E_a(s)$ and the output is the shaft angular position $\theta_a(s)$. The dynamical behavior of the DC motor as a relationship between the applied DC voltage source and the angular position of the shaft can be described by the following transfer function [32]:

$$G(s) = \frac{\theta_a(s)}{E_a(s)} = \frac{K_t}{J_m L_a s^3 + (R_a J_m + D_m L_a) s^2 + (D_m R_a + K_t K_b) s} \quad (2)$$

The parameters in (2) are defined in Table 1 [32, 33].

TABLE 1: DC servomotor parameters.

Parameter	Value	Unit
Torque constant (K_t)	0.01	N*m/amp
Equivalent inertia (J_m)	0.01	Kg*m ²
Armature inductance (L_a)	0.5	H
Armature resistance (R_a)	1	Ohm
Viscose damping (D_m)	0.1	N*m*sec/rad
Back emf constant (K_b)	0.01	Volt*sec/rad

With the parameters listed in Table 1, the DC servomotor transfer function will be

$$G(s) = \frac{\theta_a(s)}{E_a(s)} = \frac{0.01}{s(0.005s^2 + 0.06 + 0.1001)} \quad (3)$$

The unit step response of the DC motor without PID controller is shown in Figure 3.

It can be observed from Figure 3 that the DC motor system is overdamped with a settling time equal to 37.33 sec at which the response has settled to 98% of the steady state value. In the design of PID controller with combinatorial approach, K_p values are selected in the range of 45.2 to 55, K_i in the range of 0.01 to 0.5, and K_d in the range of 13.5 to 38 with step sizes equal to 0.2, 0.01, and 0.5, respectively. Accordingly, the table of all combinations is produced to have a total of $50^3 = 125000$ possible combinations. It is observed that, within this range, tuning is done effectively, as can be shown later. Each PID parameter combination is evaluated using a fitness function given by

$$J = \alpha_1 \frac{|t_r - t_r^*|}{t_r^*} + \alpha_2 \frac{|t_s - t_s^*|}{t_s^*} + \alpha_3 \frac{|t_p - t_p^*|}{t_p^*} + \alpha_4 \frac{|M_p - M_p^*|}{M_p^*} + \alpha_5 E_{ss} \quad (4)$$

where t_r , t_s , t_p , and M_p are the rise time, settling time, peak time, and maximum overshoot variables which are used to specify the system's response, respectively. The same variables assigned with asterisk (*) are used in (4) to denote the desired response specifications. E_{ss} is the steady state error. ($\alpha_1 - \alpha_5$) are weighting factors used to adjust the importance of the response specification variables and it is up to the designer to choose the values of these weights. In the simulation, t_r^* , t_s^* , t_p^* , and M_p^* are selected to be 0.475, 0.879, 0.64, and 0.0455, respectively. The weighting factors ($\alpha_1 - \alpha_5$) are all chosen to be 1 and the unit step response is calculated between 0 and 8 sec with a sampling time of 0.01 sec. Thus, (4) becomes

$$J = \frac{|t_r - 0.475|}{0.475} + \frac{|t_s - 0.879|}{0.879} + \frac{|t_p - 0.64|}{0.64} + \frac{|M_p - 0.0455|}{0.0455} + E_{ss} \quad (5)$$

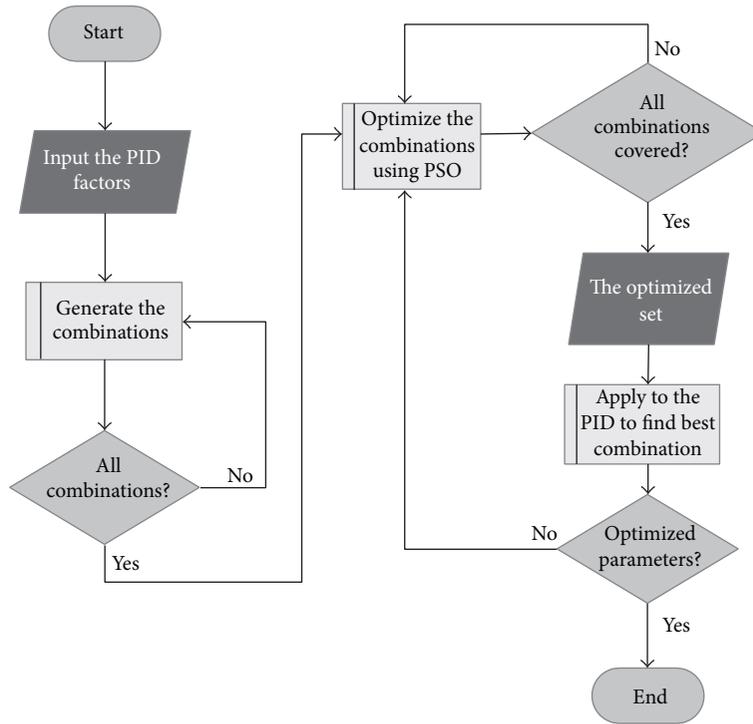


FIGURE 2: Process of combinatorial set construction.

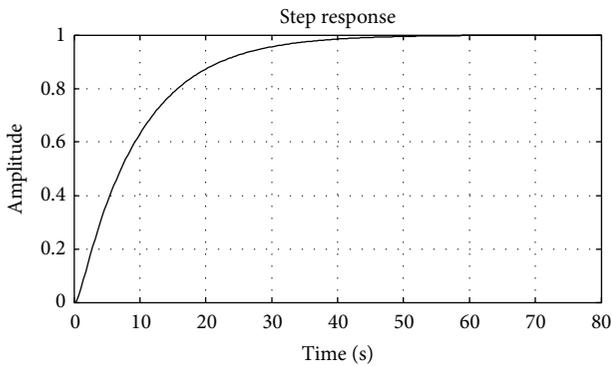


FIGURE 3: Step response of the DC motor without PID controller.

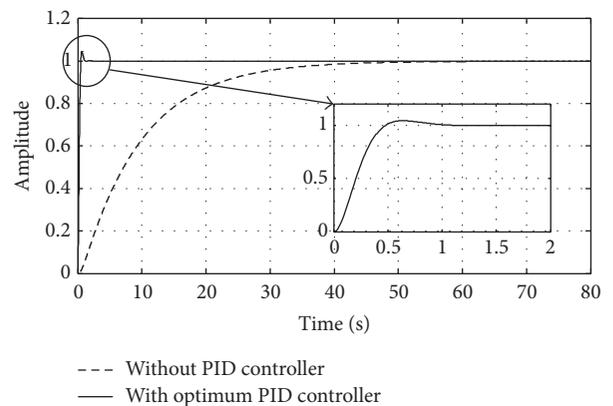


FIGURE 4: Step response of the DC motor with optimum PID controller parameters.

For this objective function, the value of J has a unique minimum value equal to zero only when t_r , t_s , t_p , and M_p are equal to 0.475, 0.879, 0.64, and 0.0455, respectively. Therefore, the optimization problem is to design a PID controller which has optimum gains such that when this PID is used to control the DC motor, the overall system step response will exhibit t_r , t_s , t_p , and M_p values equal to 0.475, 0.879, 0.64, and 0.0455, respectively.

Among all combinations, $K_p = 50$, $K_i = 0.18$, and $K_d = 24.5$ are found to be the global optimum PID parameters which can satisfy the desired response. The step response of the DC motor controlled by the PID controller with optimum parameters is shown in Figure 4.

From the unit step response shown in Figure 4, it can be observed that $t_r = 0.4755$, $t_s = 0.8784$, $t_p = 0.6437$, and $M_p = 4.5\%$. The optimal PID parameters are optimized

to achieve minimum fitness value according to the desired response specifications. The minimum fitness value (J_{min}) achieved by the optimum PID parameters is 0.009413.

In conventional PID optimization methods, such as ABC, MOL, or GA, the optimization search is performed within an open space of parameters set. For each parameter, a real range of values is defined such as the ranges selected in this section. The open space search delays the optimization process, consumes its effort, and may lead to local minimum problems. Therefore, combinatorial interaction design reduces the space of search and thus achieves effective optimization process.

The table of all PID parameters combinations is supplied to the process of combinatorial set construction explained in Figure 2 to produce 15 constructed sets of PID

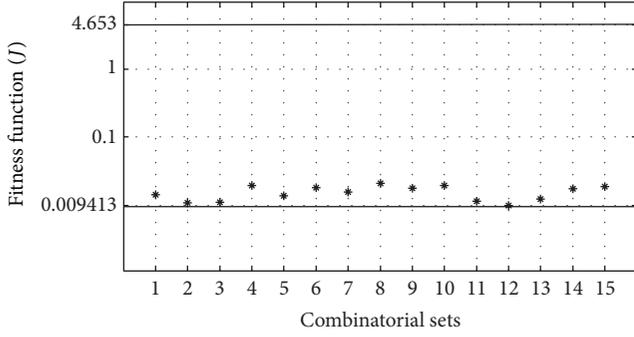


FIGURE 5: Optimal fitness function values of each optimal PID parameter combination corresponding to each constructed set.

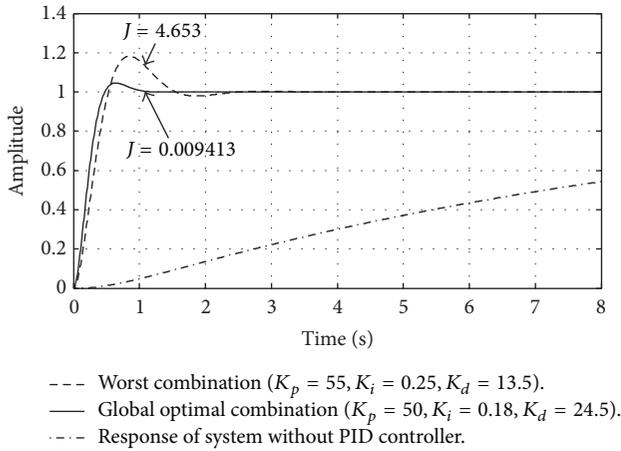


FIGURE 6: Step response of the optimal and worst combination PID parameters.

combinations. Each constructed set consists of 2571 PID parameters combinations which equals 2% of all PID parameters combinations table size. The parameters combinations of each constructed set are tested in the control system using the objective function defined in (4). The optimal parameters combination of each constructed set is the one with minimum fitness value. Therefore, from the 15 constructed sets, 15 optimal combinations of PID parameters are determined. Figure 5 depicts the optimal fitness function values of each optimal PID parameter combination corresponding to each constructed set.

The two indicated horizontal levels shown in Figure 5, that is, 0.009413 and 4.653, are the fitness function values of the global optimal combination parameters ($K_p = 50$, $K_i = 0.18$, and $K_d = 24.5$) and the worst combination parameters ($K_p = 55$, $K_i = 0.25$, and $K_d = 13.5$), respectively. Figure 6 shows the step response of the optimal and worst combination PID parameters as well as the step response of the DC motor without PID.

It can be observed from Figure 6 that the fitness function values of all the optimal PID parameters combinations corresponding to each constructed set are close to the global optimal value. This indicates that the combinatorial approach used to construct the subsets succeeded in reducing the

TABLE 2: Simulation results summary.

	K_p	K_i	K_d	t_r	t_s	t_p	M_p	J
Global optimal	50	0.18	24.5	0.4755	0.8784	0.6437	4.55%	0.0094
1	50	0.10	24.5	0.4756	0.8766	0.6438	4.53%	0.0137
2	50	0.21	24.5	0.4755	0.8791	0.6437	4.56%	0.0102
3	50	0.15	24.5	0.4756	0.8777	0.6438	4.55%	0.0105
4	50	0.02	24.5	0.4758	0.8747	0.6438	4.52%	0.0187
5	50	0.25	24.5	0.4754	0.8801	0.6437	4.56%	0.0131
6	50	0.31	24.5	0.4753	0.8815	0.6437	4.58%	0.0173
7	50	0.08	24.5	0.4757	0.8761	0.6438	4.53%	0.0150
8	50	0.35	24.5	0.4753	0.8824	0.6436	4.58%	0.0201
9	50	0.05	24.5	0.4757	0.8754	0.6438	4.53%	0.0168
10	50	0.02	24.5	0.4758	0.8747	0.6438	4.52%	0.0187
11	50	0.22	24.5	0.4755	0.8794	0.6437	4.56%	0.0109
12	50	0.18	24.5	0.4755	0.8784	0.6437	4.55%	0.0094
13	50	0.23	24.5	0.4754	0.8796	0.6437	4.56%	0.0116
14	50	0.30	24.5	0.4753	0.8812	0.6437	4.57%	0.0166
15	50	0.32	24.5	0.4753	0.8817	0.6436	4.58%	0.0180

searching space by producing subset that assures the presence of an optimal candidate to the optimization problem. Table 2 summarizes the simulation results for all the combinatorial constructed sets.

It can be observed from Table 2 that the timing parameters observed from the step response of each optimal PID parameter combination are almost equal to the corresponding desired values. In addition, the fitness values listed in the rightmost column are close to the global optimal fitness value.

6. Conclusion

In this paper, the combinatorial optimization has been employed in the application of PID controller design for DC servomotor. The objective of using the combinatorial design approach is to improve the PID tuning algorithm by reducing the search space of the optimization process. With different constructed combinatorial sets, simulation results have shown that in each constructed set there exists a combination of PID parameters with a fitness value close to the global optimal value. In this way, the proposed combination optimization method will ensure an optimal value finding by generating an effective reduced searching space. Thus, the proposed optimization based on combinatorial approach can be recommended to be adopted in conventional AI algorithms such as ABC, MOL, and GA.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This research is supported by the Information Technology Unit of the Engineering College, Salahaddin University-Hawler (SUH).

References

- [1] R. Rahmani, M. S. Mahmodian, S. Mekhilef, and A. A. Shojaei, "Fuzzy logic controller optimized by particle swarm optimization for DC motor speed control," in *Proceedings of the IEEE Student Conference on Research and Development*, pp. 109–113, 2012.
- [2] K. Ogata, *Modern Control Engineering*, Prentice Hall, New York, NY, USA, 2010.
- [3] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *Transactions of ASME*, vol. 64, pp. 759–768, 1942.
- [4] G. Cohen and G. Coon, "Theoretical consideration of retarded control," *Transactions of ASME*, vol. 75, no. 1, pp. 827–834, 1953.
- [5] S. Panda, "Differential evolution algorithm for SSSC-based damping controller design considering time delay," *Journal of the Franklin Institute*, vol. 348, no. 8, pp. 1903–1926, 2011.
- [6] S. Panda, "Multi-objective PID controller tuning for a FACTS-based damping stabilizer using Non-dominated Sorting Genetic Algorithm-II," *International Journal of Electrical Power and Energy Systems*, vol. 33, no. 7, pp. 1296–1308, 2011.
- [7] S. Panda, "Multi-objective evolutionary algorithm for SSSC-based controller design," *Electric Power Systems Research*, vol. 79, no. 6, pp. 937–944, 2009.
- [8] S. J. Ho, L. S. Shu, and S. Y. Ho, "Optimizing fuzzy neural networks for tuning PID controllers using an orthogonal simulated annealing algorithm OSA," *IEEE Transactions on Fuzzy Systems*, vol. 14, no. 3, pp. 421–434, 2006.
- [9] H. Gozde and M. C. Taplamacioglu, "Comparative performance analysis of artificial bee colony algorithm for automatic voltage regulator (AVR) system," *Journal of the Franklin Institute*, vol. 348, no. 8, pp. 1927–1946, 2011.
- [10] K. A. El-Metwally, "A fuzzy logic-based PID for power system stabilization," *Electric Power Components and Systems*, vol. 29, no. 7, pp. 659–669, 2001.
- [11] S. Panda and N. P. Padhy, "Comparison of particle swarm optimization and genetic algorithm for FACTS-based controller design," *Applied Soft Computing Journal*, vol. 8, no. 4, pp. 1418–1427, 2008.
- [12] S. Panda, B. K. Sahu, and P. K. Mohanty, "Design and performance analysis of PID controller for an automatic voltage regulator system using simplified particle swarm optimization," *Journal of the Franklin Institute*, vol. 349, no. 8, pp. 2609–2625, 2012.
- [13] D. R. Stinson, *Combinatorial Designs: Constructions and Analysis*, Springer, New York, NY, USA, 2004.
- [14] B. S. Ahmed and K. Z. Zamli, "A variable strength interaction test suites generation strategy using Particle Swarm Optimization," *Journal of Systems and Software*, vol. 84, no. 12, pp. 2171–2185, 2011.
- [15] B. S. Ahmed and K. Z. Zamli, "A review of covering arrays and their application to software testing," *Journal of Computer Science*, vol. 7, no. 9, pp. 1375–1385, 2011.
- [16] B. S. Ahmed, K. Z. Zamli, and C. P. Lim, "Application of Particle Swarm Optimization to uniform and variable strength covering array construction," *Applied Soft Computing Journal*, vol. 12, no. 4, pp. 1330–1347, 2012.
- [17] S. Y. Borodai and I. S. Grunskii, "Recursive generation of locally complete tests," *Cybernetics and Systems Analysis*, vol. 28, no. 4, pp. 504–508, 1992.
- [18] D. E. Shasha, A. Y. Kouranov, L. V. Lejay, M. F. Chou, and G. M. Coruzzi, "Using combinatorial design to study regulation by multiple input signals. A tool for parsimony in the post-genomics era," *Plant Physiology*, vol. 127, no. 4, pp. 1590–1594, 2001.
- [19] J. N. Cawse, *Experimental Design for Combinatorial and High Throughput Materials Development*, Wiley-Interscience, New York, NY, USA, 2003.
- [20] D. S. Hoskins, C. J. Colbourn, and D. C. Montgomery, "Software performance testing using covering arrays: efficient screening designs with categorical factors," in *Proceedings of the 5th International Workshop on Software and Performance*, pp. 131–136, ACM, Palma, Spain, July 2005.
- [21] D. R. Sulaiman and B. S. Ahmed, "Using the combinatorial optimization approach for DVS in high performance processors," in *Proceedings of the International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAECE '13)*, pp. 105–109, May 2013.
- [22] C. Yilmaz, M. B. Cohen, and A. Porter, "Covering arrays for efficient fault characterization in complex configuration spaces," *ACM SIGSOFT Software Engineering Notes*, vol. 29, no. 4, pp. 45–54, 2004.
- [23] A. Hartman and L. Raskin, "Problems and algorithms for covering arrays," *Discrete Mathematics*, vol. 284, no. 1–3, pp. 149–156, 2004.
- [24] M. Grindal, J. Offutt, and S. F. Andler, "Combination testing strategies: a survey," *Software Testing Verification and Reliability*, vol. 15, no. 3, pp. 167–199, 2005.
- [25] P. McMinn, "Search-based software test data generation: a survey," *Software Testing Verification and Reliability*, vol. 14, no. 2, pp. 105–156, 2004.
- [26] B. S. Ahmed and K. Z. Zamli, "Comparison of metaheuristic test generation strategies based on interaction elements coverage criterion," in *Proceedings of the IEEE Symposium on Industrial Electronics and Applications (ISIEA '11)*, pp. 550–554, IEEE Computer Society, Langkawi, Malaysia, September 2011.
- [27] N. B. Nichols and J. G. Ziegler, "Optimum settings for automatic controllers," *Journal of Dynamic Systems, Measurement and Control*, vol. 115, no. 2, pp. 220–222, 1993.
- [28] R. A. Krohling, J. P. Rey, R. A. Krohling, and J. P. Rey, "Design of optimal disturbance rejection PID controllers using genetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 5, no. 1, pp. 78–82, 2001.
- [29] Q. H. Wu, B. W. Hogg, and G. W. Irwin, "A neural network regulator for turbogenerators," *IEEE Transactions on Neural Networks*, vol. 3, no. 1, pp. 95–100, 1992.
- [30] A. Visioli, "Tuning of PID controllers with fuzzy logic," *IEEE Proceedings: Control Theory and Applications*, vol. 148, no. 1, pp. 1–8, 2001.
- [31] M. R. AlRashidi and M. E. El-Hawary, "A survey of particle swarm optimization applications in electric power systems," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 4, pp. 913–918, 2009.

- [32] A. Jalilvand, A. Kimiyaghalam, A. Ashouri, and H. Kord, "Optimal tuning of PID controller parameters on a DC motor based on advanced particle swarm optimization algorithm," *International Journal on Technical and Physical Problems of Engineering (IJTPE)*, vol. 3, no. 9, pp. 10–17, 2011.
- [33] M. M. R. A. Milani, T. Cavdar, and V. F. Aghjehkand, "Particle swarm optimization—based determination of Ziegler-Nichols parameters for PID controller of brushless DC motors," in *Proceedings of the International Symposium on Innovations in Intelligent Systems and Applications (INISTA '12)*, 2012.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

