Çukurova Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi, 27(2), ss. 13-26, Aralık 2012
Çukurova University Journal of the Faculty of Engineering and Architecture, 27(2), pp. 13-26, December 2012

Multi-objective Bees Algorithm to Optimal Tuning of PID Controller

Ramazan ÇOBAN*[1], Özden ERÇİN[1]
[1] *Çukurova University, Department of Computer Engineering, 01330 Balcali, Adana*

Abstract

In this paper, a novel intelligent design method for closed-loop auto-tuning of a proportional-integral-derivative (PID) controller based on the Multi-Objective Bees Algorithm (MOBA) is proposed, by which PID controller parameters can be tuned concurrently in order that the set of trade-off optimal solutions that is called Pareto-set optimization solution of the conflicting objective functions are able to be found. Comparing the multi-objective bees algorithm with Ziegler–Nichols, modified genetic algorithm and ant colony optimization, simulation results demonstrate that the new tuning method using the multi-objective bees algorithm has a better control system performance. Moreover, this method is applied to a direct current (dc) motor speed control in order to make comparison and show the performance of the multi-objective bees algorithm. The results obtained show good stability, set-point tracking performance and robustness.

Key words: Multi-objective optimization, Bees algorithm, PID tuning.

Çok Amaçlı Arı Algoritması Kullanarak PID Kontrolörün Parametrelerinin Optimal Ayarlanması

Özet

Özellikle ülkemizde inşaat sektöründe iletişimin önemi yeterince bilinmemektedir. Bundan dolayı iş süreçlerinde çeşitli aksaklıklar yaşanmaktadır. Bu çalışmada, inşaat işletmelerinde iletişimin yeri ve önemi konusunda farkındalık yaratmak ve iletişim kaynaklı sorunları en aza indirmek için öneriler geliştirilmiştir. Bunun için öncelikle sektördeki iletişim kaynaklı sorunlar saptanmaya çalışılmış, ardından bunların çözümüne yönelik öneriler sunulmuştur. Sektör yapısı ve ilişki farklılıkları göz önünde bulundurularak konu işletme içi ve dışı iletişim olmak üzere iki aşamada incelenmiştir.

Anahtar Kelimeler: Çok Amaçlı Optimizasyon, Arı Algoritması, PID Parametrelerinin Ayarlaması.

* Yazışmaların yapılacağı yazar: Ramazan ÇOBAN, *Çukurova University, Department of Computer Engineering, 01330 Balcali, Adana.* rcoban@cu.edu.tr

1. INTRODUCTION

Despite of all of the development in control systems over the past several decades, the proportional, integral and derivative (PID) controller remains the most widespread kind of feedback controller in use today [1]. Industrial PID control schemes based on the classical control theory have been widely used for miscellaneous process control systems for many years. They have been preferred for their functional simplicity, good robust performance and easy implementation in a wide range of operating conditions; furthermore, PID controller principle is easier to understand than other traditional controllers for the majority of industrial processes. However, since the performance of a PID controller completely depends on the tuning of its parameters many industrial plants are often confronted with many problems such as higher order, time delays and nonlinearities [2]. In many papers [3-6], different PID control methods have been applied to determine three parameters of PID controller for the given processes. Several algorithms such as manual tuning, Ziegler-Nichols [3], Cohen-Coon [6], etc. have their own advantages and disadvantages. The major drawback of the manual tuning method is that it requires experienced personnel. Some shortcomings of the Ziegler-Nichols method are the resulting in large overshoot and oscillatory responses. Besides, controller settings necessitate very aggressive tuning and also further fine tuning. This method has also poor performance for processes with a dominant delay and closed loop system is very sensitive to parameter variations, so parameters of the step response may be hard to determine due to measurement noise. A common disadvantage of the Cohen-Coon method is that it can only be used for first order models including large process delays.

In the last decades, design engineers have focused on evolutionary based approaches to improve the existing design theories and find the best design results to tune the parameters of PID controllers. The main weakness of Genetic Algorithm (GA) among evolutionary based approaches is a lack of guarantee that global optimum is found within limited period of time and slower speed of convergence. A disadvantage of Ant Colony Optimization (ACO) which is another evolutionary based approach is difficulty of theoretical analysis, sequences of random decisions and probability distribution changes by iteration. In addition, a long convergence time is a significant drawback of it but convergence is guaranteed. The properly selection of the PID parameters is so important that the closed loop system must meets design specifications. The design specifications can include minimum or no overshoot, minimal rise time, minimal steady state error and settling time in the step response of the closed loop system. The Multi-Objective Bees Algorithm (MOBA) has been used successfully to solve many engineering and scientific problems and applied to constrained and unconstrained single objective function optimizations [7-11]. In this work, the MOBA was applied to optimize the parameters of PID controllers. To indicate the effectiveness and efficiency of the proposed optimization method, the step responses of closed loop systems were compared with those of the existing methods in the literature such as Ziegler–Nichols [3], genetic algorithm [4] and ant colony optimization [5]. In addition, performance and set-point tracking capability of the purposed method were tested through various linear plants as well as a DC motor [12-14].

In this paper, an optimal or near optimal PID controller for different order transfer functions and a DC motor are developed using the MOBA. The paper is organized as follows. PID controller design is introduced in Section 2. Principle of multi-objective optimization is introduced in Section 3. The PID controller design is formulated using the multi-objective bees algorithm in Section 4. Principle of Bees Algorithm (BA) and multi-objective optimization combined with BA are introduced in Section 5. In Section 6, the simulation results of several controllers for different order linear plants and a DC motor speed using the multi-objective bees algorithm are reported in order to make comparison and illustrate the performance of the proposed method. Finally, the conclusions are given in Section 7.

2. DESIGN OF PID CONTROLLER

PID controller widely used in industrial control systems is composed of proportional control action, integral control action and derivative control action. There are many forms of PID controller implementations such as a stand-alone controller or Distributed Control System (DCS). Figure 1 is a simple diagram showing the schematic of the PID controller and it is known as non-interacting form or parallel form.

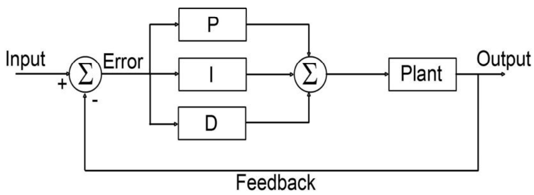

Figure 1. Block diagram of a PID controller

The parallel controllers are mostly preferred for higher order systems. The transfer function of PID controller in Laplace transform is defined for a continuous system as

$$G_c(s) = K_p + \frac{K_i}{s} + K_d s \quad (1)$$

The proportional controller response is proportional to the control error. The controller error is defined as the difference between the set point and the process output. The proportional controller output is the multiplication of the system error signal and the proportional gain. Proportional term can be mathematically expressed as

$$P_{term} = K_p \times Error \quad (2)$$

The integral control applies a control signal to the system which is proportional to the integral of the error. The offset introduced by the proportional control is removed by the integral action but a phase lag is added into the system. Integral term can be mathematically expressed as

$$I_{term} = K_I \times \int Error\, dt \quad (3)$$

There is a proportion between the derivative controller output and the rate of change of the error. Derivative control is used to decrease and eliminate overshoot of system response and introduce a phase lead action that removes the phase lag introduced by the integral action.

$$D_{term} = K_D \times \frac{d(Error)}{dt} \quad (4)$$

Combining these three types of control together, transfer function of continuous PID controller is formed as

$$G_c(s) = \frac{K_D s^2 + K_P s + K_I}{s} \quad (5)$$

where K_p, K_i and K_d are the proportional, integral and derivative gains, respectively. The control signal to the plant is given by

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt} \quad (6)$$

To design the proposed controller, four important characteristics of the output of the system are used. These four characteristics are briefly defined below and illustrated in Fig. 2.

Rise time (t_r) is defined as the time required for the step response to rise from 10% to 90% of the set point.

Settling time (t_s) is defined as the time required for the step response to stay within 2% of the set point.

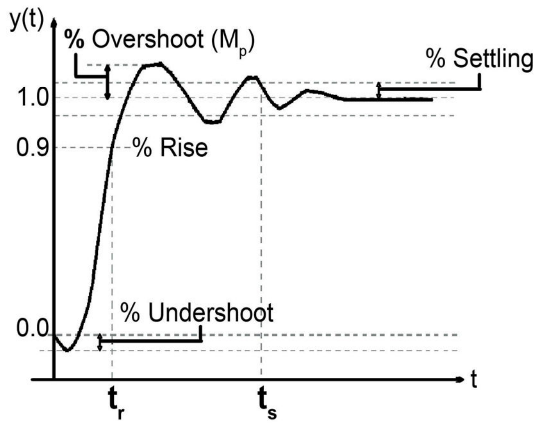

Figure 2. Rise time, setting time, and maximum overshoot

Maximum overshoot (M_p) characterizes what maximum peak value will be reached over the set point. If y_{max} designate the maximum value of y and y_{ss} show the steady-state value of it, the maximum overshoot will be expressed as:

$$M_p = y_{\max} - y_{ss} \tag{7}$$

Performance Index : *IAE*, *ISE, ITAE*, and *ITSE* are typically and popular integral error criteria [15]. Some error criteria usually have to be minimized to get the PID tuning parameters optimal or near optimal. The Integral Absolute Error (IAE) in the controlled variable is formulated by

$$IAE = \int_0^\tau |e(t)| dt = \int_0^\tau |r(t) - y(t)| dt \tag{8}$$

Now that large errors penalized by the *ISE* criterion results in the most-aggressive settings and persistent errors penalized by the *ITAE* criterion results in the most-conservative settings, moderate settings are produced between *ISE* and *ITAE* criteria by the *IAE* criterion [16]. The Integral Square Error (ISE), Integral Time Absolute Error (ITAE), and Integral Time Square Error (ITSE) are also given as follows:

$$ISE = \int_0^\tau e(t)^2 dt = \int_0^\tau (r(t) - y(t))^2 dt \tag{9}$$

$$ITAE = \int_0^\tau t|e(t)| dt = \int_0^\tau t|r(t) - y(t)| dt \tag{10}$$

$$ITSE = \int_0^\tau te(t)^2 dt = \int_0^\tau t(r(t) - y(t))^2 dt \tag{11}$$

3. PRINCIPLE OF MULTI-OBJECTIVE OPTIMIZATION

The multi-objective optimization is used to minimize all the objective design criteria functions simultaneously. The general multi-objective optimization requiring the optimization of j objectives can be written as follows [17]:

$$\underset{x \in D}{Min} \{ f_1(x) = y_1, f_2(x) = y_2, ..., f_j(x) = y_j \} \tag{12}$$

In Eq.(12), $f_j(x)$ are the j objective design criteria functions and x indicates the design parameters chosen and D indicates the set of possible design parameters. There are response surface functions $f_j(x)=y_j$ of each response. When there exists a vector of non negative weights $\Phi=[\lambda_1 ..... \lambda_j]^T$, an efficient solution is supported. Unique global optimum x is expressed in the following formula [17]:

$$\underset{x \in D}{Min} \sum_{j=1}^{N} \lambda_j f_j(x) \tag{13}$$

If there does not exist a conflict between the objective functions in Eq.(12), then a solution x, called an *ideal solution*, can be found where every objective function obtains its minimum. Generally, there is not an ideal single solution which is optimal with respect to each objective function. The objective functions are mostly in conflict, that's why the reduction of one objective function usually causes to increase another objective functions. Consequently, Pareto optimal solution is the result of the multi-objective optimization and this solution is possible to improve any of the objective function by increasing at least one of the other objective functions. Pareto optimality can not improve any criterion without deteriorating a value of at least one other criterion. There are generally a lot of Pareto optimal solutions. There is an equally acceptable solution of the problem for every Pareto optimal point. Nevertheless, the aim is generally desirable to obtain one point as a solution [17].

A solution vector $x^* \in \boldsymbol{X}$ is called *Pareto optimal* when there does not exist another solution which dominates it in Eq.(14). That is to say, solution can improve in one of the objectives if it affects at least one other objective [17].

$$\begin{aligned} &\exists x \in X : f_i(x) \le f_i(x^*) \wedge f(x) \ne f(x^*); \\ &\forall i = \{1, 2, ..., p\} \end{aligned} \tag{14}$$

The corresponding objective vector $f(x^*)$ is said to be a *Pareto dominant* vector. A solution vector x_1

dominates another feasible solution x_2, ($x_1 > x_2$) such as:

$$f_1(x_1) \le f_i(x_2) \wedge \exists j : f_j(x_1) \le f_j(x_2);\quad \forall i, j = \{1, 2, ..., p\} \tag{15}$$

If there doesn't exist any solution that dominates x_1, then x_1 is non-dominated. A set of non-dominated feasible solutions $\{x^* \mid \neg\exists x : x > x^*\}$ is called the *Pareto optimal set*. The set of objective vectors which are image of a Pareto set $\{F(x^*) \mid \neg\exists x : x > x^*\}$ is said to be on the *Pareto front*. A Pareto front for a bi-objective optimization problem is illustrated in Fig. 3.

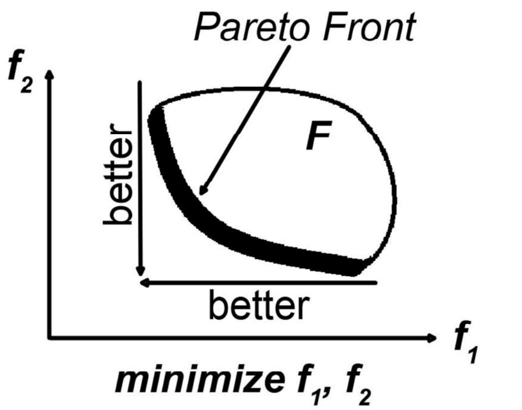

Figure 3. Illustration of Pareto front for a bi-objective optimization problem

4. DESIGN OF PID CONTROLLERS USING THE MULTI-OBJECTIVE BEES ALGORITHM

Tuning the parameters of the PID controllers using the multi-objective bees algorithm is an optimization problem which needs to be solved in such a way that output of the system attains the desired level in the shortest time as far as possible preventing a high overshoot at the same time. In feedback control loop denoted by Fig. 4, G_c presents the PID controller that is governed by Eq.(5), and G_p presents the system to be controlled. In Fig. 4, r denotes the reference input signal, e denotes the error signal, u denotes the control signal, y denotes the output signal, G_p denotes a Linear Time-Invariant (LTI) system, G_c denotes the PID Controller. Using the reference signal $r(t)$ and system output $y(t)$ the error signal are defined as $e(t) = r(t) - y(t)$.

Optimization criteria which are used to evaluate fitness are to be chosen in applying optimization method. Many indexes of PID controller affecting performance of the transient response can be combined into one objective function composed of the weighted sum of objectives. The set of objective function is represented by Eq. (16):

$$J^B = \min(\Phi F) \tag{16}$$

Where J^B denotes the value of the objective function found by the bees, $F = [f_1\, f_2\, f_3\, f_4\, f_5\, f_6\, f_7]^T$ denotes vector of objective functions, f_1 denotes the first objective function including the settling time (t_s), f_2 denotes the second objective function including rise time (t_r), f_3 denotes the third objective function including maximum overshoot (M_p), f_4 denotes the fourth objective function including Integral Absolute Error (IAE), f_5 denotes the fifth objective function including Integral Time Absolute Error (ITAE), f_6 denotes the sixth objective function including Integral Square Error (ISE), f_7 denotes the seventh objective function including Integral Time Square Error (ITSE), $\Phi = [\lambda_1\, \lambda_2\, \lambda_3\, \lambda_4\, \lambda_5\, \lambda_6\, \lambda_7]$ denotes vector of non-negative weights.

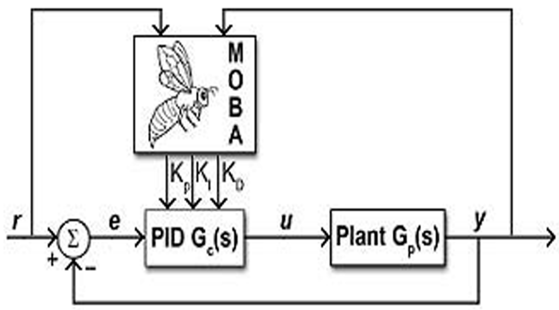

Figure 4. Block diagram of the MOBA-PID controller

5. THE MULTI-OBJECTIVE BEES ALGORITHM OPTIMIZATION

The bees algorithm which imitates the food foraging behavior of honeybees colony is a novel swarm-based search algorithm developed by Pham et al in 2006 [7]. The bees algorithm is based on a kind of neighborhood search combined with random search and can be used for multi-objective optimization [18]. The key steps of the BA are shown in Fig. 5 [19]. A number of parameters in the algorithm need to be set in advance: *n* denotes number of scout bees, *m* denotes number of sites selected for neighborhood search (out of n visited sites), e_s denotes number of top-rated (elite) sites among *m* selected sites, *nep* denotes number of bees recruited for the selected sites, *nsp* denotes number of bees recruited for the other $(m-e_s)$ selected sites, *ngh* denotes the initial size of each patch (a patch is a site in the search space that includes the visited site and its neighborhood), *sc* denotes shrinking constant [7].

At the beginning of the algorithm *n* scout bees are randomly distributed in the search space. The evaluation of the sites visited by the scout bees using the fitness function (i.e. the performance of the candidate solutions) is in step 2.

Bees Algorithms Steps

1. Start initial population with random solutions.
2. Calculate fitness of the population.
3. While (stopping criterion not met)
   >> Generate new population.
4. Choose sites for neighborhood search.
5. Determine the patch size.
6. Recruit bees for *chosen sites* and calculate fitness.
7. Choose the representative bee from each patch.
8. Amend the Pareto optimal set.
9. *Leave sites without new information.*
10. Assign remaining bees to search randomly and calculate their fitness.
11. End While.

Figure 5. Pseudo Code of the Bees Algorithm [19]

The *m* non-dominated sites are assigned as "*selected sites*" and preferred for neighborhood search in step 4. If there exist more than *m* non-dominated sites in the population, the first *m* will be chosen as it is impossible to tell the difference between them. If there are less than *m* non-dominated sites, from the dominated sites that have been dominated just once, the rest will be chosen and this subroutine is continued until an adequate number of sites have been chosen. In step 5, a wide patch dimension is selected initially. For each patch, the initial size is provided as unchanged only when the recruited bees can get better solutions in the neighborhood. When there does not exist any progress in the neighborhood search, the patch size is diminished. The purpose of this strategy is to make the local search more exploitative, to search more intensely the surrounding of the local optimum. Therefore this step is named as the "shrinking method" [19].

The algorithm searches around the chosen sites in step 6. In the multi-objective optimization adaptation of the bees algorithm, it is not always possible to rank the solution candidates. That is why, all the *selected sites* include the same number of recruited bees to search around the neighborhood. When one of the recruit bees dominates the original bee, the representative bee will be called "new non-dominated bee", in step 7. The aim of step 8 added to the fundamental bees algorithm is to solve multi-objective optimization problems. A non-dominated solution will be added to the Pareto optimal set. Furthermore, if the other solutions in the created Pareto optimal set are dominated by this solution, the dominated ones will be eliminated from the set.

In step 9, when there doesn't exist any improvement in using *shrinking method* presumed the patch is centered on a local peak of performance of the solution space. If the neighborhood search finds a local optimum, it is impossible to get further progress. As a result, the exploration of the patch is concluded. Therefore, this step is named as the "*abandon sites without new information*" [19]. In step 10, the remaining bees in the population aiming to find new potential solutions are placed randomly around the search space.

At the end of each iteration, there are two parts of colony to its new population. These are representative bees from the selected patches and scout bees appointed to attitude random searches. These steps are repeated till a stopping criterion is satisfied.

6. SIMULATION RESULTS

In this study, two examples were utilized in order to illustrate the efficiency of the proposed algorithm: (i) the multi-objective optimization of PID controllers comparing two linear plants with different order, (ii) the multi-objective optimization of PID controller design for a DC motor. The proposed method was generated in C code. The values of the parameters in the MOBA are n=200 (numbers of scout bees), m=80 (numbers of selected locations), nsp=40 (numbers of bees around each selected locations, except the elite location), nep=60 (numbers of bees around each elite locations), e_s=20 (numbers of top-rated (elite) sites), ngh=6 (neighborhood patch size), sc=2 (shrinking constant, defined as percentage (%) and range is between 0-100), iter=2000 (number of iteration) and maximum population=1000. The multi-objective bees algorithm searched based on the well known *Routh-Hurwitz* criterion within the stability boundary for controller tuning. Poles of a transfer function in the s-plane should be located to the left of the jw axis for stability by the MOBA.

6.1. The MOBA PID design for linear plants with different order

The performance of the MOBA was tested with two plants with different order. The objective function which should be minimized was composed of the objective functions f_1, f_2, f_3, and f_6 which include the settling time, the rise time, the maximum overshoot, and the integral square error, respectively. The vector of weights is defined as Φ = [0.000001 0.0001 1 0 0 0.0001 0]. Throughout the optimization process, the MOBA uses the step reference input and closed loop step response of the process.

The tuning algorithm looks for the optimal parameters for the PID controller to satisfy the desired system specifications by using the changed closed loop control performance according to the adjusted controller parameters at the each iteration. The closed loop response was compared with a step change of a number of simulated systems in order to demonstrate the effectiveness of the presented method. For PID controller tuning, two various processes with different order are used as follows [4]:

$$G_1(s) = \frac{4.228}{\left(s+0.5\right)\left(s^2+1.64s+8.456\right)} \tag{17}$$

$$G_2\left(s\right) = \frac{27}{(s+1)(s+3)^3} \tag{18}$$

In control system applications, a weighted combination of different performance characteristics such as rise time, settling time, maximum overshoot and integral of the square of the error is the chosen performance criterion. The desired system response needs minimal rise time, minimal settling time with a small or no overshoot in the step response of the closed loop system. Hence, the objective function F is defined using the performance indices consisting of integral of the square of the error (*ISE*), rise time (t_r), settling time (t_s) and percentage overshoot (M_p).

$$F = \lambda_1(t_s) + \lambda_2(t_r) + \lambda_3(M_p) + \lambda_6(ISE) \tag{19}$$

In Eq. 19, the weighting factors are the variables of λ_1, λ_2, λ_3 and λ_6. By adjusting these factors, the most convenient PID controller parameters can be provided in order to achieve the desirable closed loop characteristics of the system. For the predetermined control objectives the performance of the PID controller can be significantly improved. To obtain better solution, weighting factors are defined as $\lambda_1=0.000001$, $\lambda_2=0.0001$, $\lambda_3=1$, $\lambda_4=0$, $\lambda_5=0$, $\lambda_6=0.0001$, $\lambda_7=0$ and the best results obtained for the parameters of the PID controlled systems optimized by the algorithm are given in Table 1. The results in this table were found by using 200 scout bees. For avoiding a similar particular solution, the initial populations were generated at random within the range $0.0 \leq K_p \leq 3.0$, $0.0 \leq K_i \leq 2.0$, $0.0 \leq K_d \leq 3.0$. The closed loop PID controller was tuned for the values K_p, K_i and K_d first by using Ziegler-Nichols method [3], genetic algorithm [4] and ant colony algorithm [5]. In addition, the closed loop PID controller was tuned by using the MOBA.

Table 1. Simulation results of the PID controlled systems for different order systems

Plant	Parameters	Ziegler Nichols [3]	Genetic Algorithm [4]	Ant Colony Optimization [5]	MOBA
G1(s)	K_P	2.19	1.637	2.517	2.6213
	K_I	2.126	0.964	2.219	0.8719
	K_D	0.565	0.387	1.151	2.4816
	f_1:t_s	6.6	5.97	6.51	6.5249
	f_2:t_r	0.8	2.45	0.627	0.4553
	f_3:M_p	%16.46	%3	%16	%0.0513
	f_6:ISE	0.785	0.588	0.684	0.400
G2(s)	K_P	3.072	1.772	2.058	2.2974
	K_I	2.272	1.061	1.137	1.1017
	K_D	1.038	0.772	0.746	1.2176
	f_1:t_s	5.1	2.91	4.34	3.9734
	f_2:t_r	0.7	1.2	0.971	0.8547
	f_3:M_p	%32.53	%1.17	%6.62	%0
	f_6:ISE	0.66	0.7311	0.708	0.514

Table 2. Simulation results of the proposed algorithm for different error criteria

Plant	Parameters	MOBA IAE Error	MOBA ITAE Error	MOBA ITSE Error	MOBA MSE Error
G1(s)	K_P	2.4046	2.7860	2.7193	2.7895
	K_I	0.8973	0.9345	0.9745	0.9425
	K_D	1.3700	2.2043	1.8512	2.2173
	f_1:t_s	5.7213	6.5438	6.6587	6.5190
	f_2:t_r	0.7335	0.4804	0.5366	0.4782
	f_3:M_p	%0.0564	%0.0026	%0.0891	%0.0245
G2(s)	K_P	2.1449	2.0105	1.6602	2.4195
	K_I	1.0811	1.0578	0.9633	1.1206
	K_D	1.1892	1.0262	0.6928	1.4801
	f_1:t_s	3.8912	3.9170	3.8488	4.8280
	f_2:t_r	0.9111	1.0026	1.2461	0.77
	f_3:M_p	%0.0031	%0.000051	%0	%0

The results in Table 1 show that the value of the maximum overshoot is quite smaller, nearly zero percent and the values of the rise time, the settling time for all error criteria obtained by the MOBA are much less than the values of the other methods. The results of the other methods in Table 1 were taken from existing literature [3-5].

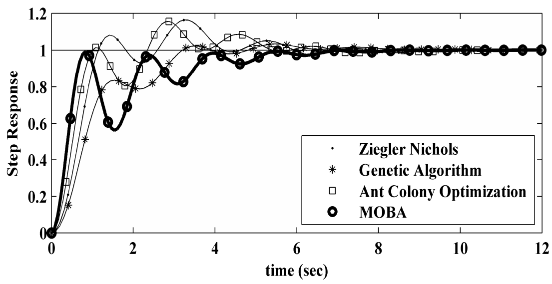

Figure 6. Comparison of step responses of the plant $G_1(s)$

Furthermore, the step responses of $G_1(s)$ and $G_2(s)$ tested with the optimum values of the parameters K_p, K_i and K_d which are obtained by the MOBA are presented in Table 2 according to some error criteria.

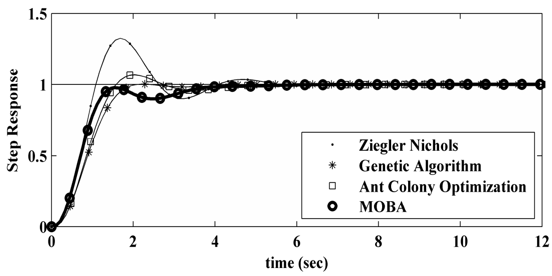

Figure 7. Comparison of step responses of the plant $G_2(s)$

The step responses of $G_1(s)$ and $G_2(s)$ respectively plotted with the optimum values of the parameters K_p, K_i and K_d which are obtained by the MOBA are shown in Figures 6 and 7. Step response results of two different processes obtained by using Ziegler Nichols, genetic algorithm and ant colony optimization algorithm are represented for comparison purposes.

Figures 8 and 9 illustrate the graphs of the obtained three-dimensional Pareto optimal fronts consisting of the settling time, overshoot and ISE error criteria for the step response of two different processes related with each transfer function.

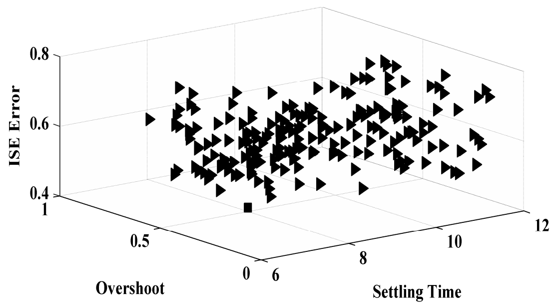

Figure 8. Multi-objective optimization Pareto-sets of the plant $G_1(s)$

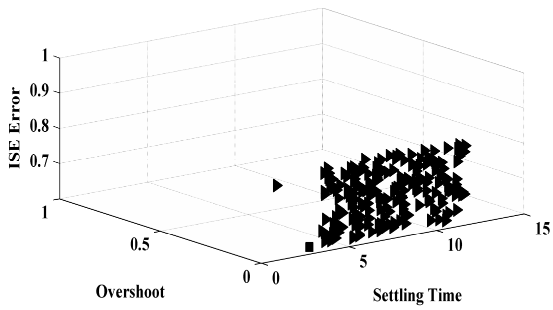

Figure 9. Multi-objective optimization Pareto-sets of the plant $G_2(s)$

Thus, a well distributed set of non-dominated solutions along the Pareto-optimal front can be found. The MOBA gives better responses than those produced by using the other methods. Thus, it can be considered that the MOBA improves the optimal system performance of the PID controllers satisfactorily. Evaluation of the objective function on the above mentioned two plants is presented in Figures 10 and 11. It is also observed that the objective function value decreases substantially and smoothly. As seen in Figures 6 and 7, the controlled systems show oscillations, especially much more in the plant $G_1(s)$. Sometimes oscillation affects stability of the controlled plants.

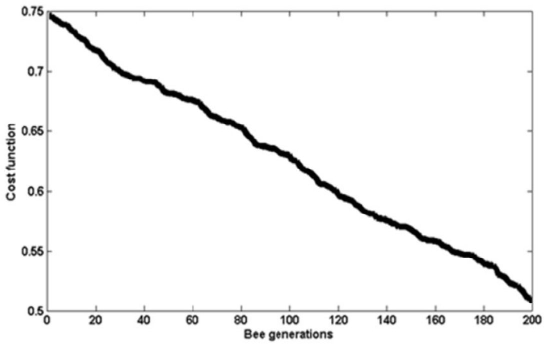

Figure 10. Convergence graph of the plant $G_1(s)$ in the MOBA method

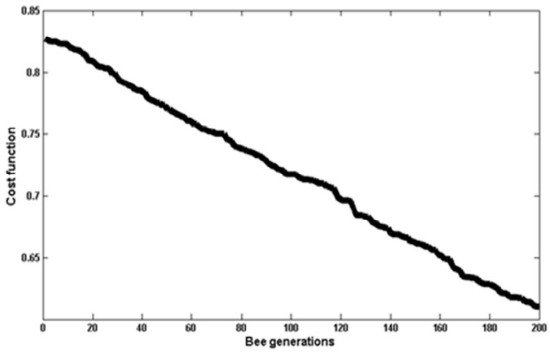

Figure 11. Convergence graph of the plant $G_2(s)$ in the MOBA method

It also causes undesirable situations. In order to cope with this problem generally some of the design specifications are modified in control system design. The smoother responses were achieved with slight concessions to the rise time. This time the vector of weights was defined as Φ = [0.000001 0.000001 1 0 0 0.0001 0]. While increasing the rise time caused longer settling time for the plant $G_1(s)$, it shortened the settling time for the plant $G_2(s)$. Nevertheless, the controlled systems gave fast response without overshoot and oscillation as seen in Figures 12 and 13. The obtained results are presented in Table 3.

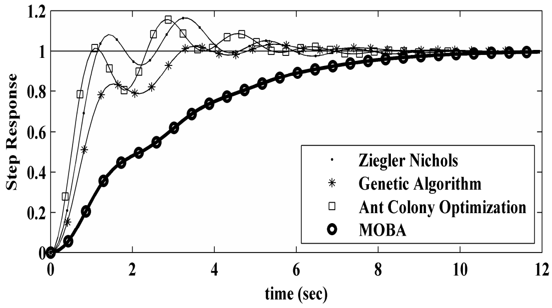

Figure 12. Step responses of the plant $G_1(s)$ with increasing rise time

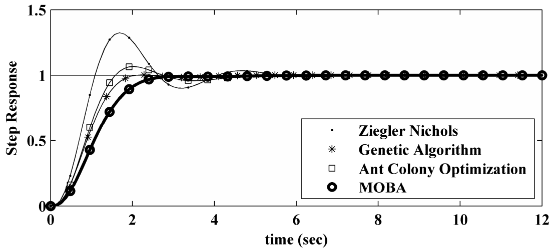

Figure 13. Step responses of the plant $G_2(s)$ with increasing rise time

Table 3. Simulation results of the proposed algorithm with increasing rise time

Plant	Parameters	MOBA ISE Error
$G_1(s)$	K_P	0.6138
	K_I	0.3521
	K_D	0.1120
	f_1:t_s	9.4256
	f_2:t_r	5.6385
	f_3:M_p	%0.0887
$G_2(s)$	K_P	1.3927
	K_I	0.8561
	K_D	0.5236
	f_1:t_s	2.5920
	f_2:t_r	1.4961
	f_3:M_p	%0

6.2. The MOBA PID design for a DC motor speed control

A design method is presented for a DC motor speed control. Simplified mathematical model of a DC motor has been used in order to build the DC motor's transfer function. There are differential equations of the electrical part and mechanical part in the DC motor model and also there exists an interconnection between them.

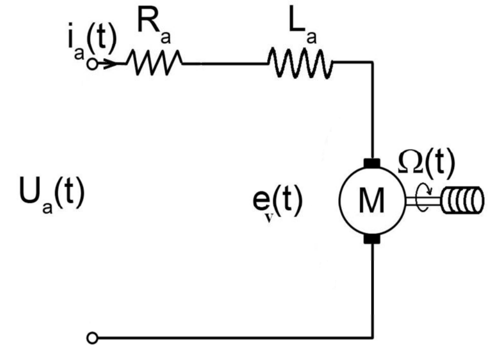

Figure 14. A DC Motor model

Using simplified equivalent electromechanical diagram of the DC motor, illustrated in Fig. 14, the differential mathematical model is written in Eq. (20-23) [12].

$$U_a(t) = R_a i_a(t) + L_a \frac{di_a(t)}{dt} + e_v(t) \quad (20)$$

$$e(t) = K_e \Omega(t) \quad (21)$$

$$C_m(t) = K_m i_a(t) \quad (22)$$

$$C_m(t) = J \frac{d\Omega(t)}{dt} + B\Omega(t) \quad (23)$$

Where C_m denotes motor torque (Nm), I_a denotes rotor circuit current (A), K_e denotes electrical constant, K_m denotes mechanical constant, L_a denotes rotor circuit inductance (H), R_a denotes rotor circuit resistance (Ohm), U_a denotes input voltage (V), B denotes friction ratio (Nms), e_v denotes electromotive voltage (V), J denotes rotor moment of inertia (kgm^2), Ω denotes rotor speed (rad/s).

The transfer function of speed model is obtained to allow the control of speed by the voltage input from the characteristic equations of the DC motor. It is presented by Eq. (24):

$$\frac{\Omega(s)}{U_a(s)} = \frac{K_m}{L_a J s^2 + (R_a J + L_a B)s + (R_a B + K_e K_m)} \quad (24)$$

This transfer function makes possible to simulate motor behavior to various inputs. The specifications of the motor used for simulation are given in Table 4.

Table 4. Parameters of the DC Motor [14]

Parameters	**Value**
Armature circuit resistance (R_a)	21.2 ohm
Torque constant (K_e)	0.1433 V/rad/s
Back-Emf constant (K_m)	0.1433 Kg-m/A
Coefficient of friction (B)	1×10^{-4} Nms
Armature circuit inductance (L_a)	0.052 H
Moment of inertia (J)	1×10^{-5} kgm^2

In this example, the objective function was composed of f_1, f_2, f_3, and f_6 which include the settling time, the rise time, the maximum overshoot and the integral square error, respectively. The vector of weights was defined as $\Phi = [0.000001 \;\; 0.0001 \;\; 1 \;\; 0 \;\; 0 \;\; 0.0001 \;\; 0]$. The results in Table 1 were found by using 200 scout bees and the initial populations were generated at random within the range $0.0 \le K_p \le 100.0$, $0.0 \le K_i \le 100.0$, $0.0 \le K_d \le 0.05$.

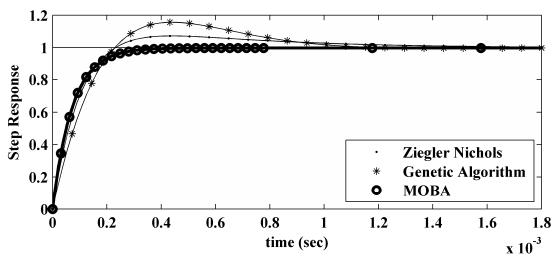

Figure 15. Comparison of step responses of the DC motor

The plant given in Eq. (24) was tested with a unit step input to show the effectiveness and performance of the proposed method. Three other approaches such as Ziegler-Nichols, genetic algorithm and ant colony algorithm were applied in order to make comparison and show the performance of the MOBA. The step response of the DC motor is depicted in Fig. 15. The obtained simulation results are given in Table 5.

Table 5. Simulation result of DC Motor Speed PID control

Plant	Parameters	Ziegler Nichols	Genetic algorithm [20]	MOBA
$\frac{\Omega(s)}{U_a(s)} = \frac{0.1433}{5.2\times10^{-7}s^2+2.172\times10^{-4}s+0.0227}$	K_P	70.556	93.1622	21.8463
	K_I	50	38.6225	48.4252
	K_D	0.039567	0.027836	0.0492
	$f_1(t_s)$	11×10^{-4}	9.83×10^{-4}	2.84×10^{-4}
	$f_2(t_r)$	1.57×10^{-4}	1.71×10^{-4}	1.61×10^{-4}
	$f_3(M_p)$	%7.166	%15.609	%0

The simulation results on the plant and the average values of standard performance measures where the objective function depends on the standard performance measures such as rise time, settling time and maximum overshoot are summarized in Table 5. Figure 16 presents the distribution of the non-dominated solutions in Pareto-optimal front using the proposed multi-objective bees optimization and the results show that the MOBA is able to find the Pareto front with good distribution of the solutions. The convergence of the objective function is depicted in Fig. 17. It can be seen from the figure that the objective function value decreases considerably.

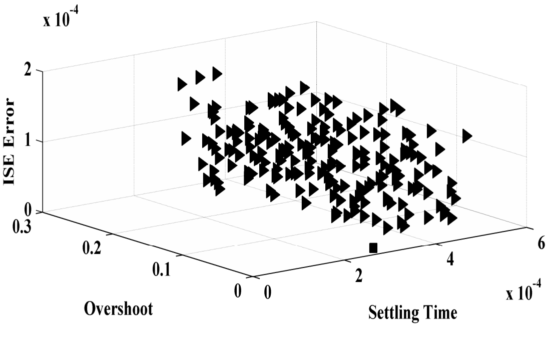

Figure 16. Multi-objective optimization Pareto-sets of the DC motor

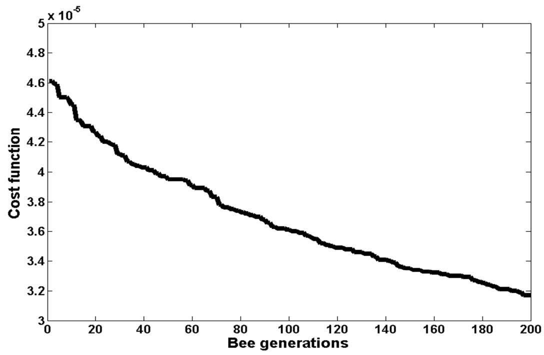

Figure 17 Convergence graph of the DC motor by using MOBA method

7. CONCLUSIONS

In this paper, a novel intelligent tuning design method for determining the PID controller parameters based on the Multi-Objective Bees Algorithm (MOBA) optimization is developed for getting good performances and tuning the Pareto-optimal PID parameters. The step response performance of the MOBA was tested with different order linear plants. It is well known that the Bees Algorithm has good results in solving numerical optimization problems. Thus, the effectiveness of the PID controller design using the MOBA was researched and was obtained a satisfactory performance. This study was also applied to tune PID controller parameters of a DC motor commonly used in industry and compared with some existing methods. The simulation results show that the new PID control tuning method using the MOBA achieve minimum overshoot and optimal or near optimal system performance. Due to the fact that some stability criteria are taken into account in the control system design, the proposed method thus can be regarded as a general controller design method that can be applied to a wide class of linear plants.

8. NOMENCLATURE

BA: Bees Algorithm
DC: Direct Current
DCS: Distributed Control System
IAE: Integral Absolute Error
ISE: Integral Square Error
ITAE: Integral Time Absolute Error
ITSE: Integral Time Square Error
PID: Proportional, Integral and Derivative
MSE: Mean Square Error
MOBA: Multi-Objective Bees Algorithm
r : The reference input signal
e: The error signal
u : The control signal
y: The output signal
y_{max} : The maximum value of y
y_{ss} : The steady-state value of y
t_r: Rise time
t_s: Settling time
M_p: Maximum overshoot
J^B : The value of the objective function
Φ : Vector of non-negative weights
G_p: Linear Time-Invariant plant's transfer function
G_c : PID Controller transfer function
K_p: Proportional gain constant of PID controller
K_d : Derivative gain constant of PID controller
K_i : Integral gain constant of PID controller
n :Number of scout bees
m:Number of sites selected for neighborhood search
e_s :Number of top-rated (elite) sites among m selected sites
nep: Number of bees recruited for the selected sites
nsp: Number of bees recruited for the other $(m-e_s)$ selected sites
ngh: The initial size of each patch
sc : Shrinking constant
C_m : Motor torque
I_a : Rotor circuit current
K_e : Electrical constant
K_m : Mechanical constant
L_a : Rotor circuit inductance
R_a : Rotor circuit resistance
U_a : Input voltage
B : Friction ratio
e_v : Electromotive voltage
J : Rotor moment of inertia
Ω :Rotor speed

9. REFERENCES

1. Tan, K.K., Wang, Q.G., Hang, C.C., Hagglund, T.J., "Advances in PID control", Springer-Verlag, London, 1999.
2. Tan, G., Zeng, Q., He, S., Cai, G., "Adaptive and robust design for PID controller based on ant system algorithm", Lect. Notes Comput. Sci. 3612, pp. 915-924, 2005.
3. Ziegler, J.G., Nichols, N.B., "Optimum settings for automatic controllers", Trans. ASME 64, pp. 759-768, 1942.
4. Bagis, A., "Determination of the PID controller parameters by modified genetic algorithm for improved performance", J. Inf. Sci. Eng. 23, pp. 1469-1480, 2007.
5. Hsiao, Y.T., Chuang, C.L., Chien, C.C., "Ant colony optimization for designing of PID controllers", In: Proceedings of the 2004 IEEE international symposium on computer aided control systems design, Taipei, Taiwan, pp. 321-326, 2004.
6. Cohen, G.H., Coon, G.A., "Theoretical consideration of related control", Trans. ASME 75, pp. 827-834, 1953.
7. Pham, D.T., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., Zaidi, M., "The bees algorithm, a novel tool for complex optimization problems", In: Proceedings of the 2nd international virtual conference on intelligent

production machines and systems, Cardiff, UK, pp. 454-459, 2006.

8. Pham, D.T., Ghanbarzadeh, A., Koc, E., Otri, S., "Application of the bees algorithm to the training of radial basis function networks for control chart pattern recognition", In: Proceedins of the 5th CIRP international seminar on intelligent computation in manufacturing engineering, Ischia, Italy, pp. 711-716, 2006.

9. Pham, D.T., Otri, S., Ghanbarzadeh, A., Koc, E., "Application of the bees algorithm to the training of learning vector quantisation networks for control chart pattern recognition", In: Proceedings of information and communication technologies, Syria, pp. 1624-1629, 2006.

10. Pham, D.T., Koç, E., Ghanbarzadeh, A., Otri, S., "Optimization of the weights of multi-layered perceptrons using the bees algorithm", In: Proceedings of the 5th international symposium on intelligent manufacturing systems, Turkey, pp. 38-46, 2006.

11. Pham, D.T., Castellani, M., Ghanbarzadeh, A., "Preliminary design using the bees algorithm", In: Proceedings of the eighth international conference on laser metrology, CMM and machine tool performance, LAMDAMAP, Cardiff, UK, pp. 420-429, 2007.

12. Ong, C.M., "Dynamic simulation of electric machinery" Prentice-Hall Inc, New Jersey, pp. 558-568, 1998.

13. Tipsuwanporn, V., Piyarat, W., Tarasantisuk, C., "Identification and control of brushless DC motors using on-line trained artificial neural networks", In: Proceedings of the power conversion conference, Osaka, pp. 1290-1294, 2002.

14. Navidi, N., Bavafa, M., Hesami, S., "A new approach for designing of PID controller for a linear brushless DC motor with using ant colony search algorithm", In: Proceedings of the Asia-Pacific Power and energy engineering conference (APPEEC 2009), China, pp. 1-5, 2009.

15. Seborg, D.E., Edgar, T.F., Mellichamp, D.A., "Process dynamics and control", Wiley, New York, 2004.

16. Lin,M.G., Lakshminarayanan, S. Rangaiah, G.P., "A comparative study of recent/popular PID tuning rules for stable, first-order plus dead time, single-input single-output processes", Ind. Eng. Chem. Res. 47, pp. 344–368, 2008.

17. Ngatchou, P., Zarei, A., El-Sharkawi, M.A., "Pareto multi objective optimization", In: Proceedings of the 13th international conference on intelligent systems applications to power system, US, pp. 84-91, 2005.

18. Sayadi, F., Ismail, M., Misran, N., Jumari, K., "Multi-objective optimization using the bees Algorithm in time-varying channel for MIMO MC-CDMA systems", Eur. J. Sci. Res. 33, pp. 411-428, 2009.

19. Pham, D.T., Ghanbarzadeh, A., "Multi-objective optimization using the bees algorithm", In: proceedings of the 3rd international virtual conference on intelligent production machines and systems, Scotland, 2007.

20. Nasri, M., Nezanabadi-pour, H., Maghfoori, M., "A PSO-Based optimum design of PID controller for a linear brushless DC motor",. Proceedings of world academy of science, engineering and technology 26, pp. 211-215, 2007.