Journal of Advanced Research (2016) 7, 125–134

Cairo University

Journal of Advanced Research

ORIGINAL ARTICLE

A new multiobjective performance criterion used in PID tuning optimization algorithms

Mouayad A. Sahib *, **Bestoun S. Ahmed**

Software Engineering Department, College of Engineering, Salahaddin University-Hawler, Erbil, Iraq

ARTICLE INFO

Article history:
Received 14 January 2015
Received in revised form 13 March 2015
Accepted 27 March 2015
Available online 3 April 2015

Keywords:
Multiobjective optimization
Pareto set
PID controller
Particle Swarm Optimization (PSO)
AVR system

ABSTRACT

In PID controller design, an optimization algorithm is commonly employed to search for the optimal controller parameters. The optimization algorithm is based on a specific performance criterion which is defined by an objective or cost function. To this end, different objective functions have been proposed in the literature to optimize the response of the controlled system. These functions include numerous weighted time and frequency domain variables. However, for an optimum desired response it is difficult to select the appropriate objective function or identify the best weight values required to optimize the PID controller design. This paper presents a new time domain performance criterion based on the multiobjective Pareto front solutions. The proposed objective function is tested in the PID controller design for an automatic voltage regulator system (AVR) application using particle swarm optimization algorithm. Simulation results show that the proposed performance criterion can highly improve the PID tuning optimization in comparison with traditional objective functions.

Introduction

Proportional plus integral plus derivative (PID) controllers have been widely used as a method of control in many industrial applications. The robustness in performance and simplicity of structure are behind their domination among other controllers [1]. The design of the PID controller involves the determination of three parameters which are as follows: 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 [2] and Cohen and Coon [3]. In these methods, optimal PID parameters are often hard to determine [4]. For this reason, 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 [5,6], multiobjective optimization [7,8], evolutionary algorithm [9], Simulated Annealing (SA) [10], fuzzy systems [11], Artificial Bee Colony (ABC) [12,13], Genetic Algorithm (GA) [14], Particle Swarm Optimization (PSO) [15], Many Optimizing Liaisons (MOL) [16], and Tabu Search (TS) algorithm [17]. In all of the above optimization techniques, an objective or

* Corresponding author. Tel.: +964 7505352987.
E-mail address: mouayad.sahib@gmail.com (M.A. Sahib).

Peer review under responsibility of Cairo University.

ELSEVIER
Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.jare.2015.03.004
2090-1232 © 2015 Production and hosting by Elsevier B.V. on behalf of Cairo University.

cost function is defined to evaluate the performance of the PID controller.

In the literature, many objective functions have been proposed as a performance criterion [15,18–20]. The objective functions can be classified as a time or frequency domain based performance criterion. The most commonly used functions are the time domain integral error performance criteria which are based on calculating the error signal between the system output and the input reference signal [4]. The integral performance function types are integral of absolute error (IAE), integral of time multiplied by absolute error (ITAE), integral of squared error (ISE), integral of time multiplied by squared error (ITSE), and integral of squared time multiplied by squared error (ISTE) [21]. A more general form of the integral performance function with a fractional order of the time weight and absolute error has been proposed by Tavazoei [22]. A disadvantage of the IAE and ISE criteria is that they may result in a response with a relatively small overshoot but a long settling time because they weigh all errors uniformly over time [21]. The ITAE and ITSE performance criteria can overcome this drawback, but it cannot ensure to have a desirable stability margin [21]. A new performance criterion in the time domain has been proposed by Zwe-Lee in which the unit step timing parameters are used with a single weighting factor [15]. Zamani et al., proposed a general performance criterion to facilitate the control strategy over both the time and frequency domain specifications [18]. The objective function comprises eight terms including two frequency parameters. The significance of each term is determined by a weight factor. Evidences have showed that the proposed performance criterion can search efficiently for the optimal controller parameters. However, the choice of the weighting factors in the objective function is not an easy task [23].

This paper proposes a new time domain performance criterion based on the multiobjective Pareto solutions. The proposed objective function has the advantage of being simple such that it employs fewer terms. Moreover, it has the ability to guide the optimization search to a predefined design specifications indicated by an importance value. The proposed objective function is tested in the PID controller design for an automatic voltage regulator system (AVR) application using PSO algorithm.

Methodology

Performance evaluation criteria

The performance of the control system is usually evaluated based on its transient response behavior. This response is the reaction when subjecting a control system to inputs or disturbances [24]. The characteristics of the desired performance are usually specified in terms of time domain quantities. Commonly, unit step responses are used in the evaluation of the control system performance due to their ease of generation. In practical control systems, the transient response often exhibits damped oscillations before reaching steady state. There are many time domain parameters which are used to evaluate the unit step response. Such parameters are, the maximum overshoot M_p, the rise time t_r, the settling time t_s and the steady state error E_{ss} [24]. In the design of an efficient controller, the objective is to improve the unit step response by minimizing these time domain parameters. This objective can defiantly be achieved by minimizing the error between the unit step input signal and the unit step response. An example of a second order system unit step response is shown in Fig. 1.

As shown in Fig. 1, the transient response of the system can be described by two important factors; the swiftness of response and the closeness of the output to the reference (desired) input. The swiftness of response is characterized by the rise and peak times. However, the closeness of the output to the desired response is characterized by the maximum overshoot and settling time [25]. In general, the error signal is expressed as,

$$e(t) = u(t) - y(t) \tag{1}$$

In the literature, the error signal defined by Eq. (1) is widely used in the four performance criteria mentioned above. Those criteria are IAE, ITAE, ISE, and ITSE, and their formulas are as follows [21]:

$$\mathrm{IAE} = \int_0^{t_{ss}} |e(t)|dt \tag{2}$$

$$\mathrm{ISE} = \int_0^{t_{ss}} e^2(t)dt \tag{3}$$

$$\mathrm{ITAE} = \int_0^{t_{ss}} t|e(t)|dt \tag{4}$$

$$\mathrm{ITSE} = \int_0^{t_{ss}} te^2(t)dt \tag{5}$$

where t_{ss} is the time at which the response reaches steady state. The IAE and ISE weight all errors equally and independent of time. Consequently, optimizing the control system response using IAE and ISE can result in a response with relatively small overshoot but long settling time or vice versa [21]. To overcome this problem the ITAE and ITSE time weights the error such that late error values are considerably taken into account as shown in Fig. 2.

Although the ITAE and ITSE performance criteria can overcome the disadvantage of the IAE and ISE, the time weighted criteria can result in a multiple minimum optimization problem. In other words, two responses can have the same ITAE or ITSE values. In addition, the ITAE and ITSE

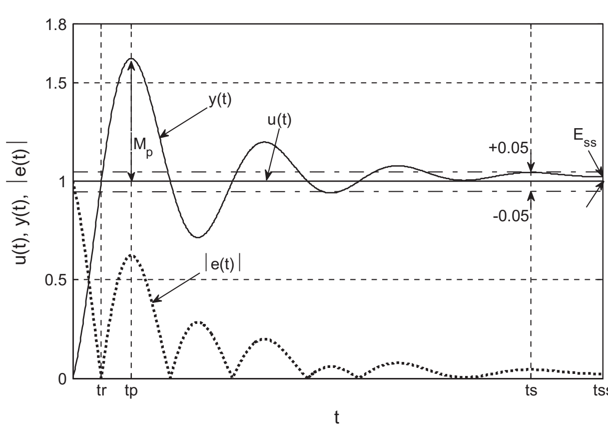

Fig. 1 Time domain parameters of the unit step response.

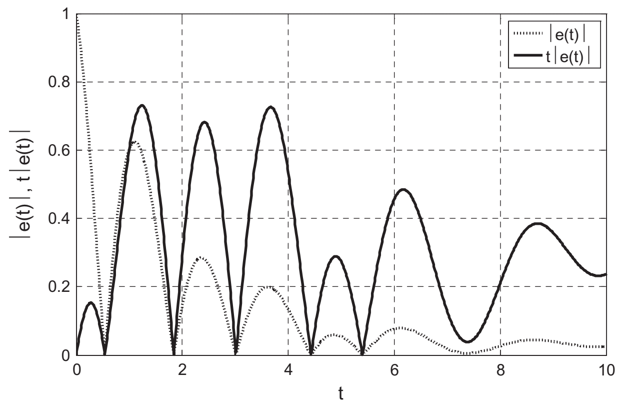

Fig. 2 Weighted and unweighted absolute error.

attempt to minimize the weighted absolute and squared error signals respectively. However, this does not necessarily mean minimizing all the basic evaluation parameters such as M_p, t_r, t_s, and E_{ss} at the same time. In addition to these parameters, the gain margin (GM) and phase margin (PM) which are used to determine the relative stability of the control system. Similarly, minimizing ITAE or ITSE does not necessarily mean minimizing the reciprocal of GM and PM. Therefore, a weighted sum of time and frequency domain parameters objective function has been proposed to overcome the multiminimum problem and improve the PID design process. For example, Zwe-Lee [15] proposed the performance criterion defined by minimizing,

$$J(K) = (1 - e^{-\beta})(M_p + E_{ss}) + e^{-\beta}(t_s - t_r) \tag{6}$$

where β is a weighting factor which can allow the designer to choose a specific requirements. To reduce the maximum overshoot and steady state error, β should be greater than 0.69. On the other hand, to reduce the time difference between settling and rise times, β should be less than 0.69. Another example, Zamani et al. [18] proposes a performance criterion defined by minimizing,

$$J(K) = w_1 M_p + w_2 t_r + w_3 t_s + w_4 E_{ss} + \int_0^{t_{ss}} (w_5 |e(t)| + w_6 u^2(t))dt + \frac{w_7}{PM} + \frac{w_8}{GM} \tag{7}$$

The objective function defined by Eq. (7) includes time domain parameters; overshoot M_p, rise time t_r, settling time t_s, steady state error E_{ss}, IAE, and integral of squared control signal and two frequency domain parameters; gain margin GM and phase margin PM. The significance of each parameter is determined by a weight factor w_i.

The choice of the weighting factors is not an easy task. The designer has to use multiple trials of weighting factors until the desired specifications can be attained. In addition, the variation range of each parameter is unknown, thus, its percentage of contribution in the overall fitness value is also unknown. For example, E_{ss} in Eq. (7) has a very small contribution value as compared to t_s or t_r. Therefore, the weight factor used for E_{ss} is usually set to a very large value as compared to the other parameters. In this paper, the proposed performance criterion evaluates the weighting factors according to their percentage of contribution in the fitness value. This will act as a calibration process and hence will identify a compromised state from which the designer can accurately apply the desired transient response specifications. The method of evaluating the weighting factors is based on the multiobjective Pareto front solutions and described in the following section.

Particle swarm optimization

Particle Swarm Optimization (PSO) is a well-known stochastic optimization technique which depends on social behavior. It uses the social behavior exploiting the solution space to determine the best value in this space [26]. In contrast to Genetic algorithm, PSO does not use operators inspired by natural evolution which are incorporated to form a new generation of candidate solutions [4]. GA mutation operation is replaced in PSO by the exchange of information between individuals, called particles, of the population which in PSO is called swarm. In effect, the particle adjusts its trajectory toward its own previous best position, and toward the global best previous position obtained by any member of its neighborhood. In the global variant of PSO, the swarm is considered as the neighborhood, in other words, all the particles are considered as a neighborhood for the individual particle. Therefore, the sharing of information takes place and the particles benefit from the exploiting process and experience of all other particles during the search for promising regions of the landscape [26].

There were various enhancement and techniques applied to PSO since the emergence of PSO by Kennedy and Eberhart for obtaining the best possible behavior related to various types of problems [27]. However, the general structure for the PSO remained the same. To understand the mathematical formation of PSO, consider a search space of N-Dimension, the i^{th} particle is represented by $X_i = [x_{i1}, x_{i2}, \ldots, x_{iN}]$ and the best particle with the best solution is denoted by the index g. The best previous position of the i-th particle is denoted by $P_i = [p_{i1}, p_{i2}, \ldots, p_{iN}]$ and the velocity (position change) is denoted by $V_i = [v_{i1}, v_{i2}, \ldots, v_{iN}]$. The particle position will be updated in each iteration of the algorithm according to the following equation:

$$V_i^{k+1} = wV_i^{k+1} + c_1 r_{i1}^k \left(P_i^k - X_i^k\right) + c_2 r_{i2}^k \left(P_g^k - X_i^k\right) \tag{8}$$

and,

$$X_i^{k+1} = X_i^k + V_i^{k+1} \tag{9}$$

where $i = 1, 2, \ldots, M$, and M is the number of population (swarm size); w is the inertia weight, c_1 and c_2 are two positive constants, called the cognitive and social parameter respectively; r_{i1} and r_{i2} are random numbers uniformly distributed within the range [0; 1]. Eq. (8) above is used to find the new velocity for the i-th particle, while Eq. (9) is used to update the i-th position by adding the new velocity obtained by Eq. (8). The behavior of each particle in the swarm is controlled by the above equation and it is subject to a function which is called fitness or objective function. The objective function determines how far or near each individual particle with respect to the optimal solution. Thus, each particle movement will be updated to get as close as possible to satisfy the objective function. The pseudocode of the PSO algorithm is presented in Fig. 3.

```
Procedure PSO
 Initialize particles population
do
      for each particle p with position x_p do
            calculate fitness value f(x_p)
            if f(x_p) is better than pbest_p then
                  pbest_p ← x_p
            endif
      endfor

      Define gbest_p as the best position found so far by any of
      p's neighbors

      for each particle p do
            v_p ← compute_velocity(x_p, pbest_p, gbest_p)
            x_p ← update_ position(x_p, v_p)
      endfor

while (Max iteration is not reached or a stop criterion is
       not satisfied)
```

Fig. 3 The pseudocode of the PSO algorithm.

At each iteration, the PSO algorithm relies on the objective function in evaluating the effectiveness of each particle as well as in calculating the current particle's velocity. Therefore, the choice of the objective function which represents the performance criterion plays an important role in the search process of the optimization algorithm.

The proposed approach

Multiobjective optimization is a multicriteria decision making problem which involves two or more conflicting objective functions to be minimized simultaneously. Multiple criteria or Multiobjective (MO) optimization has been applied in various fields where multiple objective functions are required to be optimized concurrently [28]. The main difference between single objective and MO optimization problems is that in the former the end result is a single "best solution" while in the latter is a set of alternative solutions. Each member of the alternative solutionset represents the best possible trade-offs among the objective functions. The set of all alternative solutions is called Pareto optimal set (PO) and the graph of the PO set is called Pareto front [7]. The notion of Paretooptimality is only a first step toward solving a multiobjective problem. In order to select an appropriate compromise solution from the Pareto optimal set, a decision making (DM) process is necessary [29]. In the search for compromised solutions, one of the broad classes of multiobjective methods is *priori articulation of preferences* [30]. In this method, the decision maker expresses preferences in terms of an aggregating function. The aggregated function is a single objective problem which combines individual objective values, such as M_p, t_r, and t_s, into a single utility value. The single utility function can discriminate between candidate solutions using weighting coefficients. These weightings are real values used to express the relative importance of the objectives and control their involvement in the overall utility measure [30].

In the PID tuning optimization problem the objective is to solve the following problem [31]:

$$\text{Minimize}: \quad \vec{f}(\vec{k}) = [f_1(\vec{k}), f_2(\vec{k}), \ldots, f_j(\vec{k})] \tag{10}$$

subject to the constraint functions,

$$g_i(\vec{k}) \leqslant 0 \quad i = 1, 2, \ldots m \tag{11}$$

$$h_i(\vec{k}) = 0 \quad i = 1, 2, \ldots p \tag{12}$$

where $\vec{k} = [K_p, K_i, K_d]$ is the vector of PID gain parameters, $f_i(\vec{k}) : \mathbb{R}^3 \to \mathbb{R}, i = 1, 2, \ldots j$ are the objective functions, and $g_i(\vec{k}), h_i(\vec{k}) : \mathbb{R}^3 \to \mathbb{R}, i = 1, 2, \ldots m, i = 1, 2, \ldots p$ are the constraint functions. A solution vector of PID gain parameters, $\vec{k}_u \in \mathbb{R}^3$, is said to dominate $\vec{k}_v \in \mathbb{R}^3$ (denoted by $\vec{k}_u \preceq \vec{k}_v$) if and only if $\forall i \in \{1, \ldots, j\}$ we have $f_i(\vec{k}_u) \leqslant f_i(\vec{k}_v)$ and $\exists i \in \{1, \ldots, j\} : f_i(\vec{k}_u) < f_i(\vec{k}_v)$. A feasible solution, $\vec{k}^* \in \mathbb{R}^3$, is called Pareto optimal if and only if there is no other solution, $\vec{k} \in \mathbb{R}^3$, such that $\vec{k} \preceq \vec{k}^*$. The set of all Pareto optimal solutions is called Pareto optimal set and denoted by $\mathcal{P} = \{\vec{k}_{p1}, \vec{k}_{p2}, \ldots, \vec{k}_{pl}\}$. Given $\mathcal{P}$ for a MO optimization problem defined by $\vec{f}(\vec{k})$, the Pareto front is given by:

$$\mathcal{PF} = \begin{Bmatrix} f_1(\vec{k}_{p1}), & f_2(\vec{k}_{p1}), & \ldots, & f_j(\vec{k}_{p1}) \\ f_1(\vec{k}_{p2}), & f_2(\vec{k}_{p2}), & \ldots, & f_j(\vec{k}_{p2}) \\ \vdots & \vdots & & \vdots \\ f_1(\vec{k}_{pl}), & f_2(\vec{k}_{pl}), & \ldots, & f_j(\vec{k}_{pl}) \end{Bmatrix} \tag{13}$$

The main objective functions in PID design problem are the maximum overshoot M_p, the rise time t_r, the settling time t_s and the steady state error E_{ss}. When using an optimization algorithm to find the PID gain parameters, such as the PSO algorithm, these objective functions are combined in a single weighted sum objective function defined by,

$$J(\vec{k}) = \sum_{i=1}^{j} w_i f_i(\vec{k}), \quad \text{with} \quad \sum_{i=1}^{j} w_i = 1 \tag{14}$$

The method of converting MO problem to a single weighted objective is commonly used in the application of PID controller optimization due to its simplicity. However, there are several drawbacks associated with this method. Such drawbacks are related to the choice of the weights which is a matter of trial and error [23]. In addition, the optimization search will be restricted and limited to the selected weighting factor set. Furthermore, enforcing the main objective function to have a uniform contribution of terms can be achieved by two conditions. Firstly, the terms are equally weighted, and secondly, the terms have equal standard deviation (σ) in $\mathbb{R}$. Otherwise, the terms will have a nonuniform contribution. For PID tuning application, the terms of the objective function, such as Eq. (7), usually have different standard deviations. For example, the standard deviation of E_{ss} is much less than that of t_s, i.e., $\sigma_{E_{ss}} \ll \sigma_{t_s}$. Thus, in order to compensate for this difference, the weight factor given for the E_{ss} term should be much greater than that given to the t_s term ($w_{E_{ss}} \gg w_{t_s}$). In general, for a given term, $f_i(\vec{k})$, with a standard deviation, σ_i, the corresponding contribution percentage $CP[f_i(\vec{k})]$ can be calculated using,

$$CP[f_i(\vec{k})] = \frac{\mu_i}{\sum_{n=1}^{j} \mu_n} * 100\% \tag{15}$$

where μ_i is the mean value of all the Pareto solutions (column i in $\mathcal{PF}$) corresponding to $f_i(\vec{k}_{pn})$ for $n = 1, 2, \ldots, l$, i.e.,

$$\mu_i = \frac{1}{l}\sum_{n=1}^{l} f_i(\vec{k}_{pn}) \tag{16}$$

The weighting factors are inversely proportional to the contribution percentage and are given by:

$$w_i = \frac{1}{CP[f_i(\vec{k})] * \sum_{n=1}^{j} \frac{1}{CP[f_n(\vec{k})]}} \tag{17}$$

Substituting Eq. (15) in (17) yields,

$$w_i = \frac{1}{\mu_i * \sum_{n=1}^{j} \frac{1}{\mu_n}} \tag{18}$$

Substituting Eq. (18) in (14), yields to the proposed objective function:

$$J(\vec{k}) = \sum_{i=1}^{j}\left[\frac{f_i(\vec{k})}{\mu_i * \sum_{n=1}^{j} \frac{1}{\mu_n}}\right] \tag{19}$$

The proposed objective function given by Eq. (19), can statistically ensure an equivalent contribution of the MO terms. Therefore, an optimization algorithm, like PSO, that employs the proposed objective function, is expected to produce optimized Pareto solutions. The Pareto solutions can have Pareto front values with standard deviations approximately equal to that used in deriving the proposed objective function. The proposed performance criterion can be improved by using additional weights, called importance weights, w_{ci}. The new w_{ci} weights, define the importance of each term such that the larger the weight value, the higher the importance of the objective term. Therefore, the proposed objective function given by Eq. (19) can be modified to,

$$J(\vec{k}) = \sum_{i=1}^{j} w_{ci}[w_i f_i(\vec{k})] = \sum_{i=1}^{j} w_{ci}\left[\frac{f_i(\vec{k})}{\mu_i * \sum_{n=1}^{j} \frac{1}{\mu_n}}\right] \quad \text{with} \quad \sum_{i=1}^{j} w_{ci} = 1 \tag{20}$$

In Eq. (20), w_i weights are responsible for maintaining equivalent contribution value of all the objective terms. However, w_{ci} weights are used to control the importance of each objective term. Based on this proposed performance criterion, a compromised solution can be obtained if appropriate weights are used to compensate for the different deviation ranges and when using equal importance weights.

Results and discussion

In this section, the proposed performance criterion is evaluated with PSO algorithm. The PSO algorithm is employed in the application of designing a PID controller for real practical application system represented by an automatic voltage regulator (AVR). The PID controller transfer function is

$$C_{PID} = C_{PID} = K_p + \frac{K_i}{s} + K_d s \tag{21}$$

where K_p, K_i, and K_d are the proportional, integral, and derivative gains. The transfer function of the AVR system without PID controller was previously reported [15,16,32]:

$$\frac{\Delta V_t(s)}{\Delta V_{ref}(s)} = \frac{0.1s + 10}{0.0004s^4 + 0.045s^3 + 0.555s^2 + 1.51s + 11} \tag{22}$$

where $V_t(s)$ and $V_{ref}(s)$ are the terminal and reference voltages. The unit step response of the AVR system without PID controller is shown in Fig. 4.

It can be observed from Fig. 4 that the AVR system possess an underdamped response with steady state amplitude value of 0.909, peak amplitude of 1.5 (M_p = 65.43%) at t_p = 0.75, t_r = 0.42 s, t_s = 6.97 s at which the response has settled to 98% of the steady state value. To improve the dynamics response of the AVR system a PID controller is designed. The gain parameters of the PID controller are optimized using PSO algorithm. The searching range of positions (gain parameters) and velocities is defined in Table 1.

The PID tuning optimization problem is defined by three objective functions:

$$\text{Minimize}: \quad \vec{f}(\vec{k}) = [f_1(\vec{k}) = M_p(\vec{k}),\ f_2(\vec{k}) = t_r(\vec{k}),\ f_3(\vec{k}) = t_s(\vec{k})] \tag{23}$$

subject to the constraint function,

$$M_p(\vec{k}) + t_r(\vec{k}) + t_s(\vec{k}) \leqslant b \tag{24}$$

Some sets of the PID gain parameters result in a step response of the controlled AVR system with large values of M_p, t_r, and/or t_s. Therefore, the constraint defined by Eq. (24) is used to limit the results to include only those with $M_p(\vec{k}) + t_r(\vec{k}) + t_s(\vec{k}) \leqslant b$, where b is a predefined constant and set to be 5.

A discrete form of the Pareto front for the MO problem defined in (17), can be found by considering all the combinations of the gain parameters with a step size equal to 0.005. Fig. 5 depicts the Pareto front ($\mathcal{PF}$) values of the three objective functions with their corresponding Pareto optimal solutions ($\mathcal{P}$).

From Fig. 5, it is clear that among all the combinations, 28 Pareto front sets were obtained. The corresponding nondominated Pareto optimal solutions are also shown. From the Pareto front sets, the mean values μ_{M_p}, μ_{t_r}, and μ_{t_s} are calculated using Eq. (13) to be 0.178, 0.184, and 0.730 respectively. The MO problem defined by the three objectives (maximum overshoot, rise time, and settling time) can be combined in a single weighted sum function given by:

$$J(\vec{k}) = w_{M_p} M_p(\vec{k}) + w_{t_r} t_r(\vec{k}) + w_{t_s} t_s(\vec{k}) \tag{25}$$

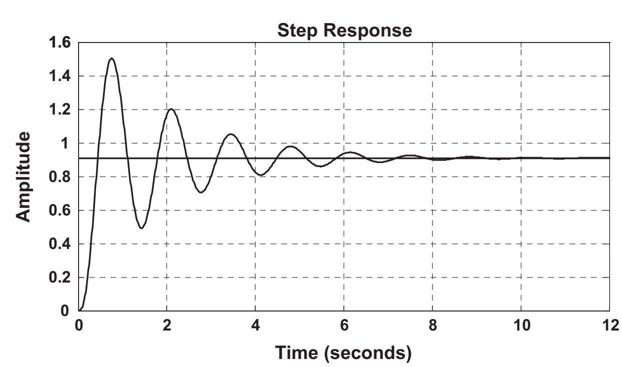

Fig. 4 Step response of the AVR system without PID controller.

Table 1 Searching range of parameters.

Parameter	Min. value	Max. value
K_p	0.0001	1.5
K_i	0.0001	1.0
K_d	0.0001	1.0
v_{K_p}	−0.75	0.75
v_{K_i}	−0.5	0.5
v_{K_d}	−0.5	0.5

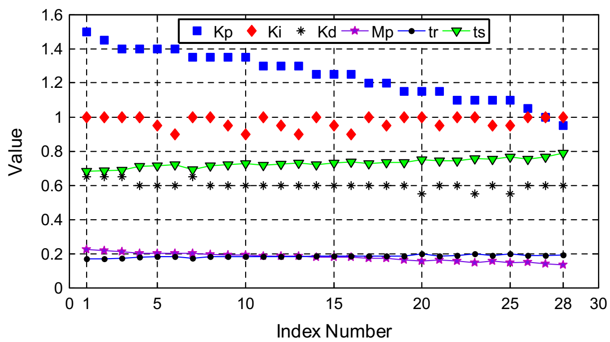

Fig. 5 Pareto front and Pareto optimal solution sets.

When combining the three objectives in a single weighted sum function the contribution of the objectives is related to their mean values. The mean values indicate that the contribution of the settling time is much greater than that of the rise time and maximum overshoot. The percentage of contribution of the $M_p(\vec{k})$, $t_r(\vec{k})$, and $t_s(\vec{k})$ objectives are 16.3%, 16.9%, and 66.8% respectively. To ensure an equivalent contribution of the three terms, the weights in Eq. (25) are calculated using Eq. (16), with $j = 3$, to be $w_{M_p} = 0.452$, $w_{t_r} = 0.438$, and $w_{t_s} = 0.110$.

In optimizing the PID gains, the PSO algorithm employs the proposed objective function defined in Eq. (3). The simulation parameters of the PSO algorithm are listed in Table 2.

Setting the number of iterations (N) to 50 in the PSO algorithm is adequate to prompt convergence and obtain good results. This was shown by Zwe-Lee Gaing in the convergence tendency of the PSO-PID controller used to control the same AVR system [15]. In PSO algorithm, initial population is commonly generated randomly hence different final solutions may be achieved. Thus, if only one trial is conducted, the result may or may not be an optimal solution. Therefore, to solve such problem, several trials are carried out, and then the optimal solution among all trials is reported. Here, the PSO algorithm is repeated 10 times (number of trials (T) = 10) and then the optimum PID controller gains corresponding to the minimum fitness value is considered. Based on some empirical study of PSO performed by Shi and Eberhart using various population sizes (20, 40, 80 and 160), it has been shown that the PSO has the ability to quickly converge and is not sensitive when increasing the population size (swarm size) above 20 [33]. Therefore in this paper the swarm size is set to $L = 30$. The constants c_1 and c_2 represent the weighting of the stochastic acceleration terms that pull each particle toward pbest and gbest positions. Low values allow particles to fly far from the target regions before being tugged back. On the other hand, high values result in abrupt movement toward, or past, target regions. Hence, the acceleration constants c_1 and c_2 were often set to be 2.0 according to past experiences [15]. The inertia weight (w) provides a balance between global and local explorations, thus requiring less iteration on average to find a sufficiently optimal solution. As originally developed, w often decreases linearly from 0.9 to 0.4 with a step size equal to the difference between the upper (0.9) and lower (0.4) limits divided by N (50), i.e., step size = 0.014 [15].

Table 2 PSO searching parameters.

Parameter	Value
Number of iterations (N)	50
Number of trials (T)	10
Swarm size (L)	30
Constants ($c_1 = c_2$)	2
Inertia weight factor (w)	[0.9:0.014:0.2]

It is worth noting that the fully connected neighborhood topology (gbest version) is used in the PSO algorithm. In this topology all particles are directly connected among each other, as a result, the PSO tends to converge more rapidly to the optimal solution [34].

Fig. 6 shows the step response of the AVR system with PID controller optimized using the PSO algorithm and the proposed objective function.

The response of the AVR system with PID controller shown in Fig. 6, exhibits $M_p = 12\%$ at $t_p = 0.28$ s, $t_r = 0.14$ s, and $t_s = 0.78$ s. These values are comparable to the corresponding mean values of the Pareto front sets shown in Fig. 5. This confirms the ability of the proposed objective function in producing optimized and compromised Pareto solution. Fig. 7 shows the result of 10 trials when using the proposed objective function with PSO.

As shown in Fig. 7, for all trials, the values of K_p, K_i, and K_d are constantly equal to 0.937, 1, and 0.558 respectively. Similarly, the values of M_p, t_r, and t_s are 0.120, 0.136, and 0.788 respectively. Therefore, the proposed function can always guide the PSO algorithm to produce a compromised nondominated Pareto solution.

With a PID controller designed using the PSO algorithm, the response of the AVR system has been improved. However, the improvement is a compromise between maximum overshoot, rise time, and settling time. Steering the optimization search to a desired response can be achieved by

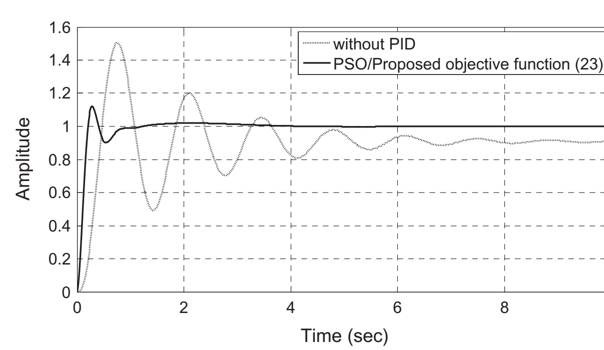

Fig. 6 AVR system response with optimized PID controller using PSO.

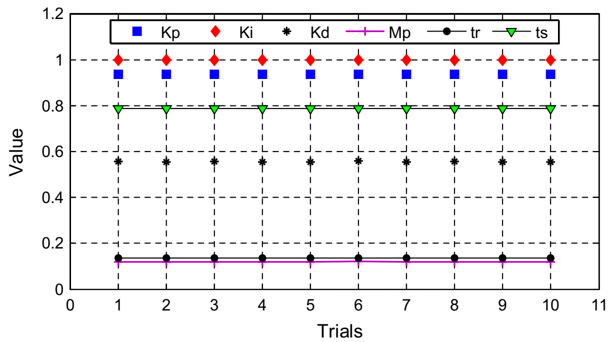

Fig. 7 Results of 10 PSO trials with the proposed objective function.

increasing the significance of the corresponding objective. Therefore, in addition to the compensation weights, the importance weights are used in the proposed objective function as in Eq. (18). In this context, three cases related to w_{cM_p}, w_{ct_r}, and w_{ct_s} are carried out for simulation. With each case the value of one importance weight varies from 0 to 0.9 with a step equal to 0.1 and the other two corresponding weights are set to have equal values satisfying the condition in Eq. (18), i.e., in case I, for each value of w_{cM_p} from 0 to 0.9, the values of w_{ct_r} and w_{ct_s} are,

$$w_{ct_r} = w_{ct_s} = (1 - w_{cM_p})/2 \tag{26}$$

Fig. 8 shows the result of the PSO algorithm when using the proposed objective function for the three cases, I, II, and III, related to the importance weights w_{cM_p}, w_{ct_r}, and w_{ct_s} respectively.

It can be observed from Fig. 8 that as the importance weight increases, the effect of optimizing (minimizing) the corresponding objective will also increase versus a decrease effect of optimizing the other two objectives. For example, in Fig. 8(a), as w_{cM_p} increase, M_p decrease, and t_r increase. Approximately, in all cases, an equivalent importance state can appear at an importance weight value equal to 0.3 and the other importance weights equal to 0.35 each. At the equivalent importance state, the values of M_p, t_r, t_s, K_p, K_i, and K_d are almost equal to those obtained without using the importance weights in the proposed objective function (i.e., almost equal to the values observed from Fig. 7). Table 3 lists the equivalent importance state results.

The proposed objective function given by Eq. (18) and some literature performance criteria is also presented in this section. Fig. 9(a) shows a comparison between the terminal voltage step responses with PID controller optimized using the proposed objective function and five literature performance criteria defined by Eqs. (2)–(7). Fig. 9(b) shows the controller signal output of each corresponding response presented in Fig. 9(a). In Eq. (6), β is chosen to be 1 [15]. Equating β to 1, is equivalent to weighting the $(M_p + E_{ss})$ term with an importance value equal to 0.632. As a result the $(t_s - t_r)$ term will have an importance value equal to 0.368. Therefore, the importance weights of the proposed objective function, w_{cM_p}, w_{ct_r}, and w_{ct_s} are set to 0.632, 0.184, and 0.184 respectively. In Eq. (7), w_1, w_2, w_3, and w_4 are set to be 0.1, 1, 1, and 1000 respectively [18].

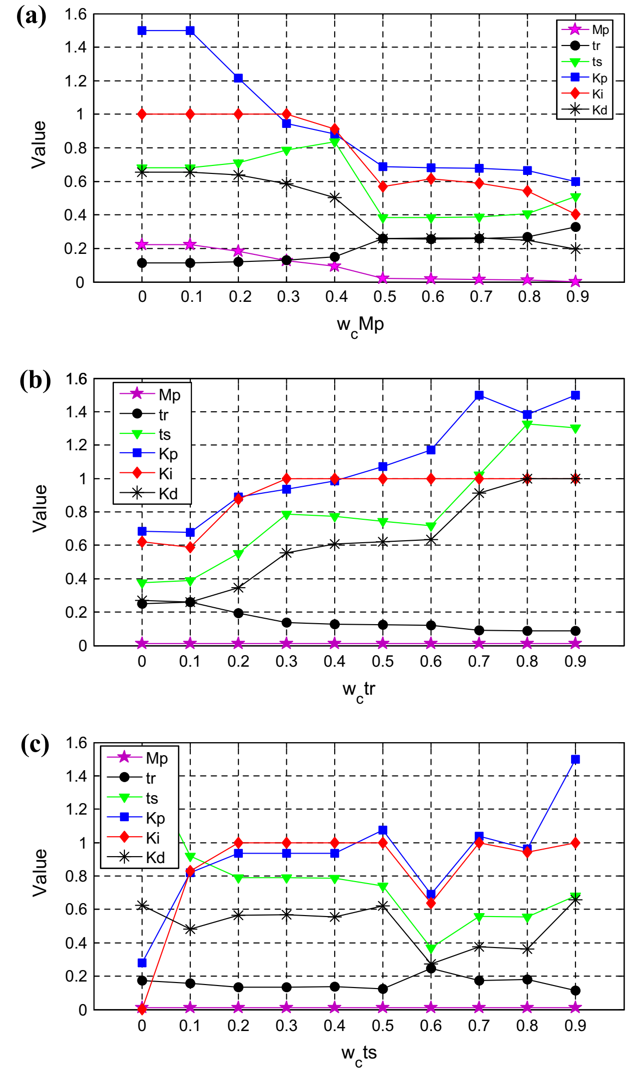

Fig. 8 Results of PSO trials with various values of (a) w_{cM_p}, (b) of w_{ct_r} and (c) w_{ct_s}.

Table 3 Equivalent importance state results.

Mean	Parameter	Case I	Case II	Case III
$\mu_{M_p} = 0.178$	$M_p = 0.120$	0.129	0.112	0.112
$\mu_{t_r} = 0.184$	$t_r = 0.136$	0.131	0.137	0.135
$\mu_{t_s} = 0.730$	$t_s = 0.788$	0.787	0.788	0.789
$\mu_{K_p} = 1.244$	$K_p = 0.937$	0.946	0.937	0.935
$\mu_{K_i} = 0.971$	$K_i = 1.000$	1.000	1.000	1.000
$\mu_{K_d} = 0.602$	$K_d = 0.558$	0.585	0.554	0.566

As can be seen from Fig. 9(a), the response of the proposed performance criterion case is comparable to the case of Eq. (6). In Fig. 9(b), the PID controller output can be obtained by filtering the ideal derivative action given by (21) using a first-order filter, i.e.,

$$C_{PID_f} = K_p + \frac{K_i}{s} + \frac{sK_d}{T_f s + 1} \tag{27}$$

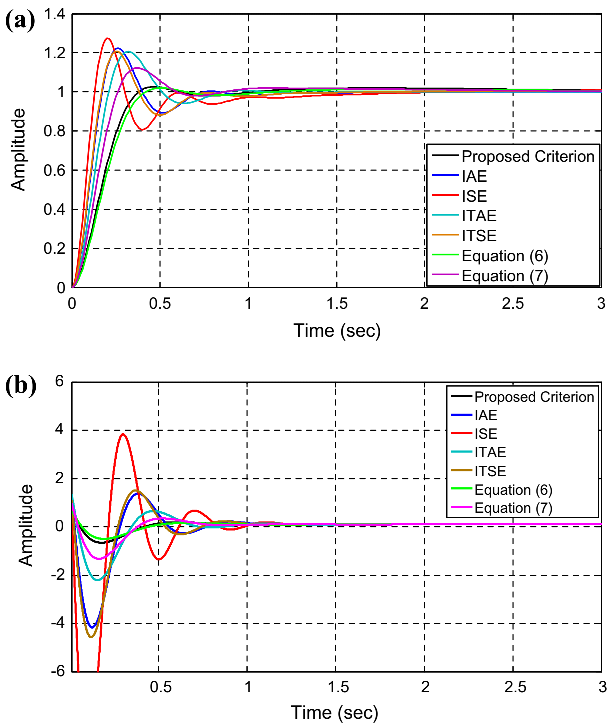

Fig. 9 AVR system controlled with optimized PID using different objective functions (a) unit-step response and (b) controller signal output.

Table 4 Step response results for various objective functions.

Case	M_p%	t_r	t_s	K_p	K_i	K_d
Prop. Criterion	**02.60**	**0.240**	**0.520**	**0.708**	**0.656**	**0.282**
IAE	22.24	0.116	0.686	1.500	1.000	0.642
ISE	27.28	0.087	1.361	1.239	1.000	1.000
ITAE	20.52	0.141	0.784	1.453	1.000	0.466
ITSE	20.75	0.114	1.048	1.348	1.000	0.675
Eq. (6)	02.00	0.260	0.510	0.686	0.571	0.255
Eq. (7)	12.23	0.175	0.556	1.031	1.000	0.375

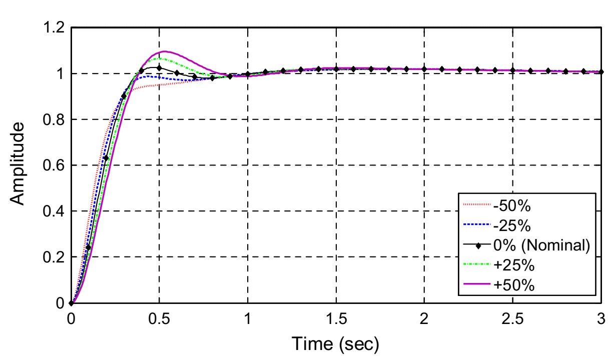

Fig. 10 Step response curves ranging from −50% to +50% for T_a.

where T_f is the time constant of the first-order filter. As T_f approaches zero, C_{PID_f} will be equivalent to the ideal PID (C_{PID}). Therefore, the time constant T_f is set to a very small

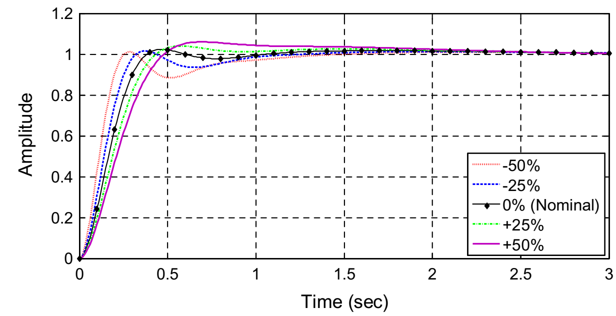

Fig. 11 Step response curves ranging from −50% to +50% for T_e.

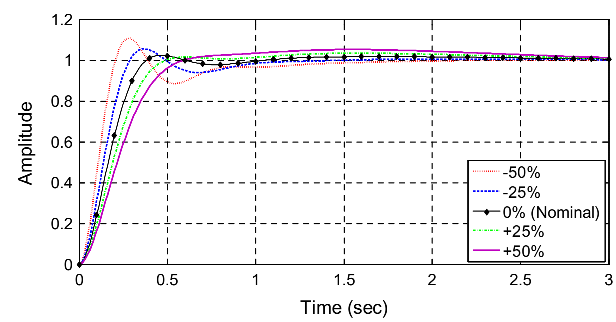

Fig. 12 Step response curves ranging from −50% to +50% for T_g.

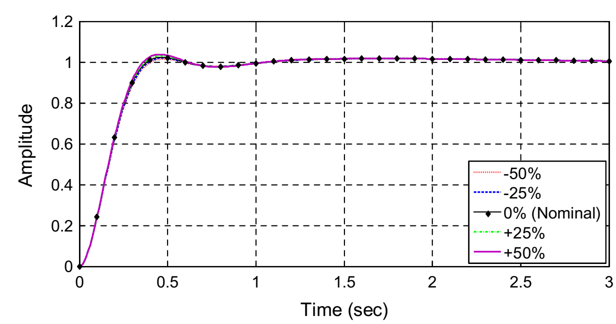

Fig. 13 Step response curves ranging from −50% to +50% for T_s.

value ($T_f = 0.001$) to make the PID controller output signal (with filtered derivative action) resembles the ideal PID output. It can be observed from Fig. 9(b) that the output of the PID controllers almost agrees with their corresponding step responses. Also, the outputs of the proposed PID and that of Eq. (6) are almost comparable and are the best among other outputs. This is evident as they require less demanding control signal. The values of M_p, t_r, t_s, K_p, K_i, and K_d for each case are listed in Table 4.

It is clear from Table 4 that the results of the proposed objective function along with its weights, highlighted in bold, are comparable to the case of Eq. (6). However, the proposed function uses only three time domain features. In addition the weights used in the proposed objective function are derived statistically, while the weighting factor β was found heuristically.

Table 5 Robustness analysis results of the AVR system with the proposed PID controller.

Parameter	Rate of change (%)	Peak value (pu)	t_s	t_r	t_p
T_a	−50	1.0183	0.8138	0.2580	1.8266
	−25	1.0188	0.8101	0.2382	1.7804
	+25	1.0640	1.7411	0.2473	0.4966
	+50	1.0950	1.8517	0.2562	0.5368
T_e	−50	1.0145	1.0929	0.1565	0.2825
	−25	1.0187	0.9293	0.1200	0.3688
	+25	1.0428	2.1156	0.2773	0.5691
	+50	1.0625	2.2325	0.3119	0.6894
T_g	−50	1.1092	1.2600	0.1374	0.2776
	−25	1.0569	0.9400	0.1878	0.3697
	+25	1.0361	2.5080	0.2939	1.5714
	+50	1.0544	2.7872	0.3488	1.5415
T_s	−50	1.0193	0.3712	0.2476	1.7403
	−25	1.0224	0.5050	0.2436	0.4701
	+25	1.0338	0.8254	0.2363	0.4589
	+50	1.0401	0.8408	0.2329	0.4586

Table 6 Total deviation ranges and maximum deviation percentage of the system.

Parameter	Total deviation range/max deviation percentage (%)			
	Peak value (pu) **1.0260**	t_s **0.5202**	t_r **0.2401**	t_p **0.4636**
T_a	0.0767/7%	1.0416/256%	0.0198/8%	1.3300/294%
T_e	0.0480/4%	1.3032/329%	0.1920/50%	0.4069/49%
T_g	0.0731/8%	1.8472/536%	0.2114/45%	1.2938/239%
T_s	0.0208/1%	0.4696/62%	0.0147/3%	1.2817/275%
Average	**0.0547/5%**	**1.1654/296%**	**0.1095/27%**	**1.0781/214%**

The robustness of the proposed controller is also investigated by changing the time constants (T_a, T_e, T_g, and T_s) of the four AVR system components separately [32]. The range of change is selected to be ±50% of the nominal time constant values with a step size of 25%. The robustness step response curves are presented in Figs. 10–13 for changing the time constants T_a, T_e, T_g, and T_s respectively. In addition, the response time parameters and the percentage values of maximum deviations are also listed in Tables 5 and 6 respectively. In Table 6, the average values of the deviation ranges and the maximum deviation percentage of the system are highlighted in bold.

It can be observed from Figs. 10–13 that the deviations of response curves (±50% and ±25%) from the nominal response for the selected time constant parameters are within a small range. The average deviation of maximum overshoot, settling time, rise time and peak time are 5%, 296%, 27% and 214% respectively. The ranges of total deviation are acceptable and are within limit. Therefore, it can be concluded that the AVR system with the proposed PID controller is robust.

Conclusions

In this paper, a new time domain performance criterion based on the multiobjective Pareto front solutions is proposed. The proposed objective function employs two types of weights. The first type, termed contribution weights, is responsible for maintaining equivalent contribution value of all the objective terms. However, the second type, termed importance weights, is used to control the importance of each objective term. The contribution weights are derived statistically from the Pareto front set which is obtained using the nondominated PID solution gain parameters. The importance weights can be selected according to the design specifications indicated by an importance value. The proposed criterion has been tested in the PSO algorithm used for the application of designing an optimal PID controller for an AVR system. In addition, the results are compared with some commonly used performance evaluation criteria such as IAE, ISE, ITAE, and ITSE. Simulation results show that the proposed performance criterion can highly improve the PID tuning optimization in comparison with traditional objective functions.

Conflict of interest

The authors have declared no conflict of interests.

Compliance with Ethics Requirements

This article does not contain any studies with human or animal subjects.

References

[1] Ghosh BK, Zhenyu Y, Xiao Ning Di, Tzyh-Jong T. Complementary sensor fusion in robotic manipulation. In: Ghosh BK, Ning Xi, Tarn TJ, editors. Control in robotics and automation. San Diego: Academic Press; 1999. p. 147–82 [chapter 5].

[2] Ziegler JG, Nichols NB. Optimum settings for automatic controllers. J Dyn Syst Meas Contr 1993;115(2B):220–2.

[3] Cohen G, Coon G. Theoretical consideration of retarded control. Trans Am Soc Mech Eng (ASME) 1953;75(1):827–34.

[4] Bansal HO, Sharma R, Shreeraman P. PID controller tuning techniques: a review. J Control Eng Technol 2012;2(4):168–76.

[5] Panda S. Differential evolution algorithm for SSSC-based damping controller design considering time delay. J Franklin Inst 2011;348(8):1903–26.

[6] Mohamed AW, Sabry HZ, Khorshid M. An alternative differential evolution algorithm for global optimization. J Adv Res 2012;3(2):149–65.

[7] Coello CAC, Pulido GT, Lechuga MS. Handling multiple objectives with particle swarm optimization. IEEE Trans Evol Comput 2004;8(3):256–79.

[8] Adly AA, Abd-El-Hafiz SK. A performance-oriented power transformer design methodology using multi-objective evolutionary optimization. J Adv Res 2014.

[9] Panda S. Multi-objective evolutionary algorithm for SSSC-based controller design. Electr Power Syst Res 2009;79(6):937–44.

[10] Ho S-J, Shu L-S, Ho S-Y. Optimizing fuzzy neural networks for tuning PID controllers using an orthogonal simulated annealing algorithm OSA. IEEE Trans Fuzzy Syst 2006;14(3):421–34.

[11] Mukherjee V, Ghoshal SP. Intelligent particle swarm optimized fuzzy PID controller for AVR system. Electr Power Syst Res 2007;77(12):1689–98.

[12] Gozde H, Taplamacioglu MC. Comparative performance analysis of artificial bee colony algorithm for automatic voltage regulator (AVR) system. J Franklin Inst 2011;348(8):1927–46.

[13] Mohamed AF, Elarini MM, Othman AM. A new technique based on artificial bee colony algorithm for optimal sizing of stand-alone photovoltaic system. J Adv Res 2014;5(3):397–408.

[14] Bindu R, Namboothiripad MK. Tuning of PID controller for DC servo motor using genetic algorithm. Int J Emerg Technol Adv Eng 2012;2(3):310–4.

[15] Zwe-Lee G. A particle swarm optimization approach for optimum design of PID controller in AVR system. IEEE Trans Energy Convers 2004;19(2):384–91.

[16] Panda S, Sahu BK, Mohanty PK. Design and performance analysis of PID controller for an automatic voltage regulator system using simplified particle swarm optimization. J Franklin Inst 2012;349(8):2609–25.

[17] Bagis A. Tabu search algorithm based PID controller tuning for desired system specifications. J Franklin Inst 2011;348(10):2795–812.

[18] Zamani M, Karimi-Ghartemani M, Sadati N, Parniani M. Design of a fractional order PID controller for an AVR using particle swarm optimization. Control Eng Pract 2009;17(12):1380–7.

[19] Sahu BK, Mohanty PK, Panda S, Mishra N, editors. Robust analysis and design of PID controlled AVR system using Pattern Search algorithm. IEEE international conference on power electronics, drives and energy systems (PEDES), 2012.

[20] Rahimian MS, Raahemifar K, editors. Optimal PID controller design for AVR system using particle swarm optimization algorithm. 24th Canadian conference on electrical and computer engineering (CCECE), 2011.

[21] Krohling RA, Rey JP. Design of optimal disturbance rejection PID controllers using genetic algorithms. IEEE Trans Evol Comput 2001;5(1):78–82.

[22] Tavazoei MS. Notes on integral performance indices in fractional-order control systems. J Process Control 2010;20(3):285–91.

[23] Aguila-Camacho N, Duarte-Mermoud MA. Fractional adaptive control for an automatic voltage regulator. ISA Trans 2013;52(6):807–15.

[24] Ogata K. Modern control engineering. Prentice Hall; 2010.

[25] Dorf RC, Bishop RH. Modern control systems. Pearson; 2011.

[26] Kennedy J, Eberhart R, editors. Particle swarm optimization. IEEE international conference on neural networks, 1995; November/December 1995.

[27] Kennedy J, Eberhart RC. Swarm intelligence. Morgan Kaufmann Publishers Inc.; 2001, 512 p.

[28] Deb K. Multi-objective optimization. Search methodologies. Springer; 2014. p. 403–49.

[29] Fonseca CM, Fleming PJ. Multiobjective optimization and multiple constraint handling with evolutionary algorithms. I. A unified formulation. IEEE Trans Syst Man Cybern Part A Syst Humans 1998;28(1):26–37.

[30] Hwang CL, Masud ASM. Multiple objective decision making, methods and applications: a state-of-the-art survey. Springer-Verlag; 1979.

[31] Reyes-Sierra M, Coello CC. Multi-objective particle swarm optimizers: a survey of the state-of-the-art. Int J Comput Intell Res 2006;2(3):287–308.

[32] Sahib MA. A novel optimal PID plus second order derivative controller for AVR system. Eng Sci Technol Int J 2015;18:194–206.

[33] Yuhui S, Eberhart RC, editors. Empirical study of particle swarm optimization. Proceedings of the 1999 congress on evolutionary computation, 1999.

[34] Kennedy J, Mendes R, editors. Population structure and particle swarm performance. Proceedings of the 2002 congress on evolutionary computation, 2002.