e-Prime - Advances in Electrical Engineering, Electronics and Energy 2 (2022) 100071

Contents lists available at ScienceDirect

e-Prime - Advances in Electrical Engineering, Electronics and Energy

journal homepage: www.elsevier.com/locate/prime

An improved RUN optimizer based real PID plus second-order derivative controller design as a novel method to enhance transient response and robustness of an automatic voltage regulator

Davut Izci [a,*], Serdar Ekinci [b]

[a] Department of Electronics & Automation, Batman University, Batman 72060, Turkey
[b] Department of Computer Engineering, Batman University, Batman 72100, Turkey

ARTICLE INFO

Keywords:
Runge Kutta optimizer
Pattern search mechanism
Real PIDD^2 controller
Automatic voltage regulator

ABSTRACT

It is crucial to maintain the terminal voltage of a power network in order to reach a desired power quality. In this regard, a system called automatic voltage regulator (AVR) holds a significant role as it allows the adjustment of the terminal voltage. However, an AVR system requires an efficient control technique in order to provide stability. Considering this requirement, a new method is presented in this paper for transient response enhancement and robustness of the automatic voltage regulator (AVR) system. In that sense, the structure of a recently developed metaheuristic algorithm named Runge Kutta optimizer (RUN) is improved. The improved RUN (iRUN) optimizer is obtained by integrating the pattern search (PS) method into the original form of the RUN optimizer. The constructed iRUN optimizer is proposed as an efficient tool to tune a real proportional-integral-derivative (PID) plus second-order derivative (PIDD^2) controller employed in an AVR system. In the proposed control mechanism, a performance index known as ZLG is used as an objective function. The proposed iRUN optimizer based real PIDD^2 controller is comparatively demonstrated to be superior in terms of transient response and robustness against other available and recently reported methods based on PID, PID with filter (PID-F), PID acceleration (PIDA), fractional order PID (FOPID) and PID plus second-order derivative (PIDD^2) controllers that are respectively tuned with metaheuristic optimizers of stochastic fractal search algorithm, Archimedes optimization algorithm, whale optimization algorithm, Henry gas solubility optimization and particle swarm optimization. The proposed method is shown to achieve no overshot and less rise and settling times compared to listed methods. It is also shown to be sufficiently robust in case of potential variations that may occur in the parameters of the system. The superiority is further demonstrated using thirty good performing and recently reported different methods that adopt different controllers and metaheuristic optimizers, as well. All comparative analyses confirm the excellent capability of the proposed iRUN optimizer based real PIDD^2 controller for the AVR system control.

1. Introduction

1.1. Motivation

The stability of a power system holds a great importance due to it is effect on the performance and lifecycle of the connected appliances and its undeniable place in daily lives of human beings. Any deviation from the desired level of a power system may have catastrophic effects for the connected users. Such a case would also cause a significant reduction in terms of the efficiency of the appliances and their standard life cycle. Therefore, a power system is expected to provide a stable and constant output voltage and prevent fluctuations regardless of the variations that may occur in the load or the system parameters [1]. An automatic voltage regulator (AVR) can be integrated with a power system in order to provide a regulated output voltage [2]. Although an AVR has a vital role due to its outstanding contribution of power quality enhancement, it cannot provide any beneficial result without an appropriate control mechanism. There has been a tremendous effort to form a control method that can effectively be used for the AVR system control. However, obtaining the best method still remains as a difficult challenge. Therefore, the motivation of this work is to present a new and more effective control method for efficient operation of an AVR system such that the output voltage of a power system can be regulated effectively.

* Corresponding author.
E-mail address: davut.izci@batman.edu.tr (D. Izci).

https://doi.org/10.1016/j.prime.2022.100071
Received 25 June 2022; Received in revised form 30 August 2022; Accepted 21 September 2022
Available online 22 September 2022
2772-6711/© 2022 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Nomenclature

ALO	ant lion optimizer
AOA	Archimedes optimization algorithm
AVR	automatic voltage regulator
ChBWO	Chaotic black widow algorithm
CS	cuckoo search algorithm
C-YSGA	chaotic yellow saddle goatfish algorithm
FOPID	fractional order proportional-integral-derivative
GA	genetic algorithm
GBO	gradient-based optimization algorithm
GOA	grasshopper optimization algorithm
GSA	gravitational search algorithm
HGSO	henry gas solubility optimization algorithm
iRUN	Improved Runge Kutta optimizer
ISMA	improved slime mould algorithm
IWOA	improved whale optimization algorithm
JOA	Jaya optimization algorithm
L-RSANM	Lévy flight-based reptile search algorithm with Nelder-Mead
OBASO	opposition-based atom search optimization algorithm
OF	objective function
PI	proportional-integral
PID	proportional-integral-derivative
PIDA	proportional-integral-derivative and acceleration
$PIDD^2$	proportional-integral-derivative plus second-order derivative
PID-F	proportional-integral-derivative with filter
PS	pattern search
PSO	particle swarm optimization
RUN	Runge Kutta optimizer
SA–MRFO	hybrid simulated annealing-manta ray foraging optimization algorithm
SCA	sine–cosine-algorithm
SFS	stochastic fractal search algorithm
SMA	slime mould algorithm
SOS	symbiotic organism search algorithm
SSA	salp SWARM algorithm
TLBO	teaching-learning based optimization
WOA	whale optimization algorithm
ZLG	the employed objective function in this paper

1.2. Literature review

As stated in the previous section, a control mechanism holds a great importance for the efficiency of the AVR system. In Table 1, we present a list of most recent and effective control mechanism in order to demonstrate the struggle of achieving a best performing method for the AVR system control. One can easily determine that a controller mechanism and a tuning algorithm are the two main keys to form an appropriate control method. In terms of controllers, several different structures can be encountered such as proportional-integral-derivative (PID) [3], PID plus second-order derivative ($PIDD^2$) [4], PID acceleration (PIDA) [5], PID with filter (PID-F) [6], sigmoid PID [7], fuzzy PID [8], fractional order PID (FOPID) [9], FOPID with reference model [10] and fractional high-order differential feedback [11] controllers.

The majority of the related works in the literature tend to employ a PID controller which is due to its simpler design and easier applicability [31]. However, the PID controller is not sufficiently capable in case of the need to enhance the dynamic performance of a plant [26]. In such a case, a FOPID controller may be preferred as it is capable of providing good control performance, however, due to fractional order in the integrator and differentiator, higher computational cost become a huge burden [33]. Higher computational cost remains an issue for PIDA controller, as well [34]. Considering those issues, this study proposes the use of a real $PIDD^2$ controller due to its ability of improving phase margin, steady-state accuracy and stability [34]. In this way a more robust controller mechanism can be created for the AVR system control.

As stated earlier, an appropriate tuning algorithm has an undoubted contribution in order to reach a desired performance of a controller. In Table 1, the corresponding tuning algorithms for the reported controller types are also provided. As can be seen from the table, all listed tuning mechanisms are metaheuristic optimizers. The reason of utilization of a metaheuristic optimizers are due to their efficient optimization abilities regardless of the nature of the problem [35]. In the previous works, the researchers attempted to combine different metaheuristic optimizers with different controller types and aimed to reach a good transient performance and robustness for the AVR system. To illustrate their effort, we gathered the performance of the methods listed in Table 1 in terms of percent overshoot, rise time and settling time. Figs. 1, 2 and 3, respectively provide a comparative illustration for the performance of the reported methods in terms of percent overshoot, rise time and settling time.

As can be observed from the comparative illustrations, some of the methods were able to reach no overshoot, however, their performance regarding rise and settling times varied. The researchers in the field tend to continuously create novel methods in order to keep all the values for percent overshoot, rise time and settling time as small as possible. As part of this effort, we develop a novel control method in this work so that the lowest values for the stated performance metrics can be achieved. In this way, an efficient and robust control approach is proposed for the AVR system control. The details and the contributions are provided in the following section.

1.3. Contribution

Although all listed metaheuristic optimizers listed in Table 1 have demonstrated good ability to tune their respective controllers, obtaining the best controller parameters has always been a challenge. As part of this effort, we aim to construct a novel metaheuristic optimizer for AVR system control by improving the structure of the recently developed Runge Kutta (RUN) optimizer [36] with the aid of pattern search (PS) method [37]. The improved RUN (iRUN) optimizer adopts the original form of RUN optimizer to perform good global search whereas the PS method aides the structure in terms of good local search. The iRUN optimizer is used to tune a real $PIDD^2$ controller adopted in an AVR system. In this way, we develop a novel improved metaheuristic optimizer and form a new control method as more efficient approach for the AVR system control. A performance index known as *ZLG* [38] is used as an objective function to help achieving better results. Comparative transient response and robustness analyses confirm the good capability of the proposed method to control an AVR system. In addition, previously reported methods listed in Table 1 are also employed to further demonstrate the capability of the proposed iRUN based real $PIDD^2$ controller for the AVR system. The contributions of this paper can briefly be listed as follows.

i A novel iRUN optimizer is developed by integrating the PS method with the original form of RUN optimizer. In this way, a new metaheuristic optimizer with better global and local search capability is obtained.
ii A real $PIDD^2$ controller is used for the control of an AVR system.
iii The constructed novel iRUN optimizer is used to tune the real $PIDD^2$ controller with the aid of the *ZLG* objective function.

Table 1
Review of the metaheuristic tuned controllers for AVR system.

Method no	Refs.	Year	Employed Controller	Tuning Algorithm
1	[4]	2015	PIDD^2	Particle swarm optimization (PSO)
2	[6]	2016	PID-F	Teaching-learning based optimization (TLBO)
3	[3]	2016	PID	Gravitational search algorithm (GSA)
4	[10]	2017	FOPID with reference model	PSO
5	[12]	2018	PID	Grasshopper optimization algorithm (GOA)
6	[13]	2018	PID	Cuckoo search algorithm (CS)
7	[9]	2018	FOPID	CS
8	[14]	2018	PID	Stochastic fractal search algorithm (SFS)
9	[15]	2019	PID	Jaya optimization algorithm (JOA)
10	[16]	2019	FOPID	Sine–cosine-algorithm (SCA)
11	[5]	2019	PIDA	Whale optimization algorithm (WOA)
12	[11]	2019	Fractional high-order differential feedback	PSO
13	[17]	2019	FOPID	Ant lion optimizer (ALO)
14	[18]	2019	FOPID	Salp swarm algorithm (SSA)
15	[19]	2020	PIDD^2	Opposition-based atom search optimization algorithm (OBASO)
16	[20]	2020	FOPID	Chaotic yellow saddle goatfish algorithm (C-YSGA)
17	[21]	2020	FOPID	Henry gas solubility optimization algorithm (HGSO)
18	[22]	2020	PIDD^2	Improved WOA (IWOA)
19	[23]	2021	FOPID with fractional filter	SCA
20	[24]	2021	PID	Slime mould algorithm (SMA)
21	[25]	2021	PIDD^2	Hybrid simulated annealing-manta ray foraging optimization algorithm (SA-MRFO)
22	[26]	2021	FOPID	Improved SMA (ISMA)
23	[27]	2021	FOPID	Gradient-based optimization algorithm (GBO)
24	[7]	2021	Sigmoid PID	Nonlinear SCA
25	[28]	2022	PID-F	Archimedes optimization algorithm (AOA)
26	[29]	2022	PID-F	Symbiotic organism search algorithm (SOS)
27	[30]	2022	Two degree of freedom fractional-order PI	WOA
28	[31]	2022	PID	Lévy flight-based reptile search algorithm with Nelder-Mead (L-RSANM)
29	[32]	2022	FOPID	Chaotic black widow algorithm (ChBWO)
30	[8]	2022	Fuzzy PID	Genetic algorithm (GA)

iv The proposed iRUN based real PIDD^2 controller is comparatively shown to provide better transient response capability and robustness for the AVR system.

v The previously reported different controller structures tuned with different metaheuristic optimizers further demonstrate the excellent efficiency of the proposed iRUN based real PIDD^2 controller for the AVR system.

1.4. Organization of the paper

This paper is structured as follows. Section 2 provides details on AVR system and its adopted parameters. In Section 3, the structures of the original form of Runge Kutta optimizer, pattern search method and the proposed improved Runge Kutta optimizer are explained. The design procedure of the proposed method and the methodology of this work are presented in Section 4. The following section provides the obtained comparative results. A related discussion is also included in this section. The paper is finally concluded in Section 6.

2. AVR system and its parameters

A typical model of an AVR system can be represented by four main components known as amplifier, exciter, generator, and sensor which are demonstrated in the schematic diagram provided in Fig. 4. The sensor component senses the terminal voltage ($V_{terminal}$) which is compared with the reference input voltage ($V_{reference}$). An error voltage is obtained after the comparison. This error is amplified via the amplifier component. Lastly, the exciter regulates the field current of the rotor in order to adjust the terminal voltage for different loading conditions.

The transfer functions of amplifier, exciter, generator, and sensor are respectively provided in Eqs. (1)–(4) [31]. In here, K_a, K_e, K_g, and K_s are the gain constants whereas τ_a, τ_e, τ_g, and τ_s are the time constants of the amplifier, exciter, generator and the sensor, respectively. In this study, the latter constants were set as $K_a = 10$, $K_e = 1$, $K_g = 1$, $K_s = 1$, $\tau_a = 0.1\ s$, $\tau_e = 0.4\ s$, $\tau_g = 1\ s$, and $\tau_s = 0.01\ s$ by considering the previously reported works [18–22].

$$G_a(s) = \frac{K_a}{1 + \tau_a s} \tag{1}$$

$$G_e(s) = \frac{K_e}{1 + \tau_e s} \tag{2}$$

$$G_g(s) = \frac{K_g}{1 + \tau_g s} \tag{3}$$

$$H_s(s) = \frac{K_s}{1 + \tau_s s} \tag{4}$$

Considering the schematic diagram, the transfer functions of the amplifier, exciter, generator and the sensor together with the respective values of the gain and time constants, the following transfer function representing the relationship between terminal and the reference voltages can be obtained.

$$G(s) = \frac{V_{terminal}(s)}{V_{reference}(s)} = \frac{0.1s + 10}{0.0004s^4 + 0.0454s^3 + 0.555s^2 + 1.51s + 11} \tag{5}$$

Fig. 5 demonstrates the step response of an uncontrolled AVR system. This response was obtained using the transfer function provided in Eq. (5). From the related response, a maximum overshoot of 65.7226%, peak time of 0.7522 s, rise time of 0.2607 s, and settling time of 6.9865 s can be observed. As the operating voltage of a typical power system is within the kilovolt ranges, the demonstrated step response would lead to catastrophic consequences; therefore, the use of an appropriate control mechanism is of utmost importance.

3. Structure of the improved Runge Kutta optimizer

3.1. Overview of the original form of Runge Kutta optimizer

The Runge Kutta (RUN) optimizer is a recent metaheuristic algorithm containing stochastic components to perform optimization tasks [36]. The Runge-Kutta method is a specific formulation used to solve ordinary differential equations. The search logic of RUN optimizer relies on the slope calculated by the latter method. The RUN optimizer starts with randomly generated N positions for a dimension of D using the following definition where r is a random number [36].

$$x_{n,l} = L_l + r \cdot (U_l - L_l) \tag{6}$$

In here, U_l and L_l (l^{th} variable of the problem) respectively stand for

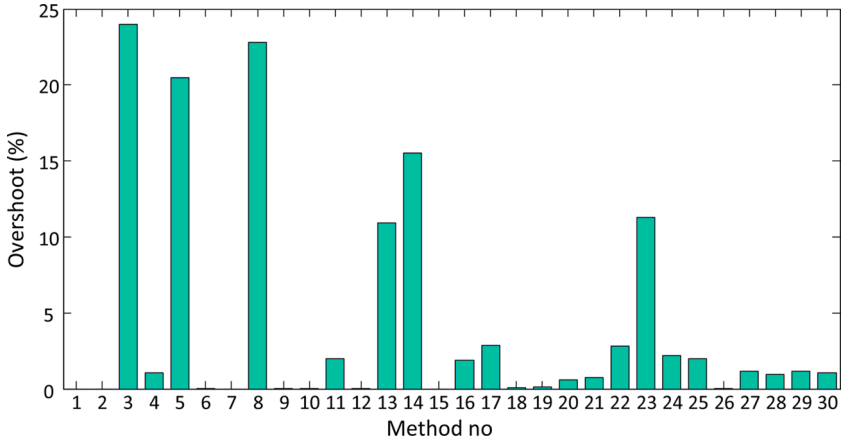


Fig. 1. Percent overshoot for each method listed in Table 1.

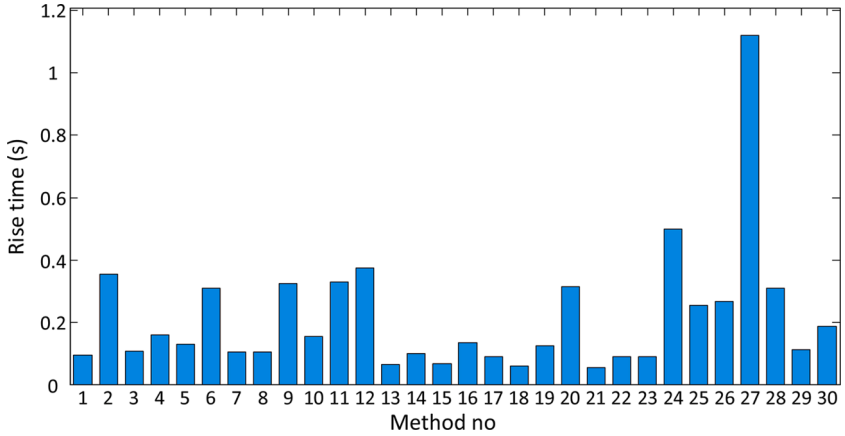


Fig. 2. Rise time for each method listed in Table 1.

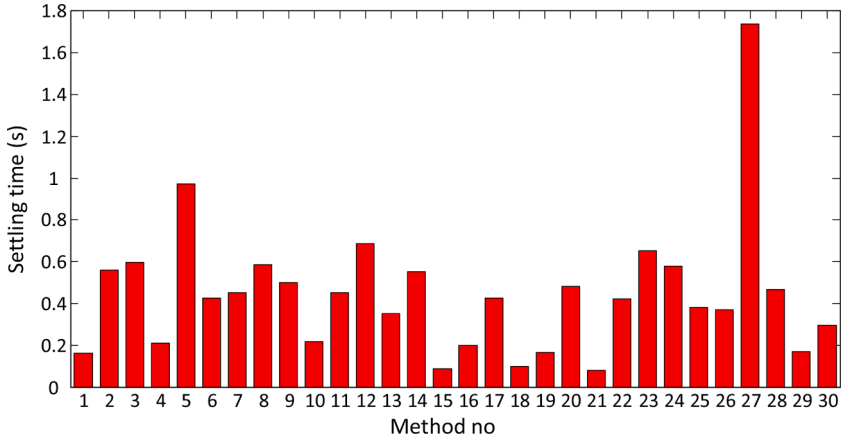


Fig. 3. Settling time for each method listed in Table 1.

the upper and lower bounds ($l = 1,\ 2, \ldots, D$). The search mechanism of RUN optimizer is determined by the coefficients of k_1, k_2, k_3 and k_4 where k_1 is defined as follows [36]:

$$k_1 = \frac{1}{2\Delta x}(r \times x_w - u \times x_b) \tag{7}$$

In here, r stands for a random number within $[0,\ 1]$, x_b is the best position whereas x_w is the worst position (at each iteration). The best and the worst positions are determined based on random solutions (x_{r1}, x_{r2} and x_{r3} where $r1 \neq r2 \neq r3 \neq n$) chosen from the population. The term u is defined as $u = round(1 + r) \times (1 - r)$. The position increment (Δx) is defined as follows [36].

$$\Delta x = 2 \times r \times |Stp| \tag{8}$$

In here, $Stp = r \times ((x_b - r \times x_{avg}) + \gamma)$ and $\gamma = r \times (x_n - r \times (u - l)) \times \exp(-4 \times (i/Maxi))$ where i is the current iteration, $Maxi$ is the maximum number of iterations and x_{avg} is the average of all solutions at each iteration. The coefficients of k_2, k_3 and k_4 are calculated as follows

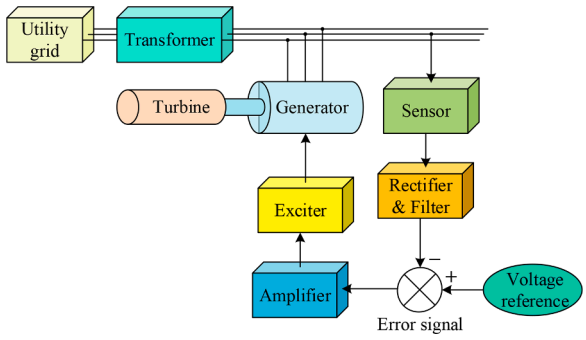


Fig. 4. Schematic diagram of a simple AVR system.

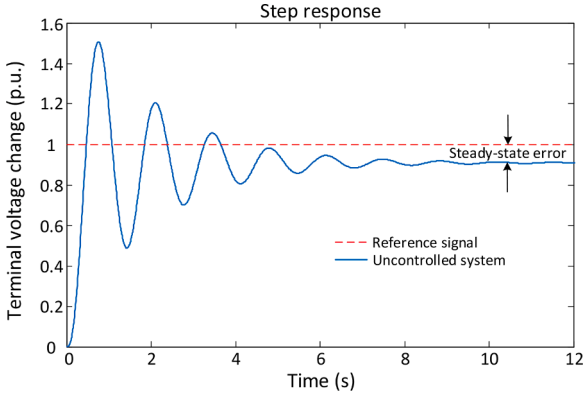


Fig. 5. Step response of uncontrolled AVR system.

[36]:

$$k_2 = \frac{1}{2\Delta x}(r\cdot(x_w + r_1\cdot k_1\cdot\Delta x) - (u\cdot x_b + r_2\cdot k_1\cdot\Delta x)) \tag{9}$$

$$k_3 = \frac{1}{2\Delta x}\left(r\cdot\left(x_w + r_1\cdot\left(\frac{k_2}{2}\right)\cdot\Delta x\right) - \left(u\cdot x_b + r_2\cdot\left(\frac{k_2}{2}\right)\cdot\Delta x\right)\right) \tag{10}$$

$$k_4 = \frac{1}{2\Delta x}(r\cdot(x_w + r_1\cdot k_3\cdot\Delta x) - (u\cdot x_b + r_2\cdot k_3\cdot\Delta x)) \tag{11}$$

where r_1 and r_2 represent random numbers within [0, 1]. The overall search mechanism of the RUN optimizer is defined as given in Eq. (12) [36].

$$SM = \frac{1}{6}(k_1 + (2\times k_2) + (2\times k_3) + k_4)\cdot\ \Delta x \tag{12}$$

The RUN optimizer employs a random number (*rand*) in each iteration and compares it with a predefined value (0.5). For random numbers smaller than 0.5, Eq. (13) is performed to update the solution (performing exploration), otherwise, Eq. (14) is used for the update (performing exploitation) [36]. In here, *SF* is an adaptive factor, μ is a random number and *randn* is a random number with normal distribution.

$$x_{n+1} = (x_c + r\times SF\times g\times x_c) + SF\times SM + \mu\times x_s \tag{13}$$

$$x_{n+1} = (x_m + r\times SF\times g\times x_m) + SF\times SM + \mu\times x_{s'} \tag{14}$$

The terms of x_s, $x_{s'}$, x_c and x_m are described as follows [36]:

$$x_s = randn\cdot(x_m - x_c) \tag{15}$$

$$x_{s'} = randn\cdot(x_{r1} - x_{r2}) \tag{16}$$

$$x_c = \varphi\times x_n + (1-\varphi)\times x_{r1} \tag{17}$$

$$x_m = \varphi\times x_{best} + (1-\varphi)\times x_{lbest} \tag{18}$$

where φ stands for a random number within [0, 1], x_{best} is the best solution found so far and x_{lbest} is the best solution in current iteration. In RUN optimizer, an enhanced solution quality (*ESQ*) mechanism is employed to avoid local minima. In this regard, a random number (w) decreasing through the iterations is used together with another random number (*rand*). The *ESQ* is performed only for *rand* < 0.5. In case of *rand* < 0.5, Eq. (19) is used for $w < 1$, otherwise, Eq. (20) is employed to enhance the solution [36].

$$x_{new2} = x_{new1} + r_{integer}\cdot w\cdot\left|\left(x_{new1} - x_{avg}\right) + randn\right| \tag{19}$$

$$x_{new2} = \left(x_{new1} - x_{avg}\right) + r_{integer}\cdot w\cdot\left|\left(u\cdot x_{new1} - x_{avg}\right) + randn\right| \tag{20}$$

In the last definitions, $r_{integer}$ stands for an integer number (1, 0 or −1) whereas x_{avg} and x_{new1} are calculated as $x_{avg} = (x_{r1} + x_{r2} + x_{r3})/3$ and $x_{new1} = \beta\times x_{avg} + (1-\beta)\times x_{best}$, respectively. However, the solution calculated by x_{new2} may not have a better fitness ($f(x_{new2})$) compared to the fitness of the current solution ($f(x_n)$). In such a case, the following definition is used to calculate a new solution (x_{new3}) [36]:

$$x_{new3} = (x_{new2} - rand\cdot x_{new2}) + SF\cdot(rand\cdot(k_1 + (2\times k_2) + (2\times k_3) + k_4) + (v\cdot x_b - x_{new2}) \tag{21}$$

where v is a random number and equals to $2\times rand$. It is worth noting that x_{new3} is calculated for *rand* < w. The detailed flowchart of the RUN optimizer is provided in Fig. 6.

3.2. Overview of the pattern search method

The pattern search (PS) method describes a derivative-free structure that has a good exploitation capability. A detailed flowchart of the PS method is provided in Fig. 7. As can be seen from this figure, a set of points (may or may not be close to the optimum solution) are generated [37] in the PS method. This method then creates a collection of points, called mesh, around an existing point. This is updated in the next iteration in case of the availability of a new point with a lower objective function value in the mesh. In the PS method, an initial point (X_0) is defined by the user as a starting point for the search.

The size of the mesh is considered as 1 in the first iteration and the pattern (direction) vectors are constructed as $X_0 + [0\ 1]$, $X_0 + [1\ 0]$, $X_0 + [-1\ 0]$ and $X_0 + [0\ -1]$ in order to produce the new mesh points. The objective functions of those points are being calculated until a smaller value than X_0 is found. If a smaller value such as $f(X_1) < f(X_0)$ is found, it is set as a source point. This procedure is called successful poll. After the successful poll, the second iteration is performed by multiplying the current mesh size by 2 such that the following new points are created as: $X_1 + 2\times[0\ 1]$, $X_1 + 2\times[1\ 0]$, $X_1 + 2\times[-1\ 0]$ and $X_1 + 2\times[0\ -1]$. The latter procedure is called expanding stage. The expanding stage continues if a newer point with lower objective function achieves. Otherwise, the mesh size is reduced (contracting stage) by multiplying it with 0.5 which is a reduction factor. The overall process continues until the termination condition is met.

3.3. Proposed improved Runge Kutta optimizer

The proposed improved Runge Kutta (iRUN) optimizer aims to present a better exploration and exploitation capability. In this regard, it employs the original form of RUN optimizer for exploration (global search) and PS method for exploitation (local search). To reach a good balance between the exploration and exploitation stages, the proposed iRUN optimizer starts with the operation of the original form of the RUN optimizer initially. During the process, the PS method is called only twice and for each time it is run for total number of iterations. In another word, at the half of the iterations the PS method is infused into the

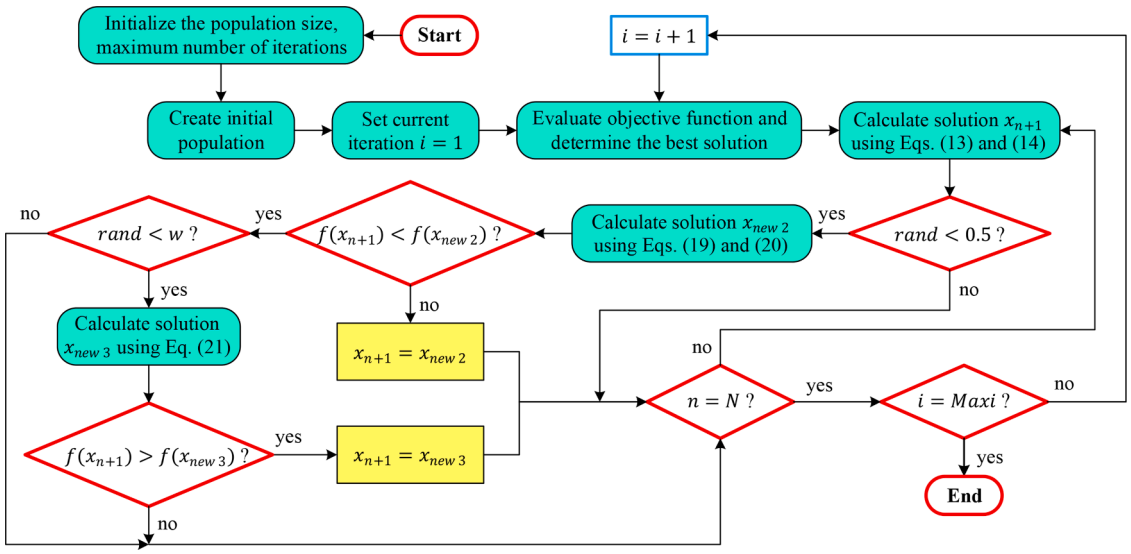


Fig. 6. Flowchart of the RUN optimizer.

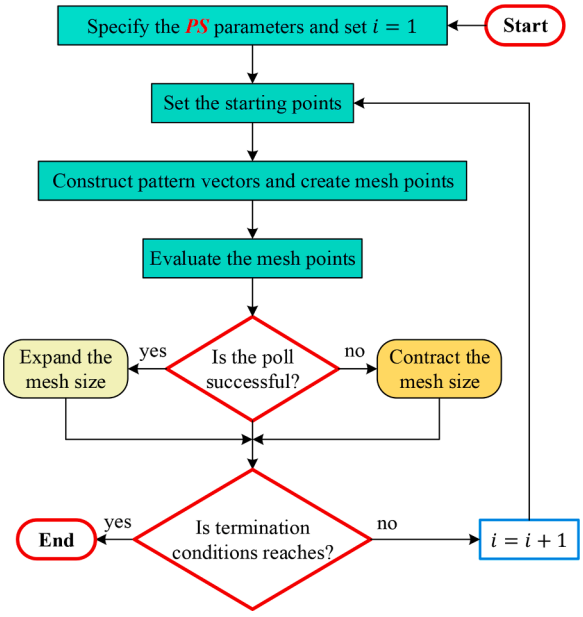


Fig. 7. Flowchart of the PS method.

process and run for total number of iterations. This is repeated at the end of the process once more. With this specific arrangement, the good global search ability of the RUN optimizer is exploited together with the good local search capability of the PS method. Fig. 8 provides a detailed flowchart for the proposed iRUN algorithm.

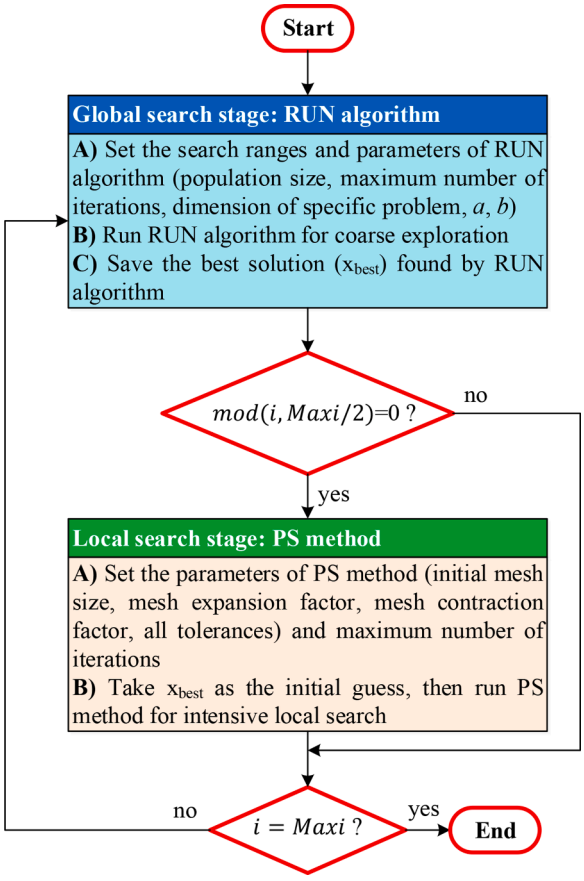


Fig. 8. Flowchart of the proposed iRUN optimizer.

4. Methodology and design

In this work, three key points are considered for the design procedure. The first one is the construction of an effective tuning algorithm. In this regard, the novel iRUN algorithm is proposed. The details of this novel algorithm have already been provided in the previous section. Apart from the provided details, the values of the parameters for the original form of RUN and the proposed iRUN optimizers are provided in Table 2. In addition of those values, the algorithms were run for 30 times for the optimization.

The second key point is related to the choice of an appropriate controller. In this study, a real PIDD^2 controller is employed. The latter controller is a recent variant of widely used PID controller [34]. The reason of employment of the real PIDD^2 controller is due to its ability of improving phase margin, steady-state accuracy and stability [34]. The following equation is used to define the real PIDD^2 controller [34].

Table 2
Parameter settings of RUN and iRUN algorithms for AVR system.

Algorithm	Population size	Total iteration number	Other parameters
RUN	30	50	$a = 20$, $b = 12$
iRUN	30	50	$a = 20$, $b = 12$, *initial mesh size* $= 1$, *mesh expansion factor* $= 2$, *mesh contraction factor* $= 0.5$, *all tolerances* $= 10^{-6}$

$$C(s) = K_p + \frac{K_i}{s} + K_{d1}\left(\frac{N_1 s}{s+N_1}\right) + K_{d2}\left(\frac{N_2 s}{s+N_2}\right)^2 \tag{22}$$

where K_p, K_i, K_{d1} and K_{d2} are proportional, integral, derivative, and second-order derivative gains, respectively. Besides, N_1 and N_2 stand for the filter coefficients. Fig. 9 provides a detailed block diagram of a real PIDD^2 controller.

The last key point of the design is about to employment of an appropriate performance index such that it can be used for minimization in order to obtain the best controller parameters. For the latter purpose, this study employs the following objective function (*OF*) which is also known as *ZLG* [38] objective function.

$$OF = (1 - e^{-\rho})\left(\frac{OS}{100} + e_{ss}\right) + e^{-\rho}(t_s - t_r) \tag{23}$$

In here, *OS*, e_{ss}, t_s and t_r respectively denote the percent overshoot, steady state error, settling time and rise time. Besides, ρ denotes the weighting coefficient which is set to 1 [39]. Fig. 10 provides a detailed implementation procedure of the proposed methodology to tune the parameters of the real PIDD^2 controller for efficient operation of the AVR system.

Table 3 lists the obtained parameters of the real PIDD^2 controller tuned with the original form of RUN and the proposed iRUN optimizers. The respective table also provides the lower and the upper limits of the individual parameters considered for this study.

5. Results and discussion

5.1. Transient response analysis

Initially, a comparative transient response analysis was performed using the values for percent overshoot (*OS*), rise time (t_r) and settling time (t_s). Fig. 11 demonstrates a comparative step response for different reported methods with good capabilities in order to show the excellent capability of the proposed iRUN based real PIDD^2 controller for the AVR system. For the comparison purpose, particle swarm optimization (PSO) based PIDD^2 [4] controller, whale optimization algorithm (WOA) based PIDA [5] controller, Henry gas solubility optimization (HGSO) based FOPID [21] controller, Archimedes optimization algorithm (AOA) based PID-F [28] controller and stochastic fractal search algorithm (SFS) based PID [14] controller were employed along with the original form of RUN and the proposed iRUN optimizers based real PIDD^2 controllers.

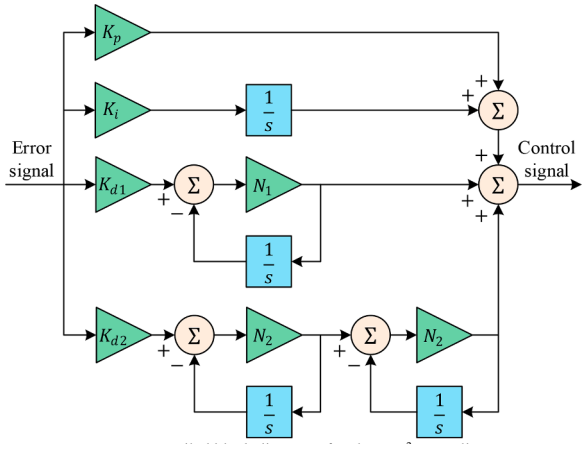


Fig. 9. Detailed block diagram of real PIDD^2 controller.

Apart from the illustrative performance demonstration, a comparative numerical analysis is also provided in Table 4. In this table, the minimized objective function (*OF*) values for different methods are also presented. As can be observed from both illustrations in Fig. 11 and the numerical values in Table 4, the proposed method is able to reach no overshoot (similar that of the PSO-based PIDD^2 [4] controller) with the lowest values for rise and settling times. It also helps achieving the lowest objective function value, as well. In terms of rise time, it demonstrates an improvement of 55.26% compared to the other best reported method (HGSO-based FOPID [21]) in literature. Similarly, it achieves an improvement of 61.71% compared to the other best reported method (PSO-based PIDD^2 [4]) in terms of settling time. The improvement can also be observed for the minimization of the objective function value as it achieved an improved minimization of 67.95% compared to the other best reported method (PSO-based PIDD^2 [4]) in the literature. Therefore, the proposed method for the AVR system controller has an excellent transient response performance.

5.2. Robustness of the proposed method

It is important to observe the behavior of the system against any parameter change as this may be the case for the components of the AVR system. Therefore, a robustness analysis was also performed in this study in order to assess the ability of the proposed method for any potential parameter variation. The robustness was performed by introducing variation (within different ranges of ±50% and ±25%) from the nominal values of time constant parameters of τ_a, τ_e, τ_g and τ_s separately. The numerical values for the transient response performance of the proposed method with respect to variations are provided in Table 5. Apart from that, Figs. 12, 13, 14 and 15, respectively illustrate the step responses for the AVR system with the proposed method against variations of τ_a, τ_e, τ_g and τ_s. As expected, the variations cause the step responses to become worse compared to their nominal values, however, the proposed iRUN based real PIDD^2 controller is able to tolerate all those variations effectively which can clearly be observed from the presented results. Therefore, the proposed method is sufficiently robust to deal with uncertainties caused by the parameter variations.

5.3. Comparison with most recent approaches

In the introduction section, the recent reported methods were briefly listed in Table 1. In this section, we aim to provide a numerical assessment of the proposed iRUN based PIDD^2 controller with respect to those previously reported methods for the AVR system control. Table 6 provides the performance of the respective methods in terms of *OS*, t_r and t_s. Looking at the latter table, one can easily spot that the proposed iRUN-based real PIDD^2 controller is able to reach the best values (highlighted in bold) for the presented transient response metrics. This clearly shows the excellent suitability of the proposed iRUN-based real PIDD^2 controller for the AVR system control. The methods proposed in Refs. [4,6,9,19] were only able to reach no overshoot (highlighted in bold). The proposed iRUN-based real PIDD^2 controller in this study, however, outperformed those methods in terms of rise time by 57.05%, 88.72%, 61.71% and 39.73% for [4,6,9,19], respectively. Similarly, it provided an enhancement of 61.71%, 88.83%, 86.11% and 27.55% with respect

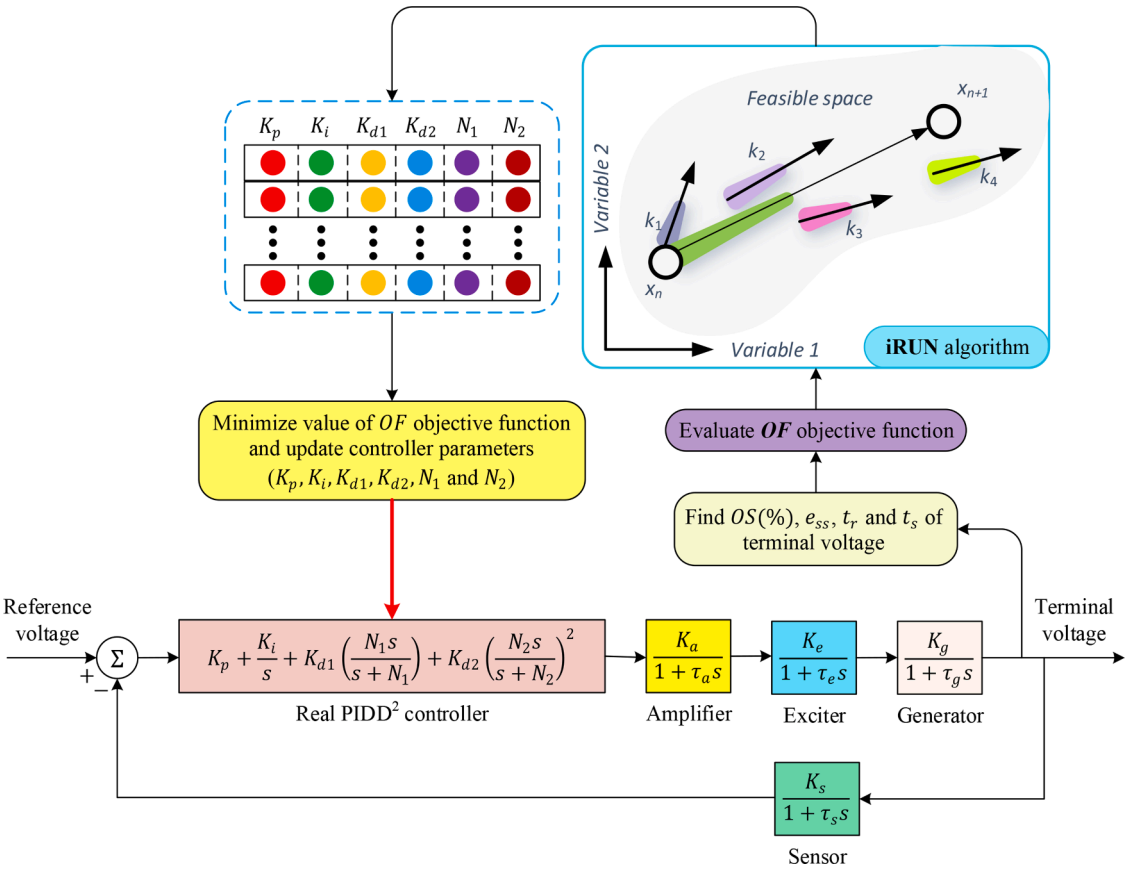


Fig. 10. Implementation of proposed approach to AVR system.

Table 3
Boundaries of real PIDD^2 controller and its optimized values.

Parameter	Lower bound	Upper bound	Tuned by RUN	Tuned by iRUN
K_p	0.001	4	3.8429	3.9904
K_i	0.001	4	2.4158	2.4275
K_{d1}	0.001	4	1.3904	1.5057
K_{d2}	0.001	4	0.10084	0.11893
N_1	100	1000	844.9538	904.0995
N_2	100	1000	717.7803	516.3785

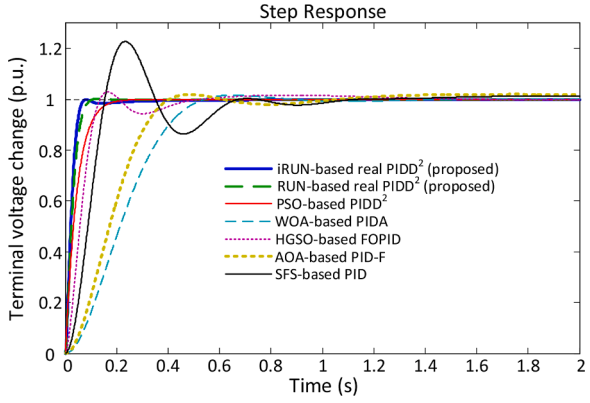


Fig. 11. Step responses of AVR system tuned by different approaches.

to [4,6,9,19] methods in terms of settling time. The latter comparisons alone demonstrate the excellent capacity of the proposed method in this study compared to the recently reported methods with good capabilities.

Table 4
Comparison of OF, OS, t_r and t_s.

Controller design method	OF	OS (%)	t_r (s)	t_s (s)
iRUN-based real PIDD^2 (proposed)	0.0083	0	0.0399	0.0626
RUN-based real PIDD^2 (proposed)	0.0129	0.2225	0.0516	0.0828
PSO-based PIDD^2 [4]	0.0259	0	0.0929	0.1635
WOA-based PIDA [5]	0.0721	1.6475	0.3281	0.4959
HGSO-based FOPID [21]	0.1419	2.8626	0.0892	0.4257
AOA-based PID-F [28]	0.0587	1.9840	0.2520	0.3774
SFS-based PID [14]	0.4564	22.7814	0.1039	0.9532

Table 5
Robustness of the iRUN-based real PIDD^2 controller against parameter variations.

Parameter	Variation	OS (%)	t_r (s)	t_s (s)
τ_a	− 50%	7.3452	0.0180	0.2965
	− 25%	0.7051	0.0285	0.2397
	+ 25%	1.9373	0.0506	0.0766
	+ 50%	4.4017	0.0600	0.2500
τ_e	− 50%	18.5628	0.0162	0.4140
	− 25%	5.1102	0.0266	0.2364
	+ 25%	0	0.0555	0.0968
	+ 50%	1.4440	0.0717	0.1218
τ_g	− 50%	21.4336	0.0159	0.1997
	− 25%	6.2501	0.0262	0.1507
	+ 25%	0	0.0571	0.1137
	+ 50%	0.2522	0.0761	0.1518
τ_s	− 50%	0	0.0533	0.1396
	− 25%	0	0.0452	0.1104
	+ 25%	3.3121	0.0367	0.0926
	+ 50%	7.1383	0.0347	0.1666

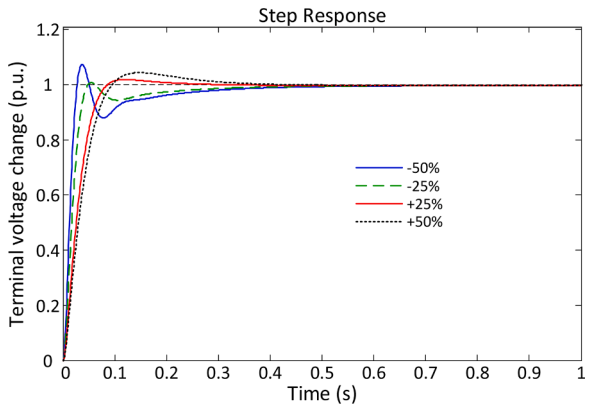


Fig. 12. Step responses of AVR system with iRUN-based real PIDD^2 controller for variation of τ_a.

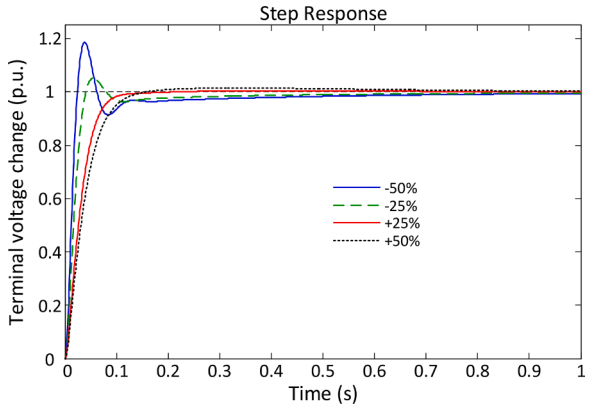


Fig. 13. Step responses of AVR system with iRUN-based real PIDD^2 controller for variation of τ_e.

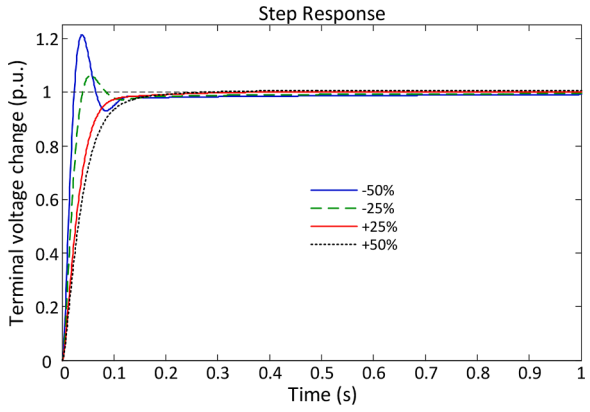


Fig. 14. Step responses of AVR system with iRUN-based real PIDD^2 controller for variation of τ_g.

6. Conclusion

In this work we have investigated the promise of a novel iRUN optimizer in terms of designing a real PIDD^2 controller adopted in an AVR system. The iRUN optimizer has been constructed from integration of PS method and original structure of the RUN optimizer. The developed novel iRUN optimizer has been used to tune a real PIDD^2 controller by minimizing the *ZLG* objective function so that an AVR system with better transient response capability and robustness can be designed. The performance of the proposed iRUN optimizer based real PIDD^2 controller has been compared with the available and best performing methods of PSO based PIDD^2 [4] controller, WOA based PIDA [5] controller, HGSO based FOPID [21] controller, AOA based PID-F [28] controller and SFS based PID [14] controller. The comparisons have demonstrated superior performance of the proposed iRUN based real PIDD^2 controller as it has achieved an improvement of 55.26% compared to HGSO-based FOPID [21] controller (the other best reported method in literature) in terms of rise time, 61.71% compared to PSO-based PIDD^2 [4] controller (the other best reported method in

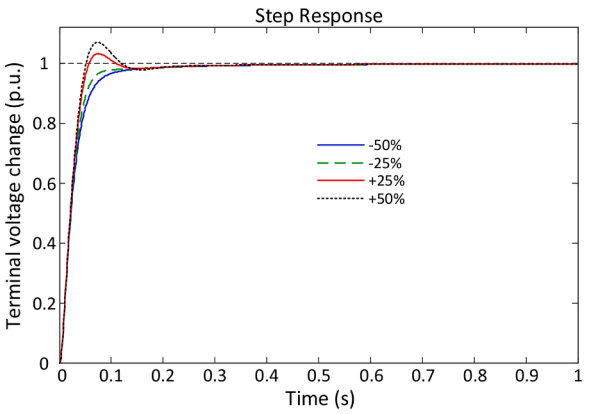


Fig. 15. Step responses of AVR system with iRUN-based real PIDD^2 controller for variation of τ_s.

Table 6
Comparative analysis of transient response evaluation metrics.

Method no	Refs.	*OS* (%)	t_r (s)	t_s (s)
1	[4]	**0**	0.0929	0.1635
2	[6]	**0**	0.3537	0.5603
3	[3]	24.0	0.107	0.597
4	[10]	1.08	0.1592	0.2114
5	[12]	20.5	0.130	0.971
6	[13]	0.0001	0.3080	0.4260
7	[9]	**0**	0.1042	0.4507
8	[14]	22.8	0.103	0.584
9	[15]	0.0016	0.32259	0.50130
10	[16]	0.01	0.1549	0.2183
11	[5]	2	0.328	0.453
12	[11]	0.001	0.373	0.685
13	[17]	10.90	0.0626	0.353
14	[18]	15.5	0.0981	0.551
15	[19]	**0**	0.0662	0.0864
16	[20]	1.89	0.1347	0.2
17	[21]	2.8626	0.0892	0.4257
18	[22]	0.0653	0.0584	0.0982
19	[23]	0.1262	0.1230	0.1670
20	[24]	0.6071	0.3149	0.4817
21	[25]	0.7562	0.0535	0.0798
22	[26]	2.8296	0.0889	0.4215
23	[27]	11.3	0.0885	0.653
24	[7]	2.2	0.498	0.579
25	[28]	1.9977	0.2549	0.3816
26	[29]	0.007	0.267	0.371
27	[30]	1.17	1.12	1.74
28	[31]	0.9582	0.3076	0.4669
29	[32]	1.1838	0.1103	0.169
30	[8]	1.0407	0.1857	0.2963
RUN-based real PIDD^2 (proposed)		0.2225	0.0516	0.0828
iRUN-based real PIDD^2 (proposed)		**0**	**0.0399**	**0.0626**

literature) in terms of settling time and 67.95% compared to PSO-based PIDD^2 [4] (the other best reported method in literature) in terms of objective function minimization. It has also been shown to be sufficiently robust in case of potential variations that may occur in the parameters of the system. The excellent performance of the system is further demonstrated in terms of its transient response performance by employing thirty good performing and recently reported different methods that adopt different controllers and metaheuristic optimizers. The comparative evaluations have confirmed the excellent capability of the proposed iRUN optimizer based real PIDD^2 controller for the AVR system. The proposed method presented in this paper has the promise for potential future works regarding controlling different systems such as magnetic levitation, vehicle cruise control, wind turbine and direct current motor.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] S. Habib, G. Abbas, T.A. Jumani, A.A. Bhutto, S. Mirsaeidi, E.M. Ahmed, Improved whale optimization algorithm for transient response, robustness, and stability enhancement of an automatic voltage regulator system, Energies 15 (2022) 5037, https://doi.org/10.3390/en15145037.

[2] M. Furat, G.G. Cucu, Design, implementation, and optimization of sliding mode controller for automatic voltage regulator system, IEEE Access 10 (2022) 55650–55674, https://doi.org/10.1109/ACCESS.2022.3177621.

[3] S. Duman, N. Yörükeren, İ.H. Altaş, Gravitational search algorithm for determining controller parameters in an automatic voltage regulator system, Turkish J. Electr. Eng. Comput. Sci. 24 (2016) 2387–2400, https://doi.org/10.3906/elk-1404-14.

[4] M.A. Sahib, A novel optimal PID plus second order derivative controller for AVR system, Eng. Sci. Technol. Int. J. 18 (2015) 194–206, https://doi.org/10.1016/j.jestch.2014.11.006.

[5] A.M. Mosaad, M.A. Attia, A.Y. Abdelaziz, Whale optimization algorithm to tune PID and PIDA controllers on AVR system, Ain Shams Eng. J. 10 (2019) 755–767, https://doi.org/10.1016/j.asej.2019.07.004.

[6] S. Chatterjee, V. Mukherjee, PID controller for automatic voltage regulator using teaching–learning based optimization technique, Int. J. Electr. Power Energy Syst. 77 (2016) 418–429, https://doi.org/10.1016/j.ijepes.2015.11.010.

[7] M.H. Suid, M.A. Ahmad, Optimal tuning of sigmoid PID controller using nonlinear sine cosine algorithm for the automatic voltage regulator system, ISA Trans. (2021), https://doi.org/10.1016/j.isatra.2021.11.037.

[8] T. Dogruer, M.S. Can, Design and robustness analysis of fuzzy PID controller for automatic voltage regulator system using genetic algorithm, Trans. Inst. Meas. Control 44 (2022) 1862–1873, https://doi.org/10.1177/01423312211066758.

[9] A. Sikander, P. Thakur, R.C. Bansal, S. Rajasekar, A novel technique to design cuckoo search based FOPID controller for AVR in power systems, Comput. Electr. Eng. 70 (2018) 261–274, https://doi.org/10.1016/j.compeleceng.2017.07.005.

[10] X. Li, Y. Wang, N. Li, M. Han, Y. Tang, F. Liu, Optimal fractional order PID controller design for automatic voltage regulator system based on reference model using particle swarm optimization, Int. J. Mach. Learn. Cybern. 8 (2017) 1595–1605, https://doi.org/10.1007/s13042-016-0530-2.

[11] M.S. Ayas, Design of an optimized fractional high-order differential feedback controller for an AVR system, Electr. Eng. 101 (2019) 1221–1233, https://doi.org/10.1007/s00202-019-00842-5.

[12] B. Hekimoğlu, S. Ekinci, Grasshopper optimization algorithm for automatic voltage regulator system, in: Proceedings of the 5th International Conference on Electrical and Electronic Engineering ICEEE 2018, 2018, pp. 152–156, https://doi.org/10.1109/ICEEE2.2018.8391320.

[13] Z. Bingul, O. Karahan, A novel performance criterion approach to optimum design of PID controller using cuckoo search algorithm for AVR system, J. Frankl. Inst. 355 (2018) 5534–5559, https://doi.org/10.1016/j.jfranklin.2018.05.056.

[14] E. Celik, Incorporation of stochastic fractal search algorithm into efficient design of PID controller for an automatic voltage regulator system, Neural Comput. Appl. 30 (2018) 1991–2002, https://doi.org/10.1007/s00521-017-3335-7.

[15] in C. Gong, F. Xhafa, S. Patnaik, M. Tavana, Jaya algorithm-optimized PID controller for AVR system, eds.. Advances in Intelligent Systems and Computing, Springer International Publishing, Cham, 2019, pp. 382–393, https://doi.org/10.1007/978-3-030-02804-6_52.

[16] J. Bhookya, R.K. Jatoth, Optimal FOPID/PID controller parameters tuning for the AVR system based on sine–cosine-algorithm, Evol. Intell. 12 (2019) 725–733, https://doi.org/10.1007/s12065-019-00290-x.

[17] B. Bourouba, S. Ladaci, H. Schulte, Optimal design of fractional order PIλDμ controller for an AVR system using ant lion optimizer, IFAC-PapersOnLine 52 (2019) 200–205, https://doi.org/10.1016/j.ifacol.2019.11.304.

[18] I.A. Khan, A.S. Alghamdi, T.A. Jumani, A. Alamgir, A.B. Awan, A. Khidrani, Salp swarm optimization algorithm-based fractional order PID controller for dynamic response and stability enhancement of an automatic voltage regulator system, Electronics 8 (2019) 1472, https://doi.org/10.3390/electronics8121472. Basel.

[19] S. Ekinci, A. Demiroren, H. Zeynelgil, B. Hekimoğlu, An opposition-based atom search optimization algorithm for automatic voltage regulator system, J. Fac. Eng. Archit. Gazi Univ. 35 (2020) 1141–1158, https://doi.org/10.17341/gazimmfd.598576.

[20] M. Micev, M. Ćalasan, D. Oliva, Fractional order PID controller design for an AVR system using chaotic yellow saddle goatfish algorithm, Mathematics 8 (2020) 1182, https://doi.org/10.3390/math8071182.

[21] S. Ekinci, D. Izci, B. Hekimoglu, Henry gas solubility optimization algorithm based FOPID controller design for automatic voltage regulator, in: Proceedings of the International Conference on Computing, Electronics., IEEE, 2020, pp. 1–6, https://doi.org/10.1109/ICECCE49384.2020.9179406.

[22] D. Mokeddem, S. Mirjalili, Improved whale optimization algorithm applied to design PID plus second-order derivative controller for automatic voltage regulator system, J. Chin. Inst. Eng. 43 (2020) 541–552, https://doi.org/10.1080/02533839.2020.1771205.

[23] M.S. Ayas, E. Sahin, FOPID controller with fractional filter for an automatic voltage regulator, Comput. Electr. Eng. 90 (2021), 106895, https://doi.org/10.1016/j.compeleceng.2020.106895.

[24] D. Izci, S. Ekinci, Comparative performance analysis of slime mould algorithm for efficient design of proportional–integral–derivative controller, Electrica 21 (2021) 151–159, https://doi.org/10.5152/electrica.2021.20077.

[25] M. Micev, M. Ćalasan, Z.M. Ali, H.M. Hasanien, S.H.E. Abdel Aleem, Optimal design of automatic voltage regulation controller using hybrid simulated annealing – manta ray foraging optimization algorithm, Ain Shams Eng. J. 12 (2021) 641–657, https://doi.org/10.1016/j.asej.2020.07.010.

[26] D. Izci, S. Ekinci, H.L. Zeynelgil, J. Hedley, Fractional order PID design based on novel improved slime mould algorithm, Electr. Power Compon. Syst. 49 (2021) 901–918, https://doi.org/10.1080/15325008.2022.2049650.

[27] S.M.A. Altbawi, A.S. Bin Mokhtar, T.A. Jumani, I. Khan, N.N. Hamadneh, A. Khan, Optimal design of Fractional order PID controller based Automatic voltage regulator system using gradient-based optimization algorithm, J. King Saud Univ. Eng. Sci. (2021), https://doi.org/10.1016/j.jksues.2021.07.009.

[28] A. Agwa, S. Elsayed, M. Ahmed, Design of optimal controllers for automatic voltage regulation using archimedes optimizer, Intell. Autom. Soft Comput. 31 (2022) 799–815, https://doi.org/10.32604/iasc.2022.019887.

[29] B. Ozgenc, M.S. Ayas, I.H. Altas, Performance improvement of an AVR system by symbiotic organism search algorithm-based PID-F controller, Neural Comput. Appl. 34 (2022) 7899–7908, https://doi.org/10.1007/s00521-022-06892-4.

[30] V. Padiachy, U. Mehta, S. Azid, S. Prasad, R. Kumar, Two degree of freedom fractional PI scheme for automatic voltage regulation, Eng. Sci. Technol. Int. J. 30 (2022), 101046, https://doi.org/10.1016/j.jestch.2021.08.003.

[31] S. Ekinci, D. Izci, R. Abu Zitar, A.R. Alsoud, L. Abualigah, Development of Lévy flight-based reptile search algorithm with local search ability for power systems engineering design problems, Neural Comput. Appl. (2022), https://doi.org/10.1007/s00521-022-07575-w.

[32] V.K. Munagala, R.K. Jatoth, Improved fractional PIλDμ controller for AVR system using chaotic black widow algorithm, Comput. Electr. Eng. 97 (2022), 107600, https://doi.org/10.1016/j.compeleceng.2021.107600.

[33] D. Izci, S. Ekinci, B. Hekimoğlu, A novel modified Lévy flight distribution algorithm to tune proportional, integral, derivative and acceleration controller on buck converter system, Trans. Inst. Meas. Control 44 (2022) 393–409, https://doi.org/10.1177/01423312211036591.

[34] S. Ekinci, D. Izci, M. Kayri, An effective controller design approach for magnetic levitation system using novel improved manta ray foraging optimization, Arab. J. Sci. Eng. 47 (2022) 9673–9694, https://doi.org/10.1007/s13369-021-06321-z.

[35] D. Izci, An enhanced slime mould algorithm for function optimization, in: Proceedings of the 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications, IEEE, 2021, pp. 1–5, https://doi.org/10.1109/HORA52670.2021.9461325.

[36] I. Ahmadianfar, A.A. Heidari, A.H. Gandomi, X. Chu, H. Chen, RUN beyond the metaphor: an efficient optimization algorithm based on Runge Kutta method, Expert Syst. Appl. 181 (2021), 115079, https://doi.org/10.1016/j.eswa.2021.115079.

[37] R.K. Khadanga, S. Panda, A modified local input signal for SSSC-based damping controller design, Electr. Power Compon. Syst. 49 (2021) 978–989, https://doi.org/10.1080/15325008.2021.1985016.

[38] D. Izci, Design and application of an optimally tuned PID controller for DC motor speed regulation via a novel hybrid Lévy flight distribution and Nelder–mead algorithm, Trans. Inst. Meas. Control 43 (2021) 3195–3211, https://doi.org/10.1177/01423312211019633.

[39] D. Izci, S. Ekinci, H.L. Zeynelgil, J. Hedley, Performance evaluation of a novel improved slime mould algorithm for direct current motor and automatic voltage regulator systems, Trans. Inst. Meas. Control 44 (2022) 435–456, https://doi.org/10.1177/01423312211037967.



Davut Izci received his BSc degree from Dicle University, Turkey, in Electrical and Electronic Engineering and his MSc and PhD degrees from Newcastle University, England - UK, in Mechatronics and Microsystems, respectively. He is currently an Assistant Professor working at Batman University, Turkey. His-research interests are in microsystems development, robotics, RF, IoT and sensing applications along with the applications of metaheuristic optimization algorithms to various control systems.



Serdar Ekinci received his BSc degree in Control Engineering, and his MSc and PhD degrees in Electrical Engineering all from Istanbul Technical University (ITU), in 2007, 2010 and 2015, respectively. He is currently an Associate Professor working in Department of Computer Engineering at Batman University, Turkey. His-areas of interest are electrical power system, stability, control technology and the applications of metaheuristic optimization algorithms to various control systems.