International Journal of Dynamics and Control (2023) 11:1247–1264
https://doi.org/10.1007/s40435-022-01046-9

Optimal PID plus second-order derivative controller design for AVR system using a modified Runge Kutta optimizer and Bode's ideal reference model

Davut Izci[1] · Serdar Ekinci[2] · Seyedali Mirjalili[3,4]

Received: 24 June 2022 / Revised: 30 July 2022 / Accepted: 14 September 2022 / Published online: 2 October 2022

Abstract
This paper presents the development of a new metaheuristic algorithm by modifying one of the recently proposed optimizers named Runge Kutta optimizer (RUN). The modified RUN (mRUN) algorithm is obtained by integrating a modified opposition-based learning (OBL) mechanism into RUN algorithm. A probability coefficient is employed to provide a good balance between exploration and exploitation stages of the mRUN algorithm. The greater ability of the mRUN algorithm over the original RUN algorithm is shown by performing statistical test and illustrating the convergence profiles. The developed algorithm is then proposed as an efficient tool to tune a proportional-integral-derivative (PID) plus second-order derivative (PIDD^2) controller adopted in an automatic voltage regulator (AVR) system. The controlling scheme is further enhanced by integrating the Bode's ideal reference model and using the performance index of integral of squared error as an objective function. The proposed reference model-based PIDD^2 controller tuned by mRUN (mRUN-RM-PIDD^2) approach is demonstrated to be superior in terms of transient and frequency responses compared to other available and best performing approaches reported in the last 5 years. In that respect, PID, fractional order PID (FOPID), PID acceleration (PIDA) and PIDD^2 controllers tuned with the most effective algorithms reported in the last 5 years are adopted for comparisons. The comparative study confirms superior performance of the proposed method.

Keywords Automatic voltage regulator · PIDD^2 controller · Bode's ideal reference model · Runge Kutta optimizer · Modified opposition-based learning

List of symbols

%OS	Percent overshoot
ASO	Atom search optimization
AVR	Automatic voltage regulator
B_W	Bandwidth
C-YSGA	Chaotic yellow saddle goatfish algorithm
D	Dimension size
D_M	Delay margin
E_{SS}	Steady state error
ECSA	Enhanced crow search algorithm
EO	Equilibrium optimizer
ESQ	Enhanced solution quality
FOPID	Fractional order proportional-integral-derivative
K_G	Generator gain
IWOA	Improved whale optimization algorithm
HGSO	Henry gas solubility optimization algorithm
k_1, k_2, k_3 and k_4	The coefficients of the search mechanism in Runge Kutta optimizer
K_A	Amplifier gain of the AVR system and acceleration gain of the PIDA controller
K_D	Derivative gain

✉ Davut Izci
davut.izci@batman.edu.tr

Serdar Ekinci
serdar.ekinci@batman.edu.tr

Seyedali Mirjalili
ali.mirjalili@gmail.com

[1] Department of Electronics and Automation, Batman University, Batman 72060, Turkey

[2] Department of Computer Engineering, Batman University, Batman 72100, Turkey

[3] Centre for Artificial Intelligence Research and Optimization, Torrens University, Adelaide, Australia

[4] Yonsei Frontier Lab, Yonsei University, Seoul, South Korea

K_{DD}	Second derivative gain
K_E	Exciter gain
K_I	Integral gain
K_P	Proportional gain
K_S	Sensor gain
L_l	Lower bound in Runge Kutta optimizer
LUS	Local unimodal sampling
MRFO	Manta ray foraging optimization
mRUN	Modified Runge Kutta optimizer
N	Number of solutions
OBL	Opposition-based learning
P_G	Peak gain
P_M	Phase margin
P_{mOBL}	Probability coefficient in modified opposition-based learning
φ	Random number
PID	Proportional-integral-derivative
PIDA	Proportional-integral-derivative and acceleration
$PIDD^2$	Proportional-integral-derivative plus second-order derivative
$Q_{indicator}$	Quality indicator
r	Random number
rand	Random number
randn	Random number with normal distribution
RM	Reference model
RUN	Runge Kutta optimizer
SA	Simulated annealing
SF	Adaptive factor in Runge Kutta optimizer
SFS	Stochastic fractal search algorithm
SSA	Salp swarm algorithm
T	Total iterations
t	Current iteration
t_{max}	Maximum iteration number
T_P	Peak time
T_R	Rise time
T_S	Settling time
TLBO	Teaching learning-based optimization
U_l	Upper bound in Runge Kutta optimizer
V_E	Error in voltage
V_{ref}	Reference input
V_S	Voltage measured by the sensor
V_T	Voltage of the synchronous generator
ω_c	Crossover frequency
WOA	Whale optimization algorithm
x_b	The best position represented in Runge Kutta optimizer
$\overline{X}$	Opposite solution in opposition-based learning
x_{lbest}	The best solution in current iteration
x_w	The worst position represented in Runge Kutta optimizer
x_{r1}, x_{r2} and x_{r3}	Random solutions in Runge Kutta optimizer
μ	Random number

1 Introduction

Automatic voltage regulator (AVR) is one of the crucial components of a power system network that attracts a great attention from the research community. Its importance arises from its ability to maintain the level of the terminal voltage which improves the power quality. However, performing such a task is not straightforward as the voltage levels vary due to continuously changing load. Therefore, it is vital to employ a controller in order to keep the AVR system at its best performance. In literature, the employment of different controlling structures such as proportional-integral-derivative (PID) controller [1–3], fractional order PID (FOPID) controller [4, 5], PID plus second-order derivative ($PIDD^2$) controller [6] and PID acceleration (PIDA) controller [7] can be observed for AVR system control.

PID controllers are of the most employed structures due to their relatively simple design and easier implementation [8]. However, a PID controller is not capable of improving the dynamic performance of the system compared to other available controller structures [6, 7, 9]. For example, a FOPID controller has an advantage of providing good control performance along with enhancing the robustness, yet the implementation imposes higher computational cost due to fractional order in the integrator and differentiator which require approximation methods. Similarly, a PIDA controller has the ability to respond faster with less overshoot [10] despite higher computational cost. Therefore, in this study, a $PIDD^2$ controller is employed as a more recent and convenient structure for the purpose of AVR system control since it offers better dynamic response and faster convergence [11]. Another reason of the employment of the $PIDD^2$ controller is also due to its demonstrated efficiency for different complex systems such as a multi-area system's automatic generation control [12], blood glucose level optimization [13] and aerodynamical system's control [14].

Apart from the choice of the controller explained above, this study proposes a further improvement of the controlling scheme by integrating Bode's ideal reference model [15] with the $PIDD^2$ controller in order to greatly enhance the ability of the AVR system. Employment of the Bode's ideal reference model helps the system to demonstrate excellent performance since such a structure forces the behavior of the system resembling to the reference model. The latter fact can also be observed in several controlling mechanisms reported for different systems [16–18]. In this regard, it is worth noting that this paper is the first report on integration of $PIDD^2$ controller and Bode's ideal reference model.

For the sake of efficiency, an appropriate tuning mechanism must also be employed so that the advantage of the PIDD^2 controller can be exploited. In that sense, a metaheuristic approach is utilized in this study in order to tune reference model-based PIDD^2 controller. The reason of utilization of a metaheuristic approach is due to efficient optimization abilities (independent of the nature of the problem) of such algorithms [19, 20]. Several different metaheuristic approaches can be seen in the literature for the optimization of the controllers adopted in an AVR system. Some of the employed algorithms, from the last 5 years, can be listed as African buffalo optimization [21], cuckoo search algorithm [22], grasshopper optimization algorithm [23], symbiotic organisms search algorithm [24], water wave optimization [25], improved kidney-inspired algorithm [26], improved spotted hyena optimizer [27], water cycle algorithm [28], ant lion optimizer [29] and sine–cosine algorithm [30]. All the listed metaheuristic algorithms have so far demonstrated incredible ability in terms of tuning the employed controllers effectively. However, obtaining the best controller parameters has always been a challenge as it is crucial for achieving the improved system response.

Considering the above-mentioned challenge together with the demonstrated promise of metaheuristic methods, this paper aims to come up with a novel metaheuristic algorithm for AVR system control by modifying the structure of the recently developed Runge Kutta optimizer (RUN) [31] with the aid of a modified version of opposition-based learning (OBL) [32] mechanism. The greater performance of the modified RUN (mRUN) algorithm is demonstrated through statistical metrics of mean, standard deviation, best, worst and median along with the convergence profile. To further showcase the performance of the mRUN algorithm against its original version, it is proposed to tune the Bode's ideal reference model-based PIDD^2 controller adopted in an AVR system. A performance index known as integral of squared error is used as an objective function to help achieving better results from the proposed controlling structure. The comparative results show the good capability of the proposed reference model-based and PIDD^2 controller tuned by the mRUN algorithm (mRUN-RM-PIDD^2). The capability of the proposed approach is further compared with the PID, FOPID, PIDA and PIDD^2 controllers tuned by different algorithms. It is worth noting that the best performing approaches of equilibrium optimizer-based PID controller [33], stochastic fractal search algorithm-based PID controller [34], enhanced crow search algorithm-based PID controller [35], chaotic yellow saddle goatfish algorithm-based FOPID controller [36], salp swarm algorithm-based FOPID controller [37], Henry gas solubility optimization algorithm-based FOPID controller [38], teaching learning optimization-based PIDA controller [7], local unimodal sampling-based PIDA controller [7], whale optimization algorithm-based PIDA controller [39], atom search optimization-based PIDD^2 controller [40], improved whale optimization algorithm-based PIDD^2 controller [41] and hybrid simulated annealing—manta ray foraging optimization algorithm-based PIDD^2 controller [42] are employed for this purpose.

The proposed approach provided the best results compared to above listed and most effective approaches. The studies from the last 5 years are intentionally chosen for the purpose of demonstrating the excellent capability of the proposed approach. Apart from the achieved results and the aforementioned advantages, the proposed approach presents two significant contributions making it a more convenient method for AVR system design. The fist contribution is the integration of the Bode's ideal reference model with the PIDD^2 controller which is reported for the first time in the literature in terms of designing a control mechanism for AVR system. Such a mechanism allows the AVR system to track an ideal response. The second contribution is the employment of the novel mRUN algorithm for tuning the controller parameters. The advantage of mRUN arises from its good balance of exploration and exploitation stages which allows the optimal tuning of the controller parameters and reaching excellent operation of the AVR system using the proposed method.

This paper is structured as follows. Section 2 provides details on RUN and the proposed mRUN algorithms. The structure of the AVR system is explained in Sect. 3. The following section discusses the analysis of the PIDD^2 controlled AVR system. Section 5 provides details on Bode's ideal reference model, and the design of novel reference model-based PIDD^2 controller using mRUN algorithm. The comparative simulation results and discussions are provided in Sect. 6. Finally, the paper is concluded in Sect. 7.

2 RUN and mRUN algorithms

In this section, the RUN algorithm is first proposed, and its drawbacks are identified. The proposed mechanism to improve its performance is then presented.

2.1 Runge Kutta optimizer

The Runge Kutta optimizer (RUN) [31] is a recent algorithm which contains stochastic components for performing optimization. The search logic of RUN relies on the slope calculated by Runge–Kutta method which is a specific formulation for solving ordinary differential equations. This optimizer is initialized by randomly generating N positions

for a dimension of D using:

$$x_{n,l} = L_l + \mathrm{r} \cdot (\mathrm{U}_l - \mathrm{L}_l) \tag{1}$$

where U_l and L_l, for the lth variable of the problem, are, respectively, the upper and lower bounds ($l = 1, 2, \ldots, D$) and r is a random number. The search mechanism of RUN algorithm is determined by the coefficients of k_1, k_2, k_3 and k_4. The coefficient of k_1 is defined as follows:

$$k_1 = \frac{1}{2\Delta x}(r \times x_\mathrm{w} - u \times x_\mathrm{b}) \tag{2}$$

where 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 latter two terms are determined based on random solutions of x_{r1}, x_{r2} and x_{r3} (chosen from the population) where $r1 \neq r2 \neq r3 \neq n$. Besides, u is defined as $u = \mathrm{round}(1+r) \times (1-r)$. The position increment is shown by Δx which is defined as:

$$\Delta x = 2 \times r \times |\mathrm{Stp}| \tag{3}$$

where $\mathrm{Stp} = r \times ((x_\mathrm{b} - r \times x_\mathrm{avg}) + \gamma)$ and $\gamma = r \times (x_n - r \times (u - l)) \times \exp(-4 \times (i/\mathrm{Maxi}))$. The current iteration is represented by i whereas Maxi stands for the maximum number of iterations. Besides, the average of all solutions at each iteration is denoted by x_avg. The other coefficients of k_2, k_3 and k_4 are calculated as follows.

$$k_2 = \frac{1}{2\Delta x}(r \cdot (x_\mathrm{w} + r_1 \cdot k_1 \cdot \Delta x) - (u \cdot x_\mathrm{b} + r_2 \cdot k_1 \cdot \Delta x)) \tag{4}$$

$$k_3 = \frac{1}{2\Delta x}\left(r \cdot \left(x_\mathrm{w} + r_1 \cdot \left(\frac{k_2}{2}\right) \cdot \Delta x\right) - \left(u \cdot x_\mathrm{b} + r_2 \cdot \left(\frac{k_2}{2}\right) \cdot \Delta x\right)\right) \tag{5}$$

$$k_4 = \frac{1}{2\Delta x}(r \cdot (x_\mathrm{w} + r_1 \cdot k_3 \cdot \Delta x) - (u \cdot x_\mathrm{b} + r_2 \cdot k_3 \cdot \Delta x)) \tag{6}$$

In here, r_1 and r_2 also stand for random numbers within [0, 1]. In RUN algorithm, the overall search mechanism is defined as follows.

$$\mathrm{SM} = \frac{1}{6}(k_1 + (2 \times k_2) + (2 \times k_3) + k_4)\Delta x \tag{7}$$

The RUN algorithm uses a random number (rand) in each iteration and compare it with a predefined value of 0.5. For random numbers smaller than 0.5, Eq. (8) is performed to update the solution (exploration), otherwise, Eq. (9) is performed (exploitation) where SF is an adaptive factor, μ is a random number and randn is a random number with normal distribution.

$$x_{n+1} = (x_\mathrm{c} + r \times \mathrm{SF} \times g \times x_\mathrm{c}) + \mathrm{SF} \times \mathrm{SM} + \mu \times x_\mathrm{s} \tag{8}$$

$$x_{n+1} = (x_\mathrm{m} + r \times \mathrm{SF} \times g \times x_\mathrm{m}) + \mathrm{SF} \times \mathrm{SM} + \mu \times x_{\mathrm{s}'} \tag{9}$$

$$x_\mathrm{s} = \mathrm{randn} \cdot (x_\mathrm{m} - x_\mathrm{c}) \tag{10}$$

$$x_{\mathrm{s}'} = \mathrm{randn} \cdot (x_{r1} - x_{r2}) \tag{11}$$

$$x_\mathrm{c} = \varphi \times x_n + (1 - \varphi) \times x_{r1} \tag{12}$$

$$x_\mathrm{m} = \varphi \times x_\mathrm{best} + (1 - \varphi) \times x_\mathrm{lbest} \tag{13}$$

In above equations, 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. Enhanced solution quality (ESQ) is employed by RUN algorithm to avoid local minima. For this purpose, a random number of w is also used (which is decreased through the iterations) alongside another random number (rand). The ESQ is applied only when rand < 0.5. Assuming the latter case is satisfied, Eq. (14) will be applied for $w < 1$, otherwise, Eq. (15) will be processed to enhance the solution.

$$x_\mathrm{new2} = x_\mathrm{new1} + r_\mathrm{integer} \cdot w \cdot \left|(x_\mathrm{new1} - x_\mathrm{avg}) + \mathrm{randn}\right| \tag{14}$$

$$x_\mathrm{new2} = (x_\mathrm{new1} - x_\mathrm{avg}) + r_\mathrm{integer} \cdot w \cdot \left|(u \cdot x_\mathrm{new1} - x_\mathrm{avg}) + \mathrm{randn}\right| \tag{15}$$

In the latter equations, $x_\mathrm{avg} = (x_{r1} + x_{r2} + x_{r3})/3$ and $x_\mathrm{new1} = \beta \times x_\mathrm{avg} + (1 - \beta) \times x_\mathrm{best}$. Besides, $r_\mathrm{integer}$stands for an integer number (1, 0 or -1). However, the solution calculated by $x_\mathrm{new2}$may not have a better fitness ($f(x_\mathrm{new2})$) compared to the fitness of the current solution ($f(x_n)$). In such a case, Eq. (16) is performed to calculate a new solution (x_new3):

$$\begin{aligned} x_\mathrm{new3} = {} & (x_\mathrm{new2} - \mathrm{rand} \cdot x_\mathrm{new2}) + SF \cdot (\mathrm{rand} \cdot (k_1 + (2 \times k_2) \\ & + (2 \times k_3) + k_4) + (v \cdot x_b - x_\mathrm{new2}) \end{aligned} \tag{16}$$

where v is a random number and equals to $2 \times$ rand. It is worth noting that x_new3 is calculated for rand $< w$.

2.2 Proposed modified Runge Kutta optimizer

The OBL [32] mechanism is a capable machine learning strategy widely used to improve the performance of metaheuristics. The advantage of this approach is due to its ability to provide good opportunity in terms of avoiding local minimum stagnation among candidate solutions [43]. Assume X to be a real number within the range of $[lb, ub]$. Then, a brief explanation regarding the OBL mechanism can be provided by calculating the opposite ($\overline{X}$) number follows:

$$\overline{X} = ub + lb - X \tag{17}$$

The above definition can further be expanded as follows for D-dimensional search space where $X_i \in [ub_i, lb_i]$ and $i \in 1, 2, \ldots, D$.

$$\overline{X}_i = ub_i + lb_i - X_i \tag{18}$$

In the literature, different versions of OBL mechanism can be seen in terms of enhancing the performance of the metaheuristic algorithms. The existing variations of OBL mechanism, proposed so far, can be listed as generalized OBL [44], quasi-OBL [45], modified OBL [46], selective OBL [47], orthogonal OBL [48] and neighborhood OBL [49]. In this study, a novel modified OBL, given below, is proposed which simultaneously calculates the modified opposite solutions:

$$\overline{X}_i = r_1 \cdot ub_i + r_2 \cdot lb_i - r_3 \cdot X_i \tag{19}$$

where r_1, r_2 and r_3 are three different numbers randomly generated within [0, 1]. The best N solutions are chosen from the union set of X and $\overline{X}$ solutions after they are calculated. It is worth noting that unlike the previously listed versions of OBL, the proposed mRUN algorithm employs a probability coefficient (P_{mOBL}) to allow the operation of RUN algorithm or the modified OBL mechanism. The respective probability coefficient is calculated as follows:

$$P_{\text{mOBL}} = v_{\max} - t\frac{v_{\max} - v_{\min}}{t_{\max}} \tag{20}$$

where $t_{\max}$ is the maximum iteration number and t is the current iteration. In order to update the P_{mOBL} in each iteration and decrease it linearly with respect to iteration numbers, $v_{\max}$ is set to 1 and $v_{\min}$ 0.01. A random number (rand) is used in each iteration for comparison with the P_{mOBL}. The modified OBL mechanism is activated for $P_{\text{mOBL}} >$ rand and for $P_{\text{mOBL}} <$ rand RUN algorithm operates only. Such an approach provides a good balance between exploration and exploitation phases. The flowchart given in Fig. 1 shows the operation of the proposed mRUN algorithm in detail.

As can be seen from related figure, the algorithm starts with initializing the population size and maximum number of iterations. Then the objective function is evaluated, through iterations, in order to determine the best solution. This is followed by applying the steps of the RUN algorithm. As can be seen the modified OBL takes part depending on the P_{mOBL}. Finally, the best N solutions are chosen from the union set of X and $\overline{X}$ solutions after they are calculated.

The proposed mRUN algorithm has good potential for designing a well performing AVR system due to its good balance between exploration and exploitation stages which is achieved by employment of the modified OBL structure and its wise integration with the parameter of P_{mOBL}. The latter parameter allows the modified OBL to take part at the beginning of the iterations, thus, increases the explorative behavior. Through iterative process, it eliminates the effect of the modified OBL and allows the algorithm to focus on exploitative tasks. Therefore, a good balance is achieved which makes the mRUN a good candidate for AVR system design.

The computational complexity of RUN algorithm can be given as $O(\text{RUN}) = O(N) + O(N \times T) + O(N \times D \times T)$ by considering the initialization, fitness function evaluation and updating of solutions. In the latter definition, the swarm size, maximum number of iterations and the dimension of the problem are denoted by N, T and D, respectively. On the other hand, the maximum computational complexity for the proposed mRUN algorithm can be given as $O(\text{mRUN})|_{\max} = O(\text{RUN}) + O(N \times T)$. As can be observed, the proposed approach has an additional load which is due to the evaluation of opposite fitness function values. Therefore, the proposed mRUN algorithm has slightly higher computational cost, however, provides better solution quality as it does not converge early and stagnate in local minimum.

3 AVR system model

The components known as amplifier, exciter, generator and sensor are used to model an AVR system. The block diagram in Fig. 2 illustrates the listed components which demonstrates AVR model in the form of closed-loop transfer function. It is also worth noting that the nonlinearity and the saturation are ignored for those components.

The sensor shown in the respective figure is used to sense the terminal voltage of the synchronous generator (V_{T}) and the voltage from the sensor (V_{S}) is compared with the reference input voltage (V_{ref}). This is then compared via the comparator in order to obtain the error in voltage (V_{E}) which is amplified and sent to the exciter. The rotor field current is regulated by the exciter and the terminal voltage of synchronous generator is adjusted to its nominal value under different loading conditions.

Table 1 provides the transfer functions of main components of an AVR system along with their ranges. This paper adopts the following parameters in order to provide a fair comparison with the works presented in [7, 33–42]; $K_{\text{A}} = 10$, $\tau_{\text{A}} = 0.1$ s, $K_{\text{E}} = 1$, $\tau_{\text{E}} = 0.4$ s, $K_{\text{G}} = 1$, $\tau_{\text{G}} = 1$ s, $K_{\text{S}} = 1$ and $\tau_{\text{S}} = 0.01s$. The block diagram of AVR system, given in Fig. 2, is obtained by using those parameters.

The transfer function, $G(s)$, of the system can be extracted as given in Eq. (21) by using this figure.

$$G(s) = \frac{V_{\text{T}}(s)}{V_{\text{ref}}(s)} = \frac{0.1s + 10}{0.0004s^4 + 0.0454s^3 + 0.555s^2 + 1.51s + 11} \tag{21}$$

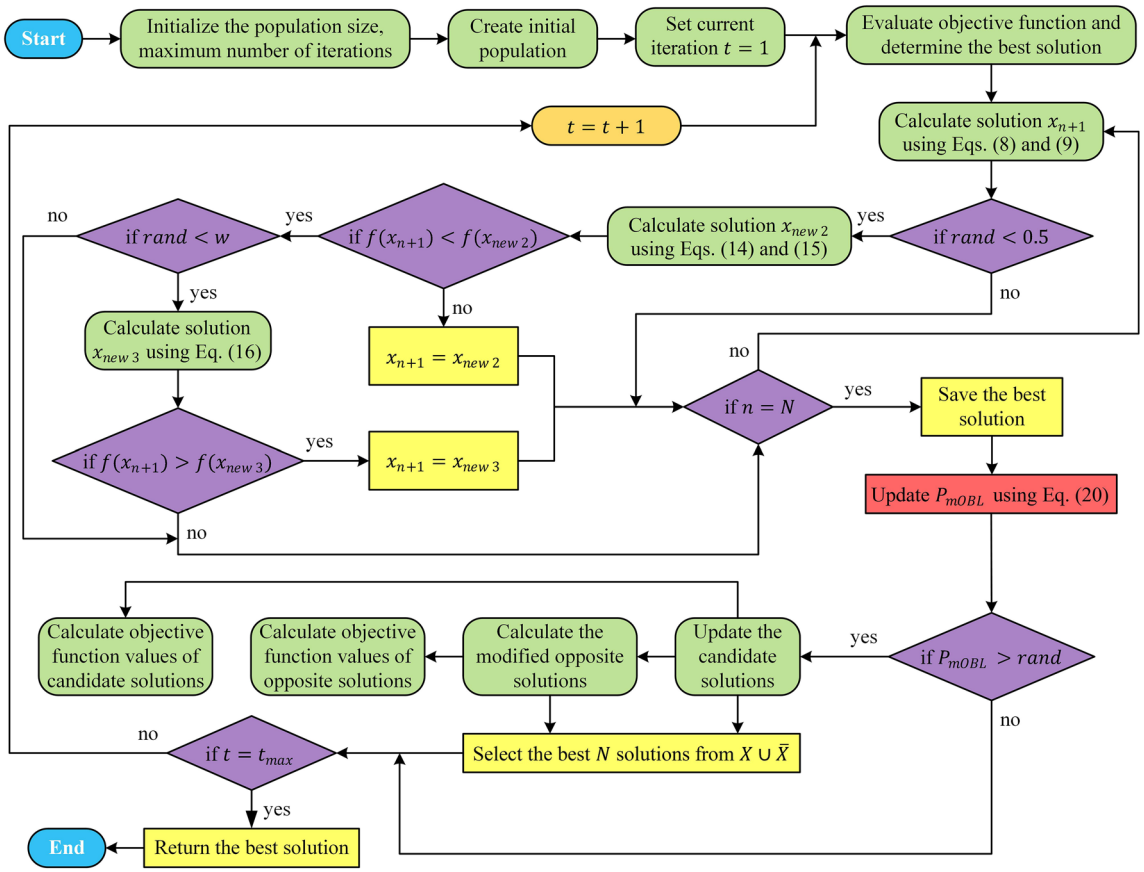

Fig. 1 Flowchart of the mRUN algorithm

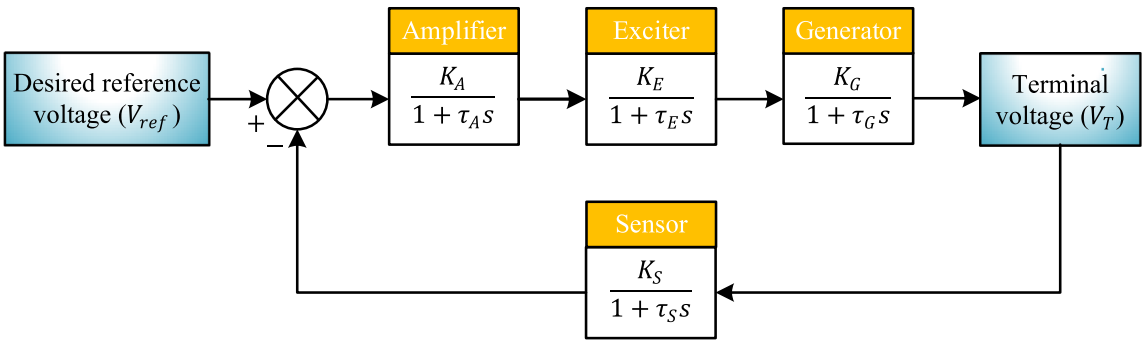

Fig. 2 Block diagram of the AVR system

Table 1 Main components of an AVR system

AVR component	Transfer function	Range of the gain	Range of the time constant (s)
Amplifier	$G_{\mathrm{A}}(s) = \frac{K_{\mathrm{A}}}{1+\tau_{\mathrm{A}} s}$	$10 \leq K_{\mathrm{A}} \leq 40$	$0.02 \leq \tau_{\mathrm{A}} \leq 0.1$
Exciter	$G_{\mathrm{E}}(s) = \frac{K_{\mathrm{E}}}{1+\tau_{\mathrm{E}} s}$	$1 \leq K_{\mathrm{E}} \leq 10$	$0.4 \leq \tau_{\mathrm{E}} \leq 1$
Generator	$G_{\mathrm{G}}(s) = \frac{K_{\mathrm{G}}}{1+\tau_{\mathrm{G}} s}$	$0.7 \leq K_{\mathrm{G}} \leq 1$	$1 \leq \tau_{\mathrm{G}} \leq 2$
Sensor	$H_{\mathrm{S}}(s) = \frac{K_{\mathrm{S}}}{1+\tau_{\mathrm{S}} s}$	$0.9 \leq K_{\mathrm{S}} \leq 1.1$	$0.001 \leq \tau_{\mathrm{S}} \leq 0.06$

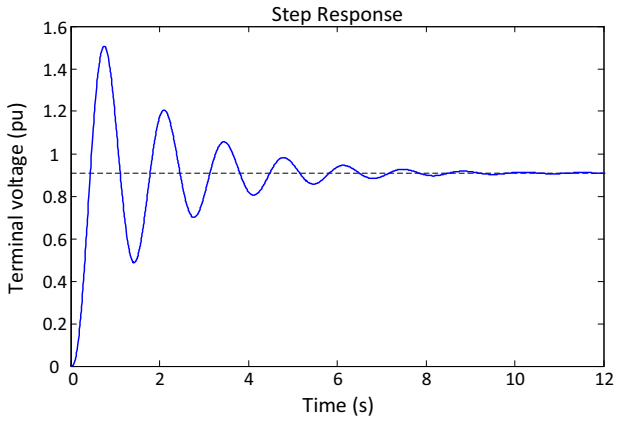

Fig. 3 Terminal voltage step response of the uncontrolled AVR system

Figure 3 illustrates the output voltage response of the AVR system for no controller case by using the Eq. (21). 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 for the output voltage response of the AVR system from the latter figure. Such a response is not acceptable in power systems as the results would be catastrophic due to operating voltage in the range of kilovolts.

4 Analysis of the PIDD^2 controlled AVR system

It is obvious that the PID controllers are used in academy and industry widely [50]. Recently, a new variant of PID controller, known as PIDD^2 controller, has also been used in applications which has been proved to be able to improve phase margin, steady state accuracy and stability of plant [46]. Therefore, the PIDD^2 controller can improve the dynamic response of the AVR system. Equation (22) provides the transfer function of PIDD^2 controller.

$$C_{\mathrm{PIDD^2}}(s) = K_{\mathrm{P}} + \frac{K_{\mathrm{I}}}{s} + K_{\mathrm{D}}s + K_{\mathrm{DD}}s^2 \tag{22}$$

where proportional, integral, derivative and second-order derivative gains are represented by K_{P}, K_{I}, K_{D} and K_{DD}, respectively. Figure 4 shows the block diagram of the AVR system with PIDD^2 controller.

The overall transfer function of an AVR system with PIDD^2 controller is given in Eq. (23).

$$T(s) = \frac{C_{\mathrm{PIDD^2}}(s)G_A(s)G_E(s)G_{\mathrm{G}}(s)}{1 + C_{\mathrm{PIDD^2}}(s)G_A(s)G_E(s)G_{\mathrm{G}}(s)H_{\mathrm{S}}(s)} \tag{23}$$

The entire transfer function of the PIDD^2 controlled AVR system can be obtained as given in Eq. (24) by substituting the respective functions (listed in Table 1) of the main components of the AVR system with the previously stated values and the transfer function of the PIDD^2 controller, given by Eq. (22), into Eq. (23).

$$\begin{aligned} T(s) &= \frac{V_{\mathrm{T}}(s)}{V_{\mathrm{ref}}(s)} \\ &= \frac{0.1K_{\mathrm{DD}}s^4 + (0.1K_{\mathrm{D}} + 10K_{\mathrm{DD}})s^3 + (10K_{\mathrm{D}} + 0.1K_{\mathrm{P}})s^2 + (0.1K_{\mathrm{I}} + 10K_{\mathrm{P}})s + 10K_{\mathrm{I}}}{0.0004s^5 + 0.0454s^4 + (10K_{\mathrm{DD}} + 0.555)s^3 + (10K_{\mathrm{D}} + 1.51)s^2 + (10K_{\mathrm{P}} + 1)s + 10K_{\mathrm{I}}} \end{aligned} \tag{24}$$

5 Design of novel PIDD^2 controller using Bode's ideal reference model and mRUN algorithm

5.1 Bode's ideal reference model

The following equation provides an ideal open-loop transfer function which was proposed in [15]:

$$L(s) = \left(\frac{\omega_c}{s}\right)^{\alpha}, \quad a \in R \tag{25}$$

where α is a real number ($0 < \alpha < 2$) and ω_{c} is the gain crossover frequency of $L(s)$. The slope of the magnitude curve on Bode plot and the phase margin of the system are determined by α. The amplitude in the Bode diagram is a straight line with constant slope (-20α dB/dec). Besides, the phase curve is a horizontal line at $-\alpha\pi/2$ rad. The latter properties indicate that against gain variation, the Bode's ideal transfer function possesses strong robustness meaning the variation of the process gain changes ω_{c} (the crossover frequency) only and maintains a constant phase margin at $\pi(1-\alpha/2)$ rad. An ideal closed-loop transfer function model ($T_{\mathrm{RM}}(s)$) under a unit feedback is provided in Eq. (26).

$$T_{\mathrm{RM}}(s) = \frac{L(s)}{1 + L(s)} = \frac{\omega_{\mathrm{c}}^{\alpha}}{s^{\alpha} + \omega_{\mathrm{c}}^{\alpha}} \tag{26}$$

5.2 PIDD^2 controller design based on reference model

It is crucial to choose a set of appropriate controller parameters of K_{P}, K_{I}, K_{D} and K_{DD} in order to design an efficient PIDD^2 controller for an AVR system. Thus, a novel parameter tuning approach for PIDD^2 controller based on Bode's ideal reference model is explained in this section. Figure 5 illustrates the proposed tuning approach using mRUN algorithm.

As illustrated in the respective figure, a desired output response, $V^*_{\mathrm{ref}}(t)$, is produced by the reference model. The

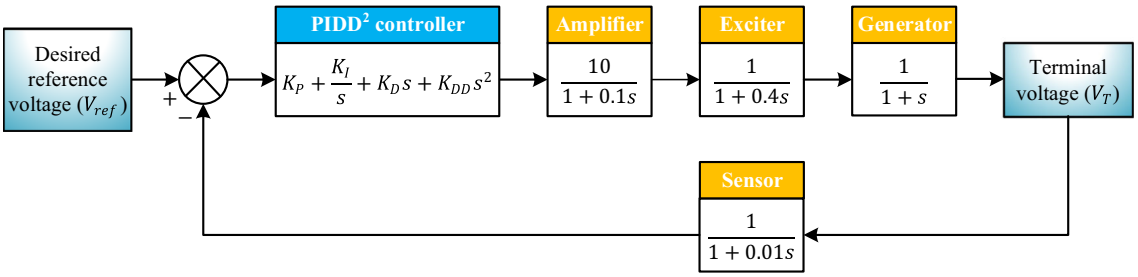

Fig. 4 Block diagram of the PIDD^2 controlled AVR system

latter is specified according to the adopted objective function. The output response of the AVR system with PIDD^2 controller, $V_{\mathrm{T}}(t)$, is then compared with $V^*_{\mathrm{ref}}(t)$ in order to obtain the error function of $e(t)$ which equals the difference between $V^*_{\mathrm{ref}}(t)$ and $V_{\mathrm{T}}(t)$. The latter case means that the controlled system would be closer to the reference model for smaller errors. Therefore, to evaluate the controller, the following performance index can be used as an objective function by considering the latter case.

$$F(K_{\mathrm{P}},\, K_{\mathrm{I}},\, K_{\mathrm{D}},\, K_{\mathrm{DD}}) = \int_0^{\infty} (V^*_{\mathrm{ref}}(t) - V_{\mathrm{T}}(t))^2 \mathrm{d}t \tag{27}$$

The objective function (F) provided above is also named as integral of squared error (ISE) which describes indirectly the level that the controlled system is close to the reference model. Therefore, it is feasible to convert the optimal PIDD^2 controller design to an optimization problem by minimizing $F(K_{\mathrm{P}},\, K_{\mathrm{I}},\, K_{\mathrm{D}},\, K_{\mathrm{DD}})$ via the proposed mRUN algorithm considering the relationship of $K_{\mathrm{P}}^{\min} \leq K_{\mathrm{P}} \leq K_{\mathrm{P}}^{\max}$; $K_{\mathrm{I}}^{\min} \leq K_{\mathrm{I}} \leq K_{\mathrm{I}}^{\max}$; $K_{\mathrm{D}}^{\min} \leq K_{\mathrm{D}} \leq K_{\mathrm{D}}^{\max}$ and $K_{\mathrm{DD}}^{\min} \leq K_{\mathrm{DD}} \leq K_{\mathrm{DD}}^{\max}$.

6 Simulation results and discussions

6.1 Related parameters

The adopted parameters of the respective reference model, controller and the mRUN algorithm have been determined as follows.

Reference model (RM) related parameters: Since the AVR system should avoid oscillations and track the terminal voltage quickly, the parameters of ω_{c} and α have been set to 65 and 1 in $L(s)$, respectively, which resulted the closed-loop system to be $T_{\mathrm{RM}}(s) = 65/(s+65)$. Those values have been determined after extensive trial-and-error evaluations.

PIDD^{2}controller related parameters: There are four parameters associated with the PIDD^2 controller. The range for each parameter has been set to be: $0.001 \leq K_{\mathrm{P}} \leq 4$; $0.001 \leq K_{\mathrm{I}} \leq 4$; $0.001 \leq K_{\mathrm{D}} \leq 4$ and $0.001 \leq K_{\mathrm{DD}} \leq 4$ [41, 46].

mRUN algorithm related parameters: The dimension of each solution (agent), population size and total number of iterations have been set to 4, 40 and 50, respectively. Besides, the values of 20, 12, 0.01 and 1 have been used for the parameters of a, b, $v_{\min}$ and $v_{\max}$, respectively. In terms of a and b, the default values of the RUN algorithm [31] have been used which were found to be efficient for AVR system design. The other parameter values have been determined after extensive simulations. Considering the dimension of the problem (K_{P}, K_{I}, K_{D} and K_{DD}), the stated numbers for agent, population size and total number of iterations have been evaluated to be sufficient. It has been observed that choosing higher numbers for those parameters does not provide any better results. Therefore, minimum possible (and also optimal) numbers have been identified for the related parameters. The values of $v_{\min}$ and $v_{\max}$ have been identified such that the proposed algorithm can reach good balance between exploration and exploitation stages.

6.2 Performance of mRUN-RM-PIDD^2 controller

This section aims to demonstrate the better performance of the mRUN algorithm compared to original version of RUN algorithm. Besides, the output of the proposed approach is also illustrated to be close to the reference model in terms of designing a PIDD^2 controller. Both the RUN and mRUN algorithms were run independently for 30 times in order to obtain the parameters of optimal PIDD^2 controller. The F objective function related statistical results are provided in Table 2. The standard deviation of F objective function can be observed to be much smaller from the respective table which indicates that mRUN algorithm is able to find the near optimal solution each time.

Figure 6 illustrates the evolving process of F objective function for the best runs of RUN and the proposed mRUN algorithms. It can be seen that the proposed mRUN algorithm provides the smallest value without stacking into local minimum. Besides, it does not converge early. These results show

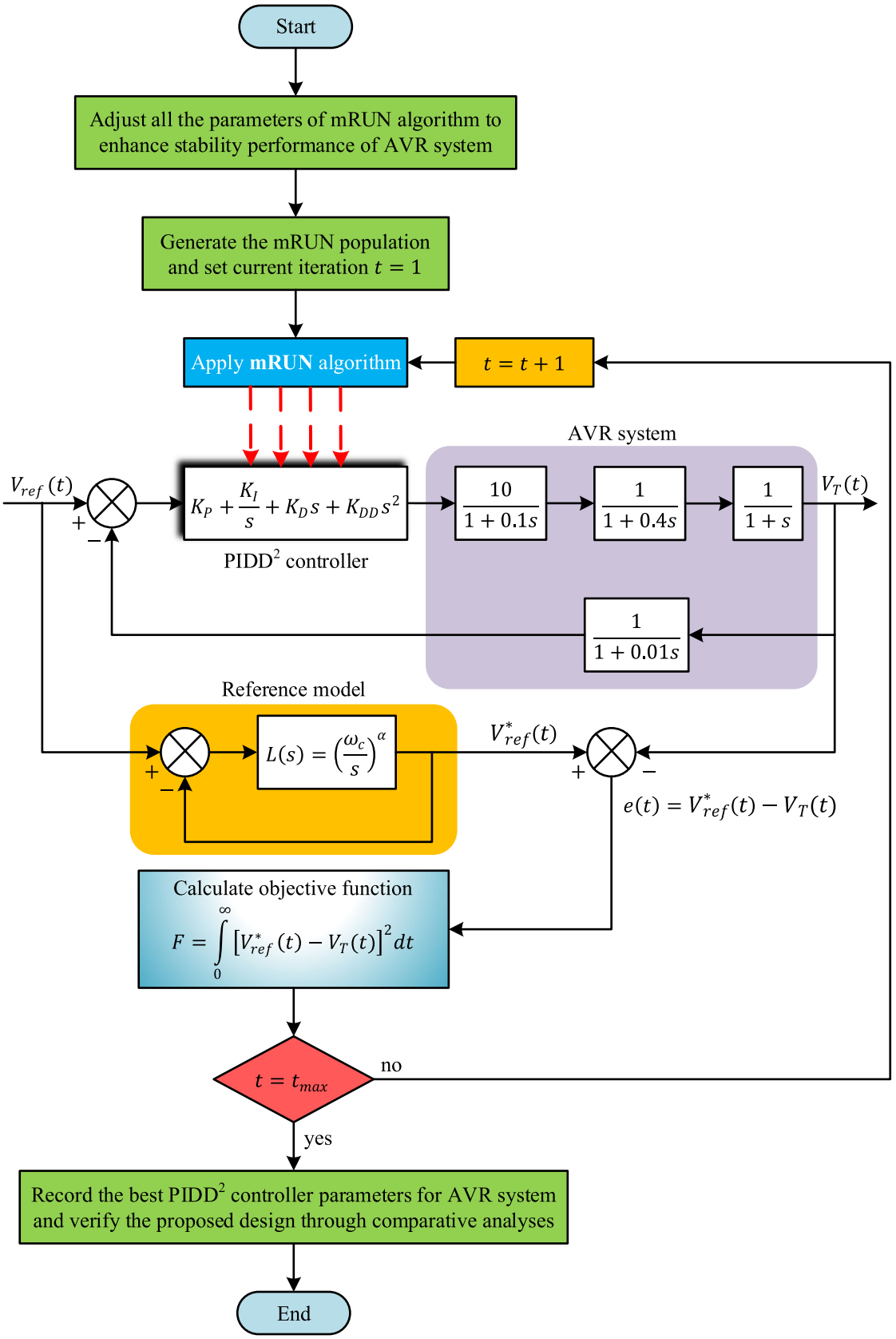

Fig. 5 System structure for PIDD^2 controller tuning based on Bode's ideal transfer function and mRUN algorithm

Table 2 Statistical results of F objective function

Algorithm	Mean	Standard deviation	Best	Worst	Median
RUN	7.6238E−04	1.8065E−05	7.3332E−04	7.9734E−04	7.6126E−04
mRUN	5.7833E−04	6.3897E−06	5.6818E−04	5.8926E−04	5.7770E−04

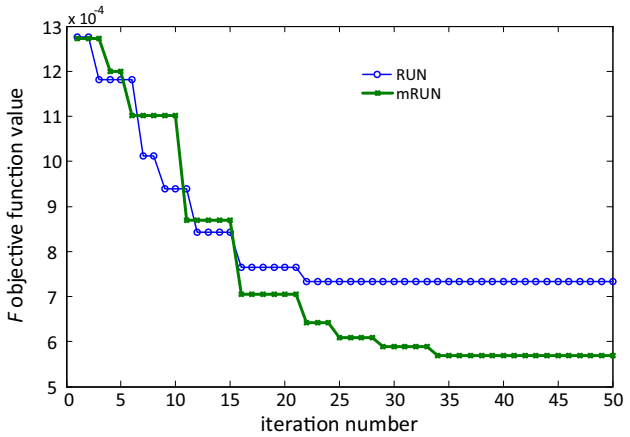

Fig. 6 The evolving process of F objective function for RUN and proposed mRUN algorithms

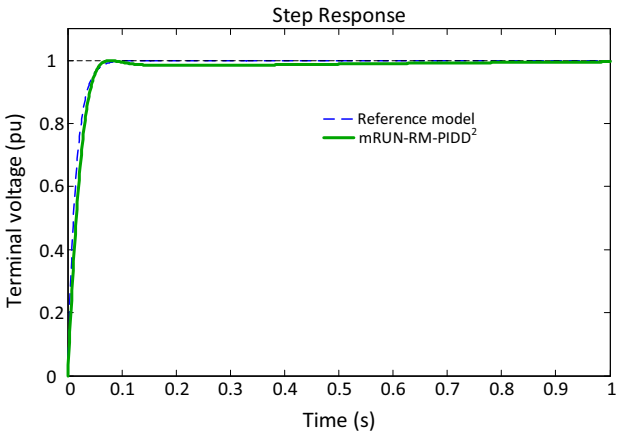

Fig. 7 Terminal voltage step response of reference model and mRUN-RM-PIDD^2 controlled AVR system

that the proposed mRUN algorithm is effective for searching the optimal PIDD^2 controller parameters.

In this study, the best PIDD^2 parameters were determined to be $K_P = 3.9176$, $K_I = 2.6070$, $K_D = 1.8124$ and $K_{DD} = 0.13285$ with the employment of RUN algorithm. The transfer function for AVR system with RUN-RM-PIDD^2 controller given by Eq. (28) would be obtained by substituting the above parameters into Eq. (24).

$$T_{\text{RUN-RM-PIDD}^2}(s) = \frac{0.01329s^4 + 1.51s^3 + 18.52s^2 + 39.44s + 26.07}{0.0004s^5 + 0.0454s^4 + 1.884s^3 + 19.63s^2 + 40.18s + 26.07} \quad (28)$$

Similarly, the best PIDD^2 parameters were determined to be $K_P = 3.9965$, $K_I = 2.5673$, $K_D = 1.8692$ and $K_{DD} = 0.13984$ with the employment of mRUN algorithm. The transfer function for AVR system with mRUN-RM-PIDD^2 controller given by Eq. (29) would be obtained by substituting the above parameters into Eq. (24).

$$T_{m\text{RUN-RM-PIDD}^2}(s) = \frac{0.01398s^4 + 1.585s^3 + 19.09s^2 + 40.22s + 25.67}{0.0004s^5 + 0.0454s^4 + 1.953s^3 + 20.2s^2 + 40.97s + 25.67} \quad (29)$$

Table 3 provides the comparative performance index in terms of percent overshoot (%OS), rise time (T_R), settling time (T_S), peak time (T_P) and steady state error (E_{ss}) for AVR system without controller, Bode's ideal reference model along with the AVR system with the RUN-RM-PIDD^2 and mRUN-RM-PIDD^2 controllers.

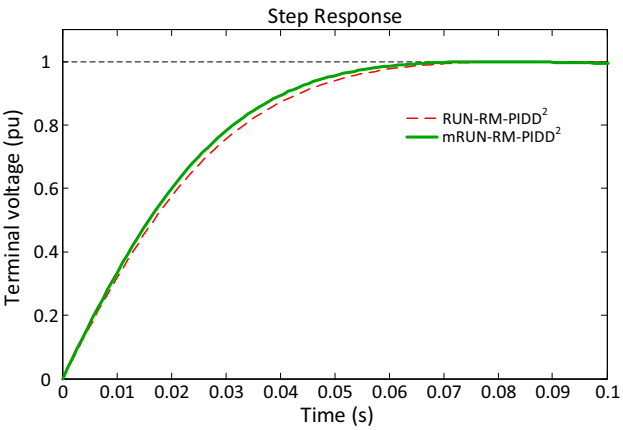

Fig. 8 Terminal voltage step response of mRUN-RM-PIDD^2 and RUN-RM-PIDD^2 controlled AVR system

Besides, terminal voltage unit step responses of both reference model and mRUN-RM-PIDD^2 controlled AVR system are illustrated in Fig. 7. The mRUN-RM-PIDD^2 controlled AVR system can be seen to fully track the step response of the reference model, from Table 3 and Fig. 7, as it has shorter rising time, settling time and peak time with no overshoot compared to the AVR system without controller. Furthermore, the comparative step responses for RUN-RM-PIDD^2 and mRUN-RM-PIDD^2 controlled AVR systems are illustrated in Fig. 8 in order demonstrate the better ability of the proposed mRUN-RM-PIDD^2 based AVR system in terms

Table 3 Performance index comparison in time domain

System type	%OS	T_R (s)	T_S (s)	T_P (s)	E_{ss} (%)
Without controller	65.7226	0.2607	6.9865	0.7522	9.0909
Bode's ideal reference model	0	0.0338	0.0602	0.1622	0
RUN-RM-PIDD^2 controller	0	0.0404	0.0615	0.0861	0
mRUN-RM-PIDD^2 controller	0	0.0379	0.0572	0.0784	0

Table 4 Obtained PID parameters with different approaches

Controller type	K_P	K_I	K_D
ECSA-PID [35]	0.5195	0.3808	0.1625
SFS-PID [34]	1.2837	1.3392	0.7780
EO-PID [33]	0.6829	0.6321	0.2716

of step response compared to RUN-RM-PIDD^2 based AVR system.

6.3 Comparison with PID controllers tuned by recent approaches used in the literature

In this section, the promise of the proposed approach is demonstrated through comparisons with the PID controllers tuned by different approaches available in the literature. The following equation provides the transfer function of a PID controller. Since the PIDD^2 controller is a variant of PID controller, the parameters of K_P, K_I and K_D also stand for proportional, integral and derivative gains, respectively.

$$C_{PID}(s) = K_P + \frac{K_I}{s} + K_D s \tag{30}$$

In terms of comparison, the available and good performing approaches of and equilibrium optimizer-based PID (EO-PID [33]) controller, enhanced crow search algorithm-based PID (ECSA-PID [35]) controller and stochastic fractal search algorithm-based PID (SFS-PID [34]) controller have been used for comparisons. The respective gain parameters obtained by the compared approaches are provided in Table 4.

The comparative step responses of those approaches are plotted against the proposed mRUN-RM-PIDD^2 approach as shown in Fig. 9. As clearly shown in the respective figure, the proposed approach with this study has far better transient response capability for terminal voltage of the AVR system.

In addition, the proposed approach has also been assessed comparatively in terms of frequency response ability. Figure 10 shows the related comparison in terms of Bode diagram. As illustrated in the related figure, the proposed approach demonstrates far better characteristics in terms of frequency response, as well.

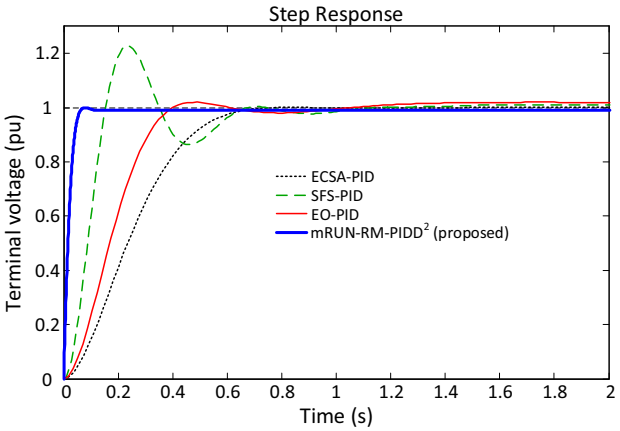

Fig. 9 Comparison of terminal voltage step response of AVR system with various optimized PID controllers and proposed design approach

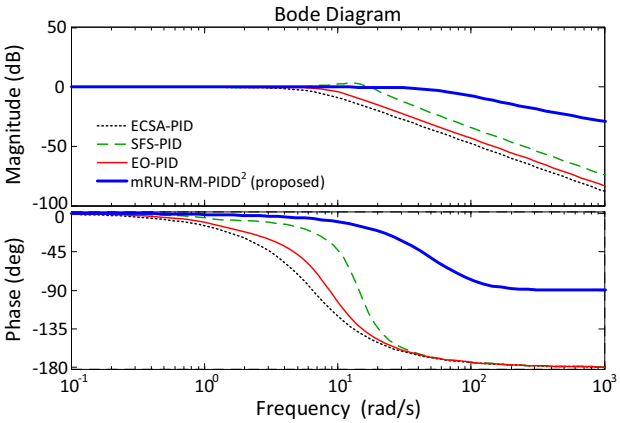

Fig. 10 Bode diagram of AVR system with various optimized PID controllers and proposed design approach

Apart from the illustrative characteristics of the proposed approach, its ability is also shown in terms of numerical values, as well. Table 5 provides the related comparative time (%OS, T_R, T_S and T_P) and frequency (phase margin, P_M, peak gain, P_G, delay margin, D_M and bandwidth B_W) domain characteristics of PID controlled AVR systems designed with different approaches. As can be observed from this table, the provided results in Figs. 9 and 10 are further supported as the minimum values of the time domain related parameters have been obtained along with the maximum values of the

Table 5 Comparative time and frequency domain characteristics of AVR system with PID controllers designed by different approaches and the proposed approach

Controller type	%OS	T_R (s)	T_S (s)	T_P (s)	P_G (dB)	P_M (deg)	D_M (s)	B_W (Hz)
ECSA-PID [35]	0.1801	0.3912	0.6181	1.6248	0.00129	173.9752	7.3853	5.5933
SFS-PID [34]	22.7814	0.1039	0.9532	0.2338	3.11	62.4323	0.0634	19.8210
EO-PID [33]	1.9888	0.2503	0.3734	1.7473	0.168	164.0117	2.0307	9.0663
mRUN-RM-PIDD^2	0	0.0379	0.0572	0.0784	0	180	Inf	56.7224

frequency domain related parameters of P_M, D_M and B_W. Besides, the minimum peak gain has also been obtained with the proposed approach. Therefore, the best system response has been obtained from the mRUN-RM-PIDD^2 based AVR system.

6.4 Comparison with FOPID controllers tuned by recent approaches used in the literature

In addition to the PID controllers tuned by different available approaches, the promise of the proposed mRUN-RM-PIDD^2 approach is also demonstrated through comparisons with the FOPID controllers tuned by different approaches available in the literature. The following equation provides the transfer function of a FOPID controller. Since the FOPID controller is a generalized version of PID controller, the parameters of K_P, K_I and K_D also stand for proportional, integral and derivative gains, respectively. The additional terms of λ and μ, respectively stand for fractional orders of integral and derivative terms.

$$C_{\mathrm{FOPID}}(s) = K_P + \frac{K_I}{s^{\lambda}} + K_D s^{\mu} \tag{31}$$

In terms of comparison, the available and good performing approaches of chaotic yellow saddle goatfish algorithm-based FOPID (C-YSGA-FOPID [36]) controller, salp swarm algorithm-based FOPID (SSA-FOPID [37]) controller and Henry gas solubility optimization algorithm-based FOPID (HGSO-FOPID [38]) controller have been used for comparisons. The respective gain parameters and fractional orders obtained by the compared approaches are provided in Table 6.

The comparative step responses of those approaches are plotted against the proposed mRUN-RM-PIDD^2 approach as shown in Fig. 11. As clearly shown in the respective figure, the proposed approach with this study has far better transient response capability for terminal voltage of the AVR system.

In addition, the proposed approach has also been assessed comparatively in terms of frequency response ability. Figure 12 shows the related comparison in terms of Bode diagram. As illustrated in the related figure, the proposed approach demonstrates far better characteristics in terms of frequency response, as well.

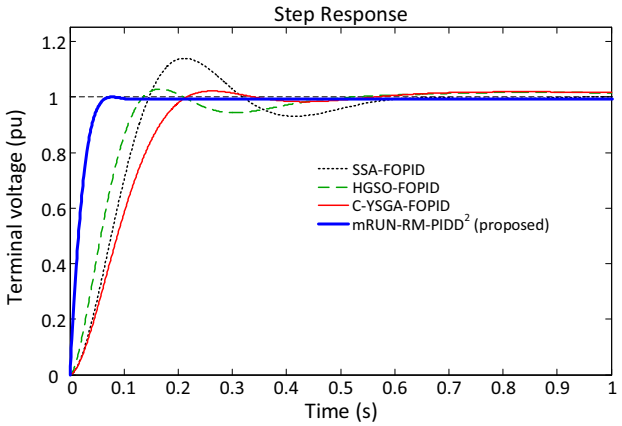

Fig. 11 Comparison of terminal voltage step response of AVR system with various optimized FOPID controllers and proposed design approach

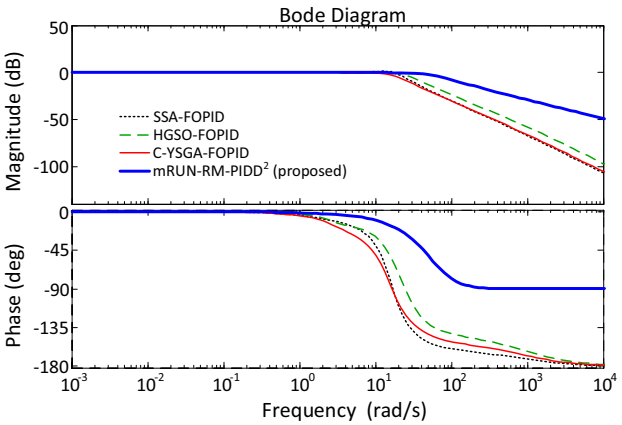

Fig. 12 Bode diagram of AVR system with various optimized FOPID controllers and proposed design approach

Apart from the illustrative characteristics of the proposed approach, its ability is also shown in terms of numerical values, as well. Table 7 provides the related comparative time and frequency domain characteristics of FOPID controlled AVR systems designed with different approaches. As can be observed from this table, the provided results in Figs. 11 and

Table 6 Obtained FOPID parameters with different approaches

Controller type	K_P	K_I	K_D	λ	μ
SSA-FOPID [37]	1.9982	1.1706	0.5750	1.1395	1.1656
HGSO-FOPID [38]	2.6632	1.1314	0.4559	1.2689	1.3663
C-YSGA-FOPID [36]	1.7775	0.9463	0.3525	1.1273	1.2606

Table 7 Comparative time and frequency domain characteristics of AVR system with FOPID controllers designed by different approaches and the proposed approach

Controller type	%OS	T_R (s)	T_S (s)	T_P (s)	P_G (dB)	P_M (deg)	D_M (s)	B_W (Hz)
SSA-FOPID [37]	13.5875	0.1030	0.5544	0.2244	1.33	88.7573	0.0906	21.1798
HGSO-FOPID [38]	2.8626	0.0892	0.4257	0.1625	0.0777	168.7771	1.0919	25.3633
C-YSGA-FOPID [36]	1.8492	0.1598	0.2186	0.8668	0.131	163.2787	0.9877	16.3966
mRUN-RM-PIDD^2	0	0.0379	0.0572	0.0784	0	180	Inf	56.7224

12 are further supported as the best numerical values have been achieved by the proposed mRUN-RM-PIDD^2 based AVR system.

6.5 Comparison with PIDA controllers tuned by recent approaches used in the literature

Apart from PID and FOPID controllers tuned by different available approaches, the promise of the proposed approach is also demonstrated through comparisons with the PIDA controllers tuned by different approaches available in the literature. The following equation provides the transfer function of a PIDA controller where K_A stands for the acceleration gain whereas α and β are used to simplify the polynomial function. An additional gain parameter and filter elements makes this controller to be different from a PID controller.

$$C_{\mathrm{PIDA}}(s) = \frac{K_A s^3 + K_D s^2 + K_P s + K_I}{s^3 + \alpha s^2 + \beta s} \tag{32}$$

In terms of comparison, the available and good performing approaches of teaching learning optimization-based PIDA (TLBO-PIDA [7]) controller, local unimodal sampling-based PIDA (LUS-PIDA [7]) controller and whale optimization algorithm-based PIDA (WOA-PIDA [39]) controller have been used for comparisons. The respective gain parameters obtained by the compared approaches are provided in Table 8.

The comparative step responses of those approaches are plotted against the proposed mRUN-RM-PIDD^2 approach as shown in Fig. 13. As clearly shown in the respective figure, the proposed approach with this study has far better transient response capability for terminal voltage of the AVR system.

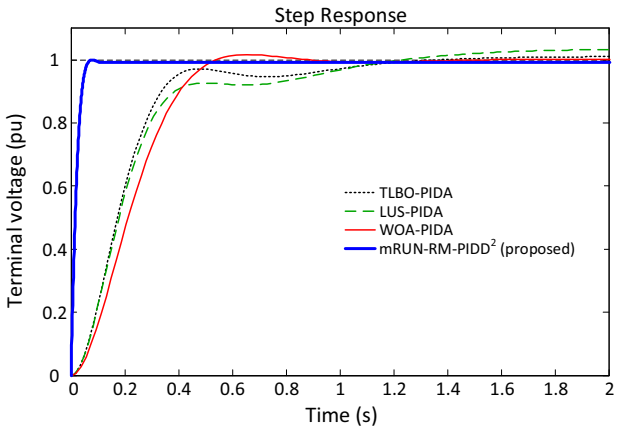

Fig. 13 Comparison of terminal voltage step response of AVR system with various optimized PIDA controllers and proposed design approach

In addition, the proposed approach has also been assessed comparatively in terms of frequency response ability. Figure 14 shows the related comparison in terms of Bode diagram. As illustrated in the related figure, the proposed approach demonstrates far better characteristics in terms of frequency response, as well.

Apart from the illustrative characteristics of the proposed approach, its ability is also shown in terms of numerical values, as well. Table 9 provides the related comparative time and frequency domain characteristics of PIDA controlled AVR systems designed with different approaches. Similar to the case provided in Table 7, Table 9 also presents that the best numerical values have been achieved by the proposed mRUN-RM-PIDD^2 based AVR system which further supports the results in Figs. 13 and 14.

Table 8 Obtained PIDA parameters with different approaches

Controller type	K_P	K_I	K_D	K_A	α	β
TLBO-PIDA [7]	850	421.601	550	150	550	900
LUS-PIDA [7]	783.442	447.048	486.846	149.096	552.567	919.971
WOA-PIDA [39]	777.401	397.741	500.652	103.02	550.118	915.041

Table 9 Comparative time and frequency domain characteristics of AVR system with PIDA controllers designed by different approaches and the proposed approach

Controller type	%OS	T_R (s)	T_S (s)	T_P (s)	P_G (dB)	P_M (deg)	D_M (s)	B_W (Hz)
TLBO-PIDA [7]	1.0029	0.2735	1.0669	2.0059	0	180	Inf	8.7835
LUS-PIDA [7]	3.2018	0.3225	2.7302	1.9906	0.16	162.4775	2.4202	8.1334
WOA-PIDA [39]	1.6475	0.3281	0.4959	0.6540	0.0069	171.7105	4.7003	6.7076
mRUN-RM-PIDD^2	0	0.0379	0.0572	0.0784	0	180	Inf	56.7224

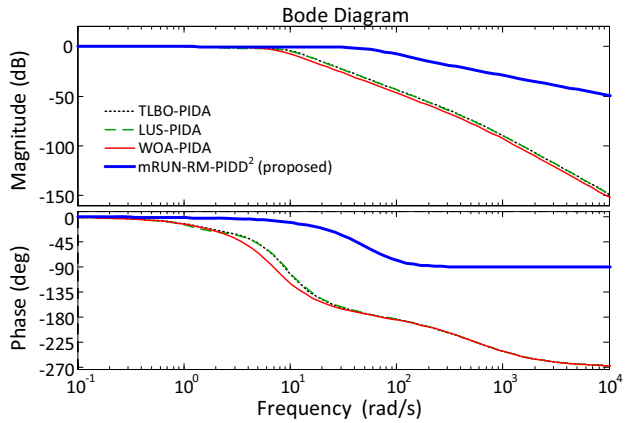

Fig. 14 Bode diagram of AVR system with various optimized PIDA controllers and proposed design approach

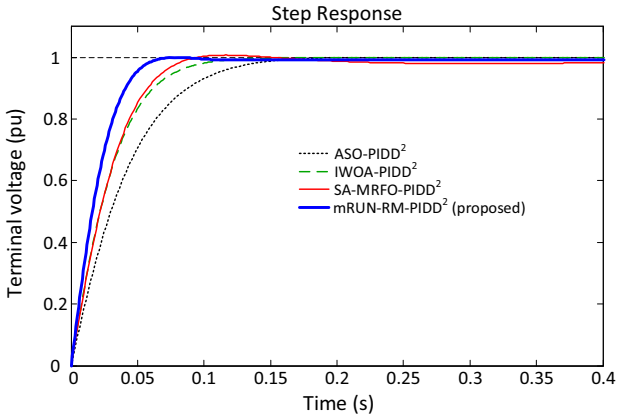

Fig. 15 Comparison of terminal voltage step response of AVR system with various optimized PIDD^2 controllers and proposed design approach

6.6 Comparison with PIDD^2 controllers tuned by recent approaches used in the literature

Lastly, the promise of the proposed approach is demonstrated through comparisons with the PIDD^2 controllers tuned by different approaches available in the literature. In terms of comparison, the available and good performing approaches of hybrid simulated annealing—manta ray foraging optimization algorithm-based PIDD^2 (SA-MRFO-PIDD^2 [42]) controller, improved whale optimization algorithm-based PIDD^2 (IWOA-PIDD^2 [41]) controller and atom search optimization-based PIDD^2 (ASO-PIDD^2 [40]) controller have been used for comparisons. The respective parameters obtained by the compared approaches are provided in Table 10.

The comparative step responses of those approaches are plotted against the proposed mRUN-RM-PIDD^2 approach as shown in Fig. 15. As clearly shown in the respective figure, the proposed approach with this study has much better transient response capability for terminal voltage of the AVR system.

In addition, the proposed approach has also been assessed comparatively in terms of frequency response ability. Figure 16 shows the related comparison in terms of Bode diagram. As illustrated in the related figure, the proposed approach demonstrates better characteristics in terms of frequency response, as well.

Apart from the illustrative characteristics of the proposed approach, its ability is also shown in terms of numerical values, as well. Table 11 provides the related comparative time and frequency domain characteristics of PIDD^2 controlled AVR systems designed with different approaches. As can be observed from this table, the best numerical values have been achieved by the proposed mRUN-RM-PIDD^2 based AVR system which further supports the results in Figs. 15 and 16.

Table 10 Obtained PIDD^2 parameters with different approaches

Controller type	K_P	K_I	K_D	K_{DD}
ASO-PIDD^2 [40]	2.9310	1.9571	1.1033	0.07771
IWOA-PIDD^2 [41]	3.9348	2.5753	1.3985	0.10453
SA-MRFO-PIDD^2 [42]	2.9943	2.9787	1.5882	0.1020

Table 11 Comparative time and frequency domain characteristics of AVR system with PIDD^2 controllers designed by different approaches and the proposed approach

Controller type	OS (%)	T_R (s)	T_S (s)	T_P (s)	P_G (dB)	P_M (deg)	D_M (s)	B_W (Hz)
ASO-PIDD^2 [40]	0	0.0825	0.1363	0.2226	0	180	Inf	26.0864
IWOA-PIDD^2 [41]	0.0068	0.0580	0.0981	0.2105	0	180	Inf	37.3651
SA-MRFO-PIDD^2 [42]	0.7561	0.0535	0.0798	0.1164	0.0575	176.8775	2.3103	39.3200
mRUN-RM-PIDD^2	0	0.0379	0.0572	0.0784	0	180	Inf	56.7224

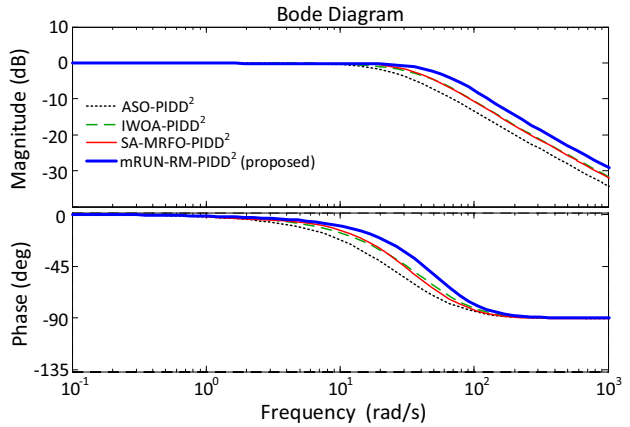

Fig. 16 Bode diagram of AVR system with various optimized PIDD^2 controllers and proposed design approach

6.7 Performance evaluation via quality indicator

Considering the literature regarding the AVR system, one can easily observe that the researchers are keen to improve system behavior in terms of overshoot in percent (%OS), steady state error (E_{ss}), settling (T_S) and rise (T_R) times of the system. The following quality indicator ($Q_{indicator}$) can therefore be used to determine those parameters easier [51]. In the respective quality indicator, ρ stands for the weighting coefficient which was taken as 1 [52].

$$Q_{indicator} = \left(1 - e^{-\rho}\right) \times \left(\frac{\%OS}{100} + E_{ss}\right) + e^{-\rho} \times (T_S - T_R) \quad (33)$$

The following figure illustrates the performance of all compared algorithms. As clearly seen from the related figure, the proposed approach presents better performance in terms of the quality indicator. As can be seen in Fig. 17, the $Q_{indicator}$ has reached the smallest value (0.0071) with the proposed mRUN-RM-PIDD^2 approach compared to other available and best performing approaches reported in the literature.

As has been demonstrated so far, the proposed approach provided the best results compared to other available and most effective approaches reported in the literature. The studies from the last 5 years have intentionally been chosen to demonstrate the greater capability of the proposed approach. Apart from the demonstrated greater results, the proposed approach has two significant novelties that make it a much convenient solution for AVR system design. First, it has a novel controlling structure for AVR design as it integrates Bode's ideal reference model with the PIDD^2 controller. This helps the controller to provide the AVR system tracking an ideal response. Secondly, mRUN algorithm is used for tuning the controller. The ability of the constructed algorithm has been demonstrated by comparing it with the other structures that has been tuned by different algorithms. The advantage of mRUN arises from its good balance of exploration and exploitation stages. Therefore, it has demonstrated a good performance in terms of tuning reference model-based PIDD^2 controller for designing a good performing AVR system.

7 Conclusion

In this paper, the promise of a novel mRUN algorithm has been investigated in terms of designing a Bode's ideal reference model-based PIDD^2 controller adopted in an AVR system. The mRUN algorithm has been formed from integration of modified OBL mechanism and original form of

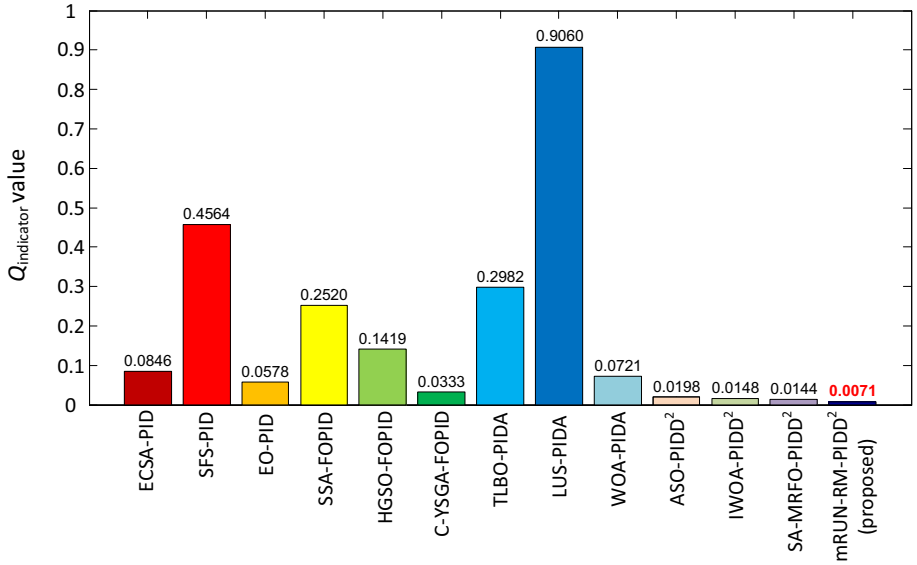

Fig. 17 Comparison of $Q_{indicator}$ objective function values

the RUN algorithm by adopting a probability coefficient. The proposed mRUN algorithm has initially been demonstrated for its excellent balance of exploration and exploitation stages by performing statistical and convergence analyses. The mRUN algorithm has then been proposed to tune a PIDD^2 controller adopted in an AVR system. The Bode's ideal reference model and the performance index known as the integral of squared error has been integrated to further enhance the control scheme. The performance of the proposed mRUN algorithm-based mRUN-RM-PIDD^2 has been assessed against original RUN algorithm-based RUN-RM-PIDD^2 and the better ability of the proposed approach has been demonstrated in terms of transient response. To further verify the superiority of the proposed approach, different structures such as PID, FOPID and PIDA controllers have been considered alongside the PIDD^2 controller, as well. In that respect, the best performing approaches (ECSA-PID controller [35], SFS-PID controller [34], EO-PID controller [33], SSA-FOPID controller [37], HGSO-FOPID controller [38], C-YSGA-FOPID controller [36], TLBO-PIDA controller [7], LUS-PIDA controller [7], WOA-PIDA controller [39], ASO-PIDD^2 controller [40], IWOA-PIDD^2 controller [41] and SA-MRFO-PIDD^2 controller [42]) reported in the last 5 years have been employed for comparisons. The comparative analyses in terms of transient and frequency responses have demonstrated the performance of the mRUN-RM-PIDD^2 approach to be far beyond the abilities of those listed best performing approaches reported in the literature for controlling the AVR system since it has achieved excellent results in terms of metrics of the transient and frequency responses. The proposed approach has the promise for potential future works regarding controlling different systems such as vehicle cruise control, direct current motor, magnetic levitation and wind turbine.

Author contributions Serdar Ekinci and Davut Izci: Conceptualization, Methodology, Investigation, Writing – original draft, Visualization, Software. Seyedali Mirjalili: Writing - Review & Editing.

Funding No funding was received.

Declarations

Competing Interest All authors declare that they have no conflict of interest.

Availability of data and material The authors confirm that all data generated or analyzed during this study are included in this published article.

Code availability The codes are available from the authors upon reasonable request.

References

1. Elsisi M (2021) Optimal design of non-fragile PID controller. Asian J Control 23:729–738. https://doi.org/10.1002/asjc.2248
2. Kiran HU, Tiwari SK (2021) Hybrid BF-PSO algorithm for automatic voltage regulator system. In: Gupta D, Khanna A, Bhattacharyya S et al (eds) Advances in intelligent systems and computing. Springer Singapore, Singapore, pp 145–153
3. Chatterjee S, Mukherjee V (2016) PID controller for automatic voltage regulator using teaching–learning based optimization technique. Int J Electr Power Energy Syst 77:418–429. https://doi.org/10.1016/j.ijepes.2015.11.010
4. Bhullar AK, Kaur R, Sondhi S (2020) Optimization of fractional order controllers for AVR system using distance and Levy-flight based crow search algorithm. IETE J Res. https://doi.org/10.1080/03772063.2020.1782779
5. Izci D, Ekinci S, Zeynelgil HL, Hedley J (2021) Fractional order PID design based on novel improved slime mould algorithm. Electr Power Compon Syst 49:901–918. https://doi.org/10.1080/15325008.2022.2049650

6. Micev M, Calasan M, Radulovic M (2021) Optimal design of real PID plus second-order derivative controller for AVR system. In: 2021 25th international conference on information technology (IT). IEEE, pp 1–4
7. Mosaad AM, Attia MA, Abdelaziz AY (2018) Comparative performance analysis of AVR controllers using modern optimization techniques. Electr Power Compon Syst 46:2117–2130. https://doi.org/10.1080/15325008.2018.1532471
8. Ćalasan M, Micev M, Djurovic Ž, Mageed HMA (2020) Artificial ecosystem-based optimization for optimal tuning of robust PID controllers in AVR systems with limited value of excitation voltage. Int J Electr Eng Educ. https://doi.org/10.1177/0020720920940605
9. Bhookya J, Jatoth RK (2020) Improved Jaya algorithm-based FOPID/PID for AVR system. COMPEL - Int J Comput Math Electr Electron Eng 39:775–790. https://doi.org/10.1108/COMPEL-08-2019-0319
10. Kumar M, Hote YV (2021) Maximum sensitivity-constrained coefficient diagram method-based PIDA controller design: application for load frequency control of an isolated microgrid. Electr Eng. https://doi.org/10.1007/s00202-021-01226-4
11. Sahib MA (2015) A novel optimal PID plus second order derivative controller for AVR system. Eng Sci Technol Int J 18:194–206. https://doi.org/10.1016/j.jestch.2014.11.006
12. Raju M, Saikia LC, Sinha N (2016) Automatic generation control of a multi-area system using ant lion optimizer algorithm based PID plus second order derivative controller. Int J Electr Power Energy Syst 80:52–63. https://doi.org/10.1016/j.ijepes.2016.01.037
13. Jaradat MA, Sawaqed LS, Alzgool MM (2020) Optimization of PIDD2-FLC for blood glucose level using particle swarm optimization with linearly decreasing weight. Biomed Signal Process Control 59:101922. https://doi.org/10.1016/j.bspc.2020.101922
14. Kumar M, Hote YV (2021) Real-time performance analysis of PIDD2 controller for nonlinear twin rotor TITO aerodynamical system. J Intell Robot Syst 101:55. https://doi.org/10.1007/s10846-021-01322-4
15. Barbosa RS, Machado JAT, Ferreira IM (2004) Tuning of PID controllers based on Bode's ideal transfer function. Nonlinear Dyn 38:305–321. https://doi.org/10.1007/s11071-004-3763-7
16. Izci D, Ekinci S, Hekimoğlu B (2022) A novel modified Lévy flight distribution algorithm to tune proportional, integral, derivative and acceleration controller on buck converter system. Trans Inst Meas Control 44:393–409. https://doi.org/10.1177/01423312211036591
17. Pradhan R, Majhi SK, Pradhan JK, Pati BB (2018) Antlion optimizer tuned PID controller based on Bode ideal transfer function for automobile cruise control system. J Ind Inf Integr 9:45–52. https://doi.org/10.1016/j.jii.2018.01.002
18. Zhuo-Yun N, Yi-Min Z, Qing-Guo W et al (2020) Fractional-order PID controller design for time-delay systems based on modified Bode's ideal transfer function. IEEE Access 8:103500–103510. https://doi.org/10.1109/ACCESS.2020.2996265
19. Izci D (2021) An enhanced slime mould algorithm for function optimization. In: 2021 3rd International congress on human–computer interaction, optimization and robotic applications (HORA). IEEE, pp 1–5
20. Izci D, Ekinci S, Orenc S, Demiroren A (2020) Improved artificial electric field algorithm using Nelder–Mead simplex method for optimization problems. In: 2020 4th international symposium on multidisciplinary studies and innovative technologies (ISMSIT). IEEE, pp 1–5
21. Odili JB, Mohmad Kahar MN, Noraziah A (2017) Parameters-tuning of PID controller for automatic voltage regulators using the African buffalo optimization. PLoS ONE 12:e0175901. https://doi.org/10.1371/journal.pone.0175901
22. Bingul Z, Karahan O (2018) A novel performance criterion approach to optimum design of PID controller using cuckoo search algorithm for AVR system. J Frankl Inst 355:5534–5559. https://doi.org/10.1016/j.jfranklin.2018.05.056
23. Hekimoğlu B, Ekinci S (2018) Grasshopper optimization algorithm for automatic voltage regulator system. In: 2018 5th international conference on electrical and electronics engineering, ICEEE 2018. pp 152–156
24. Çelik E, Durgut R (2018) Performance enhancement of automatic voltage regulator by modified cost function and symbiotic organisms search algorithm. Eng Sci Technol Int J 21:1104–1111. https://doi.org/10.1016/j.jestch.2018.08.006
25. Zhou Y, Zhang J, Yang X, Ling Y (2019) Optimization of PID controller based on water wave optimization for an automatic voltage regulator system. Inf Technol Control 48:160–171. https://doi.org/10.5755/j01.itc.48.1.20296
26. Ekinci S, Hekimoglu B (2019) Improved kidney-inspired algorithm approach for tuning of PID controller in AVR system. IEEE Access 7:39935–39947. https://doi.org/10.1109/ACCESS.2019.2906980
27. Zhou G, Li J, Tang Z et al (2020) An improved spotted hyena optimizer for PID parameters in an AVR system. Math Biosci Eng 17:3767–3783. https://doi.org/10.3934/mbe.2020211
28. Pachauri N (2020) Water cycle algorithm-based PID controller for AVR. COMPEL - Int J Comput Math Electr Electron Eng 39:551–567. https://doi.org/10.1108/COMPEL-01-2020-0057
29. Bourouba B, Ladaci S, Schulte H (2019) Optimal design of fractional order PIλDμ controller for an AVR system using Ant Lion Optimizer. IFAC-PapersOnLine 52:200–205. https://doi.org/10.1016/j.ifacol.2019.11.304
30. Bhookya J, Jatoth RK (2019) Optimal FOPID/PID controller parameters tuning for the AVR system based on sine–cosine-algorithm. Evol Intell 12:725–733. https://doi.org/10.1007/s12065-019-00290-x
31. Ahmadianfar I, Heidari AA, Gandomi AH et al (2021) RUN beyond the metaphor: an efficient optimization algorithm based on Runge Kutta method. Expert Syst Appl 181:115079. https://doi.org/10.1016/j.eswa.2021.115079
32. Tizhoosh HR (2005) Opposition-based learning: a new scheme for machine intelligence. In: International conference on computational intelligence for modelling, control and automation and international conference on intelligent agents, web technologies and internet commerce (CIMCA-IAWTIC'06). IEEE, pp 695–701
33. Micev M, Ćalasan M, Oliva D (2021) Design and robustness analysis of an automatic voltage regulator system controller by using equilibrium optimizer algorithm. Comput Electr Eng 89:106930. https://doi.org/10.1016/j.compeleceng.2020.106930
34. Celik E (2018) Incorporation of stochastic fractal search algorithm into efficient design of PID controller for an automatic voltage regulator system. Neural Comput Appl 30:1991–2002. https://doi.org/10.1007/s00521-017-3335-7
35. Bhullar AK, Kaur R, Sondhi S (2020) Enhanced crow search algorithm for AVR optimization. Soft Comput 24:11957–11987. https://doi.org/10.1007/s00500-019-04640-w
36. Micev M, Ćalasan M, Oliva D (2020) Fractional order PID controller design for an AVR system using chaotic yellow saddle goatfish algorithm. Mathematics 8:1182. https://doi.org/10.3390/math8071182
37. Khan IA, Alghamdi AS, Jumani TA et al (2019) Salp swarm optimization algorithm-based fractional order PID controller for dynamic response and stability enhancement of an automatic voltage regulator system. Electronics 8:1472. https://doi.org/10.3390/electronics8121472
38. Ekinci S, Izci D, Hekimoglu B (2020) Henry gas solubility optimization algorithm based FOPID controller design for automatic voltage regulator. In: 2020 International conference on electrical, communication, and computer engineering (ICECCE). IEEE, pp 1–6

39. Mosaad AM, Attia MA, Abdelaziz AY (2019) Whale optimization algorithm to tune PID and PIDA controllers on AVR system. Ain Shams Eng J 10:755–767. https://doi.org/10.1016/j.asej.2019.07.004
40. Ekinci S, Demiroren A, Zeynelgil H, Hekimoğlu B (2020) An opposition-based atom search optimization algorithm for automatic voltage regulator system. J Fac Eng Archit Gazi Univ 35:1141–1158. https://doi.org/10.17341/gazimmfd.598576
41. Mokeddem D, Mirjalili S (2020) Improved whale optimization algorithm applied to design PID plus second-order derivative controller for automatic voltage regulator system. J Chin Inst Eng 43:541–552. https://doi.org/10.1080/02533839.2020.1771205
42. Micev M, Ćalasan M, Ali ZM et al (2021) Optimal design of automatic voltage regulation controller using hybrid simulated annealing—Manta ray foraging optimization algorithm. Ain Shams Eng J 12:641–657. https://doi.org/10.1016/j.asej.2020.07.010
43. Izci D, Ekinci S, Eker E, Kayri M (2020) Improved Manta ray foraging optimization using opposition-based learning for optimization problems. In: 2020 International congress on human–computer interaction, optimization and robotic applications (HORA). IEEE, pp 1–6
44. Wang H, Wu Z, Rahnamayan S et al (2011) Enhancing particle swarm optimization using generalized opposition-based learning. Inf Sci (NY) 181:4699–4714. https://doi.org/10.1016/j.ins.2011.03.016
45. Mandal B, Roy PK (2013) Optimal reactive power dispatch using quasi-oppositional teaching learning based optimization. Int J Electr Power Energy Syst 53:123–134. https://doi.org/10.1016/j.ijepes.2013.04.011
46. Izci D, Ekinci S, Zeynelgil HL, Hedley J (2022) Performance evaluation of a novel improved slime mould algorithm for direct current motor and automatic voltage regulator systems. Trans Inst Meas Control 44:435–456. https://doi.org/10.1177/01423312211037967
47. Dhargupta S, Ghosh M, Mirjalili S, Sarkar R (2020) Selective opposition based grey wolf optimization. Expert Syst Appl 151:113389. https://doi.org/10.1016/j.eswa.2020.113389
48. Wang W, Xu L, Chau K et al (2021) An orthogonal opposition-based-learning Yin–Yang-pair optimization algorithm for engineering optimization. Eng Comput. https://doi.org/10.1007/s00366-020-01248-9
49. Zhao X, Feng S, Hao J et al (2021) Neighborhood opposition-based differential evolution with Gaussian perturbation. Soft Comput 25:27–46. https://doi.org/10.1007/s00500-020-05425-2
50. Izci D, Ekinci S (2021) Comparative performance analysis of slime mould algorithm for efficient design of proportional–integral–derivative controller. Electrica 21:151–159. https://doi.org/10.5152/electrica.2021.20077
51. Gaing Z-L (2004) A particle swarm optimization approach for optimum design of PID controller in AVR system. IEEE Trans Energy Convers 19:384–391. https://doi.org/10.1109/TEC.2003.821821
52. Izci D (2021) 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 43:3195–3211. https://doi.org/10.1177/01423312211019633