fractal and fractional

MDPI

Article

Augmenting the Stability of Automatic Voltage Regulators through Sophisticated Fractional-Order Controllers

Emad A. Mohamed [1,2], Mokhtar Aly [3], Waleed Alhosaini [4,*] and Emad M. Ahmed [4]

[1] Department of Electrical Engineering, College of Engineering, Prince Sattam bin Abdulaziz University, Al Kharj 16278, Saudi Arabia; e.younis@psau.edu.sa
[2] Department of Electrical Engineering, Faculty of Engineering, Aswan University, Aswan 81542, Egypt
[3] Facultad de Ingeniería, Arquitectura y Diseño, Universidad San Sebastián, Bellavista 7, Santiago 8420524, Chile; mokhtar.aly@uss.cl
[4] Department of Electrical Engineering, College of Engineering, Jouf University, Sakaka 72388, Saudi Arabia; emamahmoud@ju.edu.sa
* Correspondence: wsalhosaini@ju.edu.sa

Abstract: The transition from traditional to renewable energy sources is a critical issue in current energy-generation systems, which aims to address climate change and the increased demand for energy. This shift, however, imposes additional burdens on control systems to maintain power system stability and quality within predefined limits. Addressing these challenges, this paper proposes an innovative Modified Hybrid Fractional-Order (MHFO) automatic voltage regulator (AVR) equipped with a fractional-order tilt integral and proportional derivative with a filter plus a second-order derivative with a filter FOTI-PDND^{2}N^2 controller. This advanced controller combines the benefits of a (FOTI) controller, known for enhancing dynamic performance and steady-state response, with a (PDND^{2}N^2) controller to improve system robustness and adaptability. The proposed MHFO controller stands out with its nine tunable parameters, providing more extensive control options than the conventional three-parameter PID controller and the five-parameter FOPID controller. Furthermore, a recent optimization approach using a growth optimizer (GO) has been formulated and applied to optimally adjust the MHFO controller's parameters simultaneously. The performance of the proposed AVR based on the MHFO-GO controller is scrutinized by contrasting it with various established and developed optimization algorithms. The comparative study shows that the AVR based on the MHFO-GO controller surpasses other AVR controllers from the stability, robustness, and dynamic response speed points of view.

Keywords: automatic voltage regulator; fractional-order control; growth optimizer (GO); renewable energy microgrids; voltage control

MSC: 68N30

check for updates

Citation: Mohamed, E.A.; Aly, M.; Alhosaini, W.; Ahmed, E.M. Augmenting the Stability of Automatic Voltage Regulators through Sophisticated Fractional-Order Controllers. *Fractal Fract.* **2024**, *8*, 300. https://doi.org/10.3390/fractalfract8050300

Academic Editors: Isabela Roxana Birs, Cristina I. Muresan and Clara Ionescu

Received: 4 April 2024
Revised: 6 May 2024
Accepted: 17 May 2024
Published: 20 May 2024

1. Introduction

A significant change in the dynamics of electrical power networks has occurred, mostly as a result of changes in energy sources, grid structure, and power consumption patterns. The rising prevalence of renewable energy sources in newly installed power systems has become a particularly noticeable indicator of this transition, resulting in modifications to the grid's attributes. These advancements have made maintaining steady voltage levels and frequencies a crucial objective in control system design. Frequency and voltage variations can result in a negative impact on integrated loads, reducing their dependability and durability [1]. These variations in voltage and frequency have a direct impact on a system's power losses, as well as its active and reactive power. Even small voltage variations can have a significant effect on reactive power. When the voltage deviates more than the typical

±5% from its rated value, it seriously affects the efficiency and operational life of appliances or other connected power system components [2,3].

In particular, it is possible to effectively reduce the impact by managing voltage changes in a proper way. Voltage control can be used in the generation, transmission, and distribution phases of power networks. Several kinds of reactive power compensation devices can be used in the transmission and distribution phases. Several examples exist in the literature, including tap changers in transformers, different filtering devices, flexible AC transmission systems (FACTSs), etc. Automatic voltage regulators (AVRs) are often used to manage voltage variations at the generating stage as they show high efficiency in controlling voltage on the synchronous generator's generation side. However, adjusting the AVR settings to account for many uncertainties and differences in AVRs, sensors, gains, and other elements is an essential and crucial issue for the proper operation of these devices [4].

Actually, numerous AVR controllers based on the Proportional–Integral–Derivative (PID) have been introduced in [5]. The PID controller, as a preferred solution in industrial applications due to its straightforward structure and the well-understood impact of its parameters on the output of controlled systems, has a prevalent advantage despite some limitations existing [3,6]. Various PID controller designs have been addressed for AVR systems, but traditional PID tuning methods like trial-and-error, Ziegler–Nichols, and root locus often fall short in handling uncertain system parameters and load disturbances [7]. To address these shortcomings, a conventional PID controller was designed for AVR systems, employing several metaheuristic algorithms to determine the best controller parameters [7–10]. These controllers' performance has been assessed and compared with other studies.

Several algorithms in the literature for designing PID controllers include the pattern-searching algorithm (PSA) [11], particle swarm optimization (PSO) [12], artificial bee colony (ABC) [13], tree-seed algorithm (TSA) [14], grasshopper optimization algorithm (GOA) [15], whale optimization algorithm (WOA) [16], improved WOA (IWOA) [17], genetic algorithm (GA) [18], sine–cosine-based algorithm (SCA) [19], symbiotic organism searching (SOS) algorithm [20], salp-swarm-based algorithm (SSA) [21], bacteria-foraging-based optimization algorithm (BFOA) [22], ant–lion optimization (ALO) algorithm [23], differential evolution algorithm (DE) [13], etc. The PID possesses only three tunable parameters represented by the gains of its terms (K_P in the P-term, K_I in the I-term, and K_D in the D-term). The literature shows a weak disturbance rejection capability with high sensitivity to the process uncertainties of conventional PID control methods.

The conventional PID controller has an integer-order manner, but there has been a shift towards Fractional-Order PID (FOPID) controllers. The FOPID-based controllers are more flexible due to the additional included FO operator parameters. Optimization-algorithm-based FOPID AVR controllers have been proposed in the literature, such as the marine predator algorithm (MPA) [4], PSO [24,25], GA [25], ABC [26], Chaotic Ant Swarms (CASs) [27], Multi-objective Extremal Optimizer (MOEO) [28], Cuckoo Searching (CS) [29], SCA [30], Improved NSGA-II [31], salp swarm optimizer (SSO) [32], etc. Similar algorithms have been presented using [33,34] PIDD2 based on PSO [35]. The FOPID has five tunable parameters represented by the gains (K_P, K_I, and K_D) and FO operators (λ in the FO-I term and μ in the FO-D term). The slime mold optimization algorithm has been employed in [36] for optimizing FOPID parameters.

Recent advancements in the FOPID include variable-order modifications [37]. Additionally, state-feedback PID controllers have been developed [5,38], with a robust two-degree-of-freedom (2DOF) state-feedback PI controller [5] that eliminates steady-state errors using integral control, in addition to outperforming the one-degree-of-freedom (1DOF) PI controller. Further enhancements to the 2DOF state-feedback PI controller employed a dynamic weighted state-feedback method [20], offering flexibility under varying system conditions. For the AVR, a distinct robust state-feedback controller was proposed, considering bounded system uncertainties and external disturbances in its design [39]. To tackle AVR system uncertainties, a non-fragile PID controller, optimized with a genetic algorithm, and a model predictive controller using Angle of Arrival (AOA) optimization were also introduced in [40,41], respectively.

Although there has been a significant amount of study conducted on PID controllers, the literature also showcases a wide range of control strategies. The use of a strong controller that combines H_∞ and μ-synthesis techniques is suggested to enhance resilience against uncertainties and disturbances that are parametric and structured in nature [42]. The research work [43] introduced a model reference adaptive control method with a fractional order, which was optimized using a genetic algorithm. In addition, a neural network predictive controller for the AVR was optimized using an imperialist competitive algorithm [44]. In [45], the researchers created an Emotional Deep Learning Programming Controller (EDLPC) for AVR systems. The EDLPC incorporates an Emotional Deep Neural Network (EDNN) structure and an artificial emotional Q-learning algorithm. Furthermore, a recent investigation conducted in [46] focuses on improving AVR systems by employing a deep deterministic policy gradient (DDPG) agent. This strategy prioritizes enhancing the AVR's ability to quickly and effectively adapt to changes in its environment, such as variations in the load and alterations in the parameters, while also ensuring its resilience and stability.

An extensive overview of the several optimization strategies for the AVR control system tuning is shown in Table 1. It draws attention to the wide variety of methods used in the literature to modify AVR controller structures. Each of these algorithms has a different working principle, which determines its effectiveness. The various employed objective functions are summarized in Table 2.

Table 1. Literature review for AVR controllers and employed design algorithms.

Method	Controller TF	Reference	Algorithm	No. of Tunable Parameters
PID	$C(s) = K_P + \frac{K_I}{s} + K_D\ s$	[11]	PSA	3 (K_P, K_I, K_D)
		[12]	PSO	
		[13]	ABC	
		[14]	TSA	
		[15]	GOA	
		[16]	WOA	
		[17]	I-WOA	
		[18]	GA	
		[19]	SCA	
		[20]	SOS	
		[21]	SSA	
		[22]	BFOA	
		[23]	ALO	
FOPID	$C(s) = K_P + \frac{K_I}{s^\lambda} + K_D\ s^\mu$	[4]	MPA	5 $(K_P, K_I, K_D, \lambda, \mu)$
		[24]	PSO	
		[25]	GA	
		[26]	NC-ABC	
		[27]	CAS	
		[28]	MOEO	
		[29]	CS	
		[30]	SCA	
		[31]	NSGA-II	
		[25]	PSO	
		[32]	SSO	
PIDF	$C(s) = K_P + \frac{K_I}{s} + \frac{K_D\ s}{N_f\ s+1}$	[12]	BBO	4 (K_P, K_I, K_D, N_f)
PIDD2	$C(s) = K_P + \frac{K_I}{s} + K_D\ s + K_{DD}\ s^2$	[47]	enAO	4 (K_P, K_I, K_D, K_{DD})
PIDND^{2}N^2	$C(s) = K_P + \frac{K_I}{s} + K_D \frac{N_1 s}{s+N_1} + K_{DD} \frac{(N_2 s)^2}{(s+N_2)^2}$	[48]	b-AOA	6 $(K_P, K_I, K_D, K_{DD} N_1, N_2)$
Proposed	$C(s) = K_t s^{-\frac{1}{n}} + \frac{K_I}{s^\lambda} + K_P + K_D \frac{N_1 s}{s+N_1} + K_{DD} \frac{(N_2 s)^2}{(s+N_2)^2}$	Proposed	GO	9 $(K_P, K_T, K_I, K_D, K_{DD}, n, \lambda, N_1 N_2)$

Table 2. Employed single- and multi-objective cost functions from the literature.

Type	Reference	Cost Function
Single	[26]	$Obj = IAE = \int \lvert e_v \rvert dt$
	[26]	$Obj = ISE = \int e_v^2 dt$
	[32]	$OF = ITAE = \int t \cdot \lvert e_v \rvert dt$
	[25]	$Obj = (\omega_1 \cdot OS)^2 + \omega_2 \cdot T_s^2 + \frac{\omega_3}{dV_{max}^2}$
	[24]	$Obj = \omega_1 \cdot OS + \omega_2 \cdot T_r + \omega_3 \cdot T_s + \omega_4 \cdot E_{ss} + \int (\omega_5 \cdot \lvert e_v \rvert dt + \omega_6 \cdot V_f(t)^2) dt + \frac{\omega_7}{P_m} + \frac{\omega_8}{G_m}$
	[27]	$Obj = (1 - e^{-\beta}) \cdot (OS + E_{ss}) + e^{-\beta} \cdot (T_s - T_r)$
	[25]	$Obj = \omega_1 \cdot OS + \omega_2 \cdot T_s + \omega_3 \cdot E_{ss} + \omega_4 \cdot \int \lvert e_v \rvert dt + \omega_5 \cdot \int u^2 dt$
Multiples	[28]	$Obj_1 = IAE, Obj_2 = 1000 \lvert E_{ss} \rvert, Obj_3 = T_s$
	[31]	$Obj_1 = \omega_{cf}, Obj_2 = P_m$
	[49]	$Obj_1 = ITSE = \int t \cdot e_v^2 dt, Obj_2 = \int t \cdot \Delta u^2(t) dt, Obj_3 = ITSE = \int t \cdot e_{load}^2 dt$

Obj = objective, ISE = integral squared error, IAE = integral absolute error, ITSE = integral time-squared error, ITAE = integral time absolute error, OS = overshoot, T_s = settling time, T_r = rise time, E_{ss} = steady-state error, u = control signal, e_v = error voltage, e_{load} = error signal during load disturbance, G_m = gain margin, P_m = phase margin, ω_{cf} = gain crossover frequency, ω_1–ω_8 = weighting factors, dV_{max} = maximum point of voltage signal derivative.

In this study, a new modified hybrid FO (MHFO) AVR based on the FO tilt integral (FOTI) proportional derivative with a filter double derivative with a filter (PDND^{2}N^2) controller, namely FOTI-PDND^{2}N^2, is proposed. The proposed FOTI-PDND^{2}N^2 controller combines the benefits of using the FOTI controller with the PDND^{2}N^2 for ensuring better dynamic performance and steady-state response and for enhancing controller robustness and flexibility. Moreover, thanks to the authors' knowledge, a new application of the growth optimizer (GO) is proposed in the paper for optimally tuning the controller parameters to obtain better system performance compared to the other controller or metaheuristic methods in the literature. The major contributions of this paper can be summarized as follows:

- A new modified hybrid FO (MHFO) controller is proposed for AVR applications in this paper. The new proposed MHFO AVR method is developed based on the FO tilt integral (FOTI) proportional derivative with a filter double derivative with a filter (PDND^{2}N^2) controller, namely FOTI-PDND^{2}N^2. The newly proposed controller merges the benefits of the FOPID, PIDF, and TID controllers, leading to better performance and enhanced characteristics. The tuning process of the control parameters is made offline, which benefits the power and speed of recent microprocessor technologies.
- The proposed FOTI-PDND^{2}N^2 controller combines the benefits of the FOTI controller with PDND^{2}N^2 for ensuring better dynamic performance and steady-state response, and for enhancing controller robustness and flexibility. Also, the inclusion of filters with derivative terms improves their responses, reduces noise, smooths the control action, and has better stability.
- New practical applications of the recently developed growth optimizer (GO) method is introduced in this paper for optimally optimizing the proposed FOTI-PDND^{2}N^2 controller's parameters in a simultaneous manner. Both the recent GO algorithm's benefits and the associated benefits of the proposed FOTI-PDND^{2}N^2 controller are combined to provide a more robust and wide-ranging, stable AVR control method. Moreover, the GO algorithm guarantees the optimum parameter set together for achieving minimization of the defined objective function.

The remainder of the paper is organized as follows: Section 2 provides the mathematical and structure representation of the AVR system. The proposed MHFO AVR controller is presented in Section 3. The proposed design and optimization algorithm are detailed in Section 4. Section 5 presents the obtained performance evaluation and comparison results. Finally, the paper's conclusions are provided in Section 6.

2. Mathematical Representations of AVR Systems

The main elements of the AVR system include the generator, sensing, the AVR controller, the amplifier, and the excitation system, as shown in Figure 1. The main objective of the AVR controller is the regulation of the generator's output voltage with various load variations and disturbances. The control is achieved through controlling the generator's excitation system based on the error signal fed into the AVR controller.

The AVR system is affected by the connected electrical loads at its terminals. When there is an increase in the connected loads, the AVR terminal voltage V_{out} drops. In accordance, error voltage signal E_v (between measured signal V_m and reference setting V_{ref}) increases in the positive value direction. This, in turn, increases generator excitation, reducing steady-state error voltage till reaching its minimum. In the steady state, generator excitation is preserved constant to maintain stable voltage supply for all of the connected loads, whereas in the load-reduction condition, terminal voltage V_{out} increases, leading to a decrease in the error signal in the negative value direction. In the same way, excitation is decreased till having the minimized steady-state error.

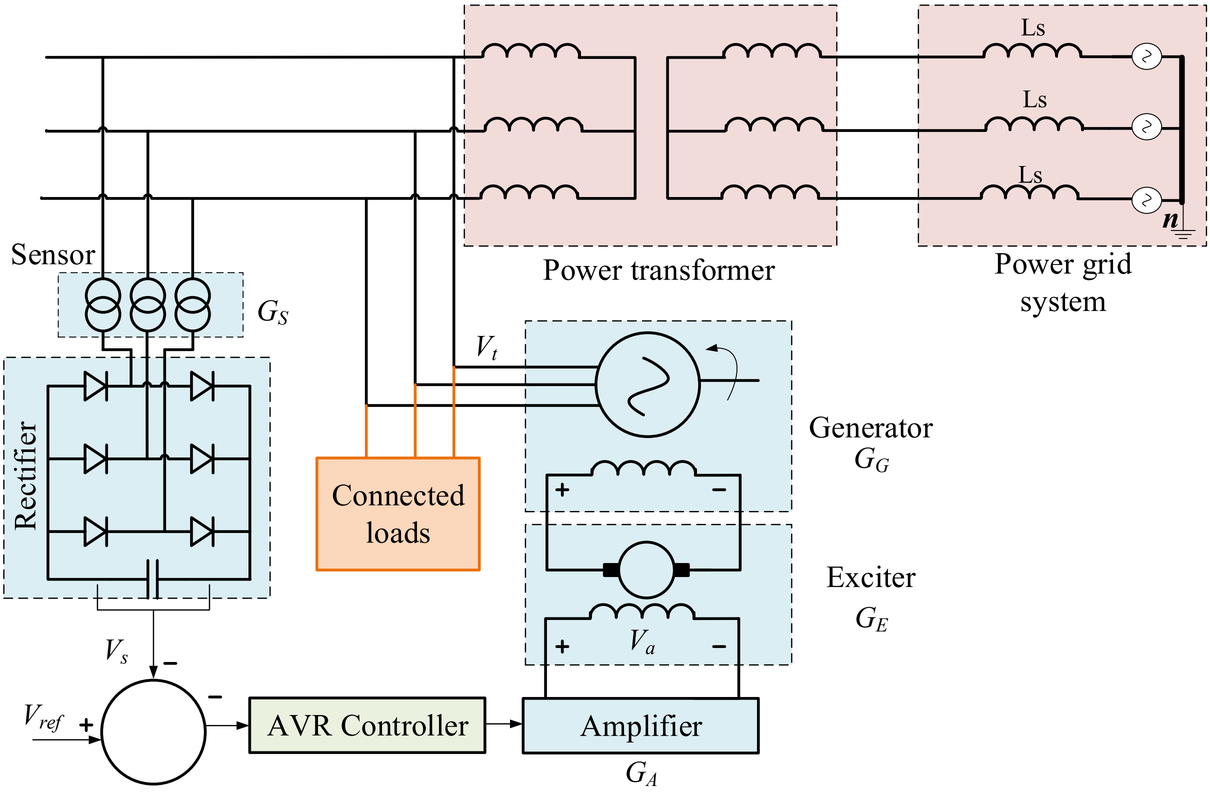


Figure 1. Overall structure of AVR components connected to the grid system.

The AVR system elements' transfer functions (TFs) are normally represented by the Laplace transform of each block. Different elements of the AVR (including the generator, amplifier, sensing system, and exciter) are modeled based on linearized first-order TFs to facilitate the representation process. Their TFs (amplifier $G_A(s)$, generator $G_G(s)$, exciter $G_E(s)$, and sensing system $G_S(s)$) and their associated parameters' range from the literature are as follows [30,50]:

$$\begin{gathered} G_A(s) = \frac{K_A}{1+sT_A}, \text{ with} \\ 10 \leq K_A \leq 400, \quad \text{and} \quad 0.02\,s \leq T_A \leq 0.1\,s \end{gathered} \tag{1}$$

$$G_E(s) = \frac{K_E}{1+sT_E}, \quad \text{with} \\ 1 \le K_E \le 10, \quad \text{and} \quad 0.4\,s \le T_E \le 1\,s \tag{2}$$

$$G_G(s) = \frac{K_G}{1+sT_G}, \quad \text{with} \\ 0.7 \le K_G \le 1, \quad \text{and} \quad 1\,s \le T_G \le 2\,s \tag{3}$$

$$G_S(s) = \frac{K_S}{1+sT_S}, \quad \text{with} \\ 1 \le K_S \le 2, \quad \text{and} \quad 0.001\,s \le T_S \le 0.06\,s \tag{4}$$

where K_A, K_S, K_G, and K_E are gains and T_A, T_S, T_G, and T_E are time constants for the amplifier, voltage sensing, generator, and exciter, respectively. The AVR's complete modeling using first-order TFs for its elements is represented in Figure 2. The voltage error between desired reference V_{ref} (1 p.u., normally) and sensed voltage V_m represents the controller input signal. The AVR controller functionality is to continuously minimize this error, leading to a zero steady-state value in efficient control design.

Based on Figure 2, the complete AVR TF with the controller TF of $C(s)$ is represented by $G_{sys}(s)$. The TF input/output is expressed as:

$$G_{sys}(s) = \frac{C(s)G_A(s)G_E(s)G_G(s)}{1 + C(s)G_A(s)G_E(s)G_G(s)G_S(s)} \tag{5}$$

More details about the characteristics of the AVR system response without the controller can be found in [4,48,51]. The dynamics of the AVR system without the controller (with $C(s)$=1 in (5)) exhibits very low values of the damping ratio for the existing complex poles, which indicates the need for enhancing the uncontrolled AVR system's performance.

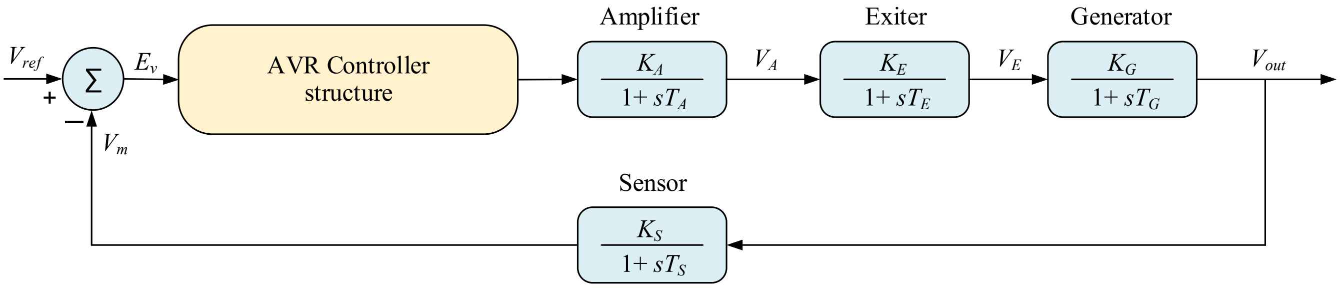


Figure 2. First-order-based TF of AVR elements.

3. Proposed MHFO-AVR Controller

3.1. FOC Modeling and Theory

In various and wide applications in the literature, the FO control (FOC) has proven itself as more flexible with the possibility of a higher degree of control optimization. The inclusion of FO operators in FOC increases the number of tuning parameters of the control systems. This, with proper design, can enhance the stability and response of different processes. In FOC, the general representation $D^\alpha|_a^t$ is categorized as:

$$D^\alpha|_a^t = \begin{cases} \alpha > 0 \; \rightarrow \; \frac{d^\alpha}{dt^\alpha} & \text{FO derivative (FOD)} \\ \alpha < 0 \; \rightarrow \; \int_{t_0}^{t_f} dt^\alpha & \text{FO integrator (FOI)} \\ \alpha = 0 \; \rightarrow \; 1 & \text{Integer Order (IOC)} \end{cases} \tag{6}$$

The principal theories to represent FOC using the FO derivative (FOD) are summarized as follows:

1. Grunwald–Letnikov (GL)-based FOD representation: The α_{th} FOD is represented by a function (f) within [a – t] boundaries as:

$$D^{\alpha}|_{a}^{t} = \lim_{h \to 0} \frac{1}{h^{\alpha}} \sum_{r=0}^{\frac{t-a}{h}} (-1)^{r} \binom{n}{r} f(t-rh) \tag{7}$$

where h refers to the sampling period, and n can be used for fulfilling ($n-1 < \alpha$ and $\alpha < n$). The associated binomials' coefficients can be determined as:

$$\binom{n}{r} = \frac{\Gamma(n+1)}{\Gamma(r+1)\Gamma(n-r+1)'} \tag{8}$$

$$\Gamma(n+1) = \int_{0}^{\infty} t^{x-1} e^{-t}\, dt. \tag{9}$$

2. Riemann–Liouville (RL)-based FOD representation: In the RL-based FOD, summations and bounds are avoided and the IO-based derivative is employed. The FOD is defined as:

$$D^{\alpha}|_{a}^{t} = \frac{1}{\Gamma(n-\alpha)} \left(\frac{d}{dt}\right)^{n} \int_{a}^{t} \frac{f(\tau)}{(t-\tau)^{\alpha-n+1}}\, d\tau \tag{10}$$

3. Caputo-based FOD representation: The FOD based on the Caputo definition is defined as: Another representation of the FO derivative was made by Caputo, and it is defined as follows:

$$D^{\alpha}|_{a}^{t} = \frac{1}{\Gamma(n-\alpha)} \int_{a}^{t} \frac{f^{(n)}(\tau)}{(t-\tau)^{\alpha-n+1}}\, d\tau \tag{11}$$

From the practical implementation and discretization point of view, Oustaloup's recursive approximation (ORA) is the best way and has found several real-time implementation. It can be programmed easily using digital control platforms, leading to simplifying its use and widening its industrial applications. Moreover, it represents a suitable and familiar way for tuning the procedures of the optimum control design. Accordingly, ORA is focused on and employed in this work due to its dominance. In the ORA method, the αth FOD (s^{α}) is defined as:

$$s^{\alpha} \approx \omega_{h}^{\alpha} \prod_{k=-N}^{N} \frac{s+\omega_{k}^{z}}{s+\omega_{k}^{p}} \tag{12}$$

where ω_{k}^{p} refers to the poles and ω_{k}^{z} refers to the zeros within ω_{h}. Their mathematical definitions are expressed as follows:

$$\omega_{k}^{z} = \omega_{b}\left(\frac{\omega_{h}}{\omega_{b}}\right)^{\frac{k+N+\frac{1-\alpha}{2}}{2N+1}} \tag{13}$$

$$\omega_{k}^{p} = \omega_{b}\left(\frac{\omega_{h}}{\omega_{b}}\right)^{\frac{k+N+\frac{1+\alpha}{2}}{2N+1}} \tag{14}$$

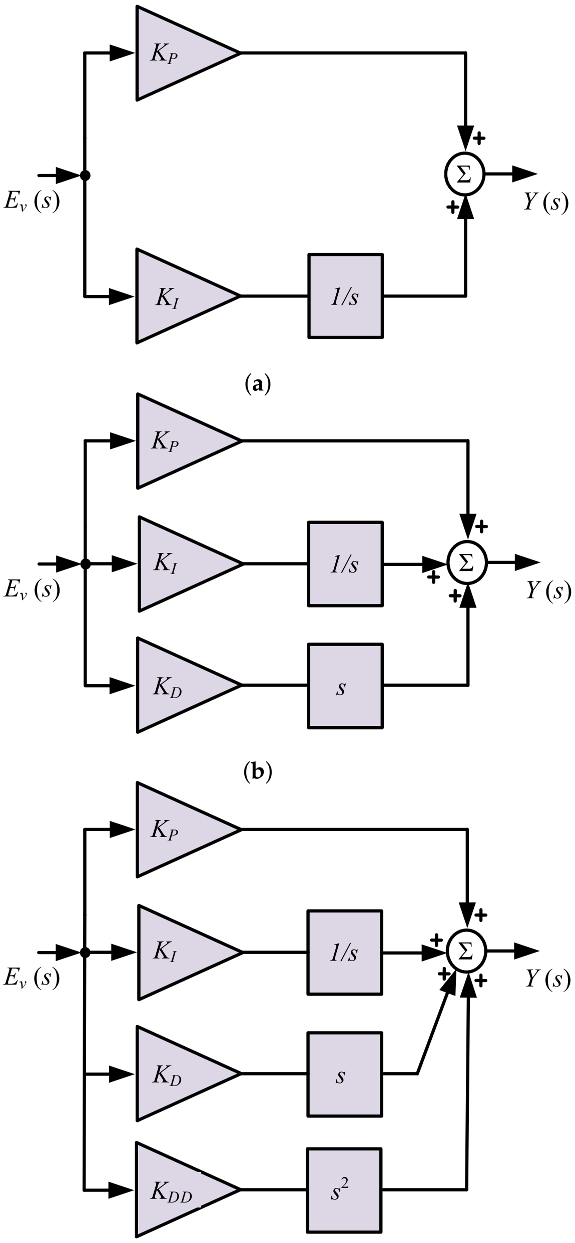
$$\omega_{h}^{\alpha} = \left(\frac{\omega_{h}}{\omega_{b}}\right)^{\frac{-\alpha}{2}} \prod_{k=-N}^{N} \frac{\omega_{k}^{p}}{\omega_{k}^{z}} \tag{15}$$

in which this approximated representation possesses $(2N+1)$ poles/zeros, whereas N defines the order of the ORA's filter within $(2N+1)$. The ORA representation in this work is based on using $(N=5)$ within ($\omega \in [\omega_{\mathrm{b}}, \omega_{\mathrm{h}}]$), and is set within the $[10^{-3}, 10^{3}]$ rad/s range.

3.2. Some Related AVR Methods

Generally, the IOC methods based on the PI and PID have found wide application in several industrial processes and the AVR as well. The PI-based IOC is shown in Figure 3a, and its TF is as follows:

$$C(s) = \frac{Y(s)}{E(s)} = K_P + \frac{K_I}{s} \tag{16}$$



(**a**)

(**b**)

(**c**)

Figure 3. Block diagrams for some of the existing IOC methods. (**a**) IOC based on PI control. (**b**) IOC based on PID control. (**c**) IOC based on PIDD control.

On the other hand, the IOC based on the PID controller is shown in Figure 3b, and it can be expressed as follows:

$$C(s) = \frac{Y(s)}{E(s)} = K_P + \frac{K_I}{s} + K_D\ s \tag{17}$$

in which K_P, K_I, and K_D refer to the proportional (P-term), integral (I-term), and differential (D-term) gains in the IOC based on the PID controller. The IOC PID method represents a simple structure and easily implementable controller. However, the IOC PID method loses its high performance with disturbances. Also, the PID possesses only three tunable parameters in its design. Thence, wide concerns and the focus are targeted at developing more robust, more flexible, and intelligent control methods for AVR applications. Another PID with the double derivative is shown in Figure 3c, and it is represented as follows:

$$C(s) = \frac{Y(s)}{E(s)} = K_P + \frac{K_I}{s} + K_D\ s + K_{DD}\ s^2 \tag{18}$$

The alternative and general method is using FOC methods with the extra added FO operators, leading to more flexibility with a higher number of parameters to tune. The FOC-based PID (FOPID) structure is widely used and has become more common. Figure 4a presents the FOPID block diagram with the FOI and FOD terms. It is expressed as:

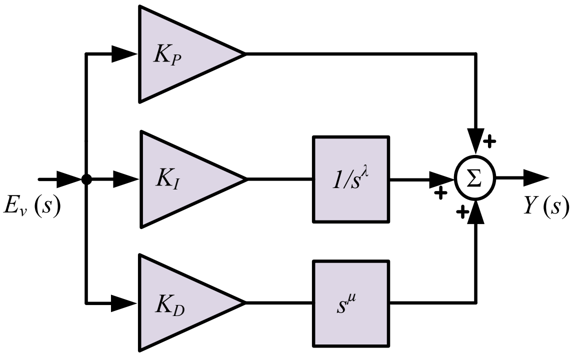
$$C(s) = \frac{Y(s)}{E(s)} = K_P + \frac{K_I}{s^\lambda} + K_D\ s^\mu \tag{19}$$

where λ and μ refer to the FOI operator and FOD operator, respectively. In AVR applications, λ and μ can be tuned within the range [0, 2]. It can be seen that extra flexibility with better performance are obtained through using FOC methods. Also, the FOPID has shown in the literature a wide ability to deal with existing disturbances. The FOPID control is capable of simultaneously handling multiple objectives at wide dynamical operating ranges compared with their IOC-based counterparts. Another FOC based on the tilt integral—derivative (TID) control method has been presented. Figure 4b presents the TID block diagram, and it is mathematically represented as follows:

$$C(s) = \frac{Y(s)}{E(s)} = K_T\ s^{-\left(\frac{1}{n}\right)} + \frac{K_I}{s} + K_D\ s \tag{20}$$

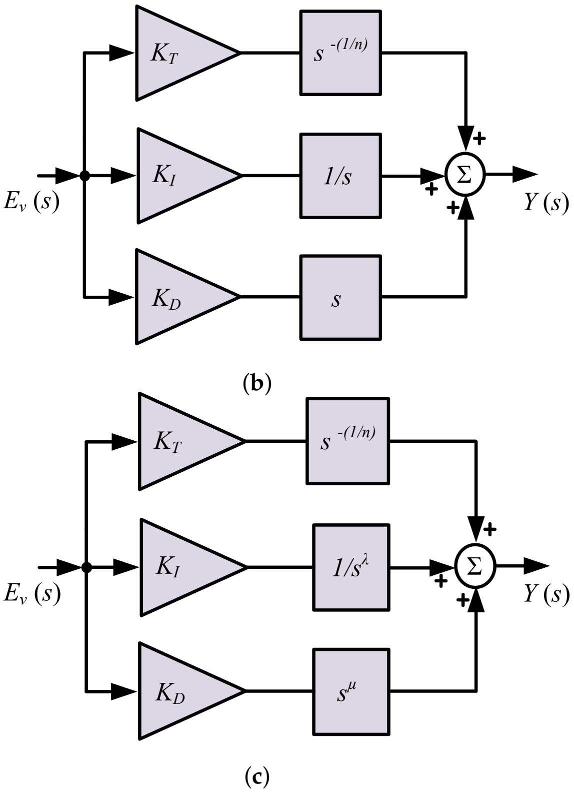
where K_T represents the tilt gain and n refers to the tilt component's FO operator. The inclusion of n presents a simpler tuning process, enhancing the disturbance rejection ability and improving the system robustness against the parameters' uncertainties. A hybrid FOPID with the TID is presented, named the FOTID, as shown in Figure 4c. Its TF is represented as follows:

$$C(s) = \frac{Y(s)}{E(s)} = K_T\ s^{-\left(\frac{1}{n}\right)} + \frac{K_I}{s^\lambda} + K_D\ s^\mu \tag{21}$$



(**a**)

Figure 4. *Cont.*



(**b**)

(**c**)

Figure 4. Block diagrams for some of the existing FOC methods. (**a**) FOC based on FOPID control. (**b**) FOC based on TID control. (**c**) FOC based on FOTID control.

3.3. The Proposed MHFO AVR Controller

The proposed AVR control method is based on a modified hybrid FO (MHFO) controller for regulating the voltage. The proposed MHFO AVR controller combines the advantages and features of IOC methods with FOC methods to provide a new modified structure. It employs the FOC integral (FOI) and FOC tilt (FOT) from the FOTID control method in the first part. In addition, it employs the IOC proportional (P)–derivative with a filter (DN), and the double derivative with a filter (D^2N^2). Hence, a modified structure with five branches is proposed with the FOT, FOI, P, DN, and D^2N^2 terms, forming a new MHFO (FOTI-PDND^{2}N^2) controller. The hybridization of the IOC with FOC enhances the system robustness and stability, in addition to increasing the controller flexibility. Also, the number of tunable control parameters is increased from 5 to 9 parameters in the case of the FOPID compared to the proposed FOTI-PDND^{2}N^2 controller. This, in turn, leads to increased system capability to reject disturbances and keep the system stable even with parameter uncertainty.

Therefore, the proposed FOTI-PDND^{2}N^2 controller combines the benefits and the features from the IOC and FOC methods. It can be mathematically expressed as follows:

$$\begin{aligned} C(s) = \frac{Y(s)}{E(s)} &= \left(K_t s^{-\frac{1}{n}} + \frac{K_I}{s^{\lambda}} \right) \\ &+ \left(K_P + K_D \frac{N_1 s}{s + N_1} + K_{DD} \frac{(N_2 s)^2}{(s + N_2)^2} \right) \end{aligned} \tag{22}$$

The block diagram for the proposed FOTI-PDND^{2}N^2 controller is shown in Figure 5. It can be seen that the proposed FOTI-PDND^{2}N^2 controller has 5 different branches compared to the 3 branches in the FOPID and TID controllers. In addition, the proposed structure has 9 tunable control parameters compared with the 5 parameters in the FOPID and the 4 parameters in the TID controller. Therefore, the proposed FOTI-PDND^{2}N^2 controller provides better flexibility with a higher degree of freedom due to having more parameters to tune. The increased parameters enable providing better control robustness and performance. In addition, proper parameter tuning is necessary for optimizing the proposed FOTI-PDND^{2}N^2 controller's performance. Recently developed metaheuristic optimizers have proven to be easier and accurate ways for tuning different control methods in a wide variety of applications. The control parameters can be optimized and determined simultaneously using optimization algorithms based on the set objective function for the optimization problem. In this work, the recent powerful growth optimizer (GO) is presented for a new implementation in determining the control parameters for AVR applications.

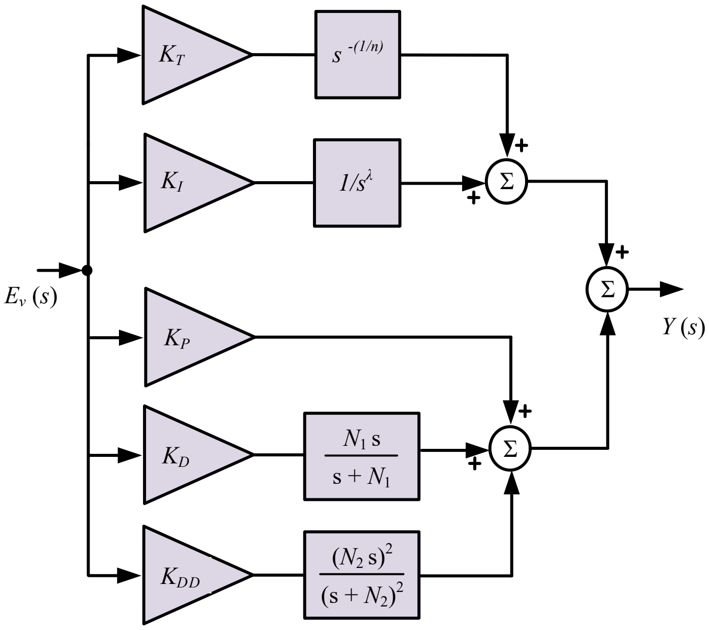


Figure 5. The proposed MHFO AVR controller based on new modified FOTI-PDND^{2}N^2 control method.

4. Optimum Design of Proposed MHFO-AVR Controller

4.1. Growth Optimization Algorithm

The growth optimization algorithm (GO) is a metaheuristic algorithm for optimizing processes [52]. It is mainly inspired by the learning performed by individuals and its reflection mechanisms on their growth in the society. Thence, the GO algorithm is composed of two main phases: learning-based phase and reflection-based phase. The learning-based phase is the first stage of the process, in which individual persons use their knowledge about people's differences in practice, whereas, in the reflection-based phase, individual persons use different techniques for identifying and correcting their shortcomings in the learning process [52].

The solutions in the GO algorithm for a certain problem are called individuals [52], whereas decision variables are represented by necessary elements for individuals, such as emotions, morality, beliefs, perseverance, cultivation, etc. A society or a population with a certain number of individuals is represented by a set of decision variables as the matrix. For the ith individual with $i \in \{1, 2, 3, \ldots, \mathrm{N}\}$, within the search space, is represented by $x_i \in \{x_{i,1}, x_{i,2}, \ldots, x_{i,D}\}$, where $x_{i,D}$ represents the D^{th} element of the i^{th} individual. The speed of individuals' growth in the GO algorithm is defined according to the growth resistance (GR). In general, the objective function of the optimization process receives the i^{th} individual, then it returns its corresponding output, represented by GR_i of the i^{th} individual. With lower GR of the individual, it absorbs more knowledge, and hence, it is possible to be an elite member in the society. In the GO algorithm, population x_i representing the problem solution is generated as [52]:

$$X_i = r \times (U - L) + L \qquad , \; i = 1,2,\ldots,N \tag{23}$$

where r stands for a random value and U and L stand for the search domain's limits of the optimization problem, whereas N stands for the solutions' total number within x_i. In GO, x_i is split into three different parts based on setting parameter P_1, with P_1 = 5 based on [52]. In the first part, the leader and elites are compromised between 2 and P_1. In the second part, the middle level from $P_1 + 1$ to $N - P_1$ is included, whereas, the bottom level from $N - P_1 + 1$ to N is contained. The best solution among the individuals is represented by the leader of the upper level.

4.1.1. Learning Phase

The progress of the individuals is greatly enhanced through confronting disparities among individual people, examining the causes behind their differences, and learning from them. The learning phase of the GO simulates four different key gaps, which that are formulated as [53]:

$$\begin{aligned} G_1 &= X_b - X_{bt} \\ G_2 &= X_b - X_w \\ G_3 &= X_{bt} - X_w \\ G_4 &= X_{r1} - X_{r2} \end{aligned} \tag{24}$$

where X_b, X_{bt}, and X_w refer to the best, better, and worst solutions, respectively. Moreover, X_{r1} and X_{r2} refer to two randoms solutions. G_k (with $k \in \{1, 2, 3, 4\}$) denotes the employed gap to improve the learned skills of individuals and to decrease the differences among them. In addition, the learning factor (LF) represents a parameter to be applied to reflect the variations among groups. The parameter LF is formulated as follows [54]:

$$LF_k = \frac{||G_k||}{\sum_{k=1}^{4} ||G_k||} \tag{25}$$

Based on [52], each individual assesses the learned knowledge through parameter (SF_i), which is represented as [52]:

$$SF_i = \frac{GR_i}{GR_{max}} \tag{26}$$

where GR_{max} and GR_i are the maximum GR of X and the growth of individual X_i, respectively. Based on the collected information from LF_k and SF_i, new knowledge can be received for every X_i from the solution related to each gap G_k using knowledge acquisition (KA_k), which is determined as [53]:

$$KA_k = SF_i \times LF_k \times G_k \qquad , \ k = 1,2,3,4 \tag{27}$$

Afterwards, the solution X_i can enhance its information through using the following [54]:

$$X_i(t+1) = X_i(t) + \sum_{k=1}^{4} KA_k \tag{28}$$

The quality of the updated X_i value is calculated and compared with the last value to define if they are significantly different. The value of $X_i(t+1)$ is determined as [53]:

$$X_i(t+1) = \begin{cases} X_i(t+1) & \text{if } f_i^{t+1} < f_i^t \\ \begin{cases} X_i(t+1) & \text{if } r_1 < P_2 \text{ and} \\ & ind(i) = ind(1) \\ X_i(t) & else \end{cases} & otherwise \end{cases} \tag{29}$$

where r_1 represents a random number and P_2 represents the probability of retention (P_2 = 0.001), whereas $ind(i)$ represents the X_i ranking in ascending order using the fitness value.

4.1.2. Reflection Phase

As explained earlier, the GO algorithm is based on the learning and reflection phases. Thence, individuals have to learn how to reflect instead on only learning. Thence, individuals have to check and identify all their weakness areas. In addition, a systematic learning process is whenever understanding particular issues cannot be solved. They have to learn from their outstanding individuals to repair their bad issues. In addition, they should retain and continue their good aspects. Accordingly, the reflective process in the GO algorithm is mathematically represented as follows [52]:

$$X_i(t+1) = \begin{cases} X_m(t) & \text{if } r_2 < P_3 \\ X_i(t) & Otherwise \end{cases} \tag{30}$$

where $X_m(t)$ is represented as follows [54]:

$$X_m(t) = \begin{cases} r_4 \times (U - L) & \text{if } r_3 < AF \\ X_i(t) + r_5 \times (X_R - X_i(t)) & else \end{cases} \tag{31}$$

$$AF = 0.01 + 0.99 \times (1 - \frac{FE_s}{mac_{FE}}) \tag{32}$$

where r_3, r_4, and r_5 stand for random variables. X_R stands for the defined solution by the top $P_1 + 1$ solutions within X, whereas AF stands for the attenuation factor, which relies on the evaluated function FE and the total number for the function evaluations max_{FE}. After a complete reflection phase, X_i have to determine its growth as in the learning phase. Thence, (29) can be employed for this evaluation.

4.2. Application to Optimum Design of Proposed MHFO AVR Controller

A schematic diagram representing the optimization process of the proposed MHFO AVR controller is shown in Figure 6. In addition, the main procedures for the entire operation of the GO algorithm are shown in Figure 7. Firstly, the system is modeled and the controller is connected to the AVR system, enabling adjusting the parameters through the m-file by the optimization algorithm to search for the optimum values. Secondly, the optimization algorithm is run, and in every iteration, the objective function is calculated and compared with the previous global optimum objective function. The objective function is updated when there are smaller values in the current iteration. Finally, when the maximum number of iterations is reached, the final optimum control parameters with the optimum objective function and convergence curves are output and used for the AVR system's simulation and evaluations.

On the other hand, the performance of the AVR system is highly determined according to the employed objective function type. The objective function is responsible for driving the optimization process. As shown in the literature, there are several objective functions for single and multiple objectives. In general, the combination of different tunable control parameters is designed and optimized in a way to continuously minimize the set objective function. The measurement of the output voltage with the reference voltage is employed to form the error-based objective function. In the literature, there are four principal error e_i-based objective functions, as follows:

1. Integral-squared error (ISE):

$$ISE = \int \sum_{i=1}^{m} (e_i^2) \, \mathrm{d}t \tag{33}$$

2. Integral time-squared error (ITSE):

$$ITSE = \int \sum_{i=1}^{m} (e_i^2) \, t.\mathrm{d}t \tag{34}$$

3. Integral absolute error (IAE):

$$IAE = \int \sum_{i=1}^{m} abs(e_i)\ \mathrm{d}t \tag{35}$$

4. Integral time absolute error (ITAE):

$$ITAE = \int \sum_{i=1}^{m} abs(e_i)\ t.\mathrm{d}t \tag{36}$$

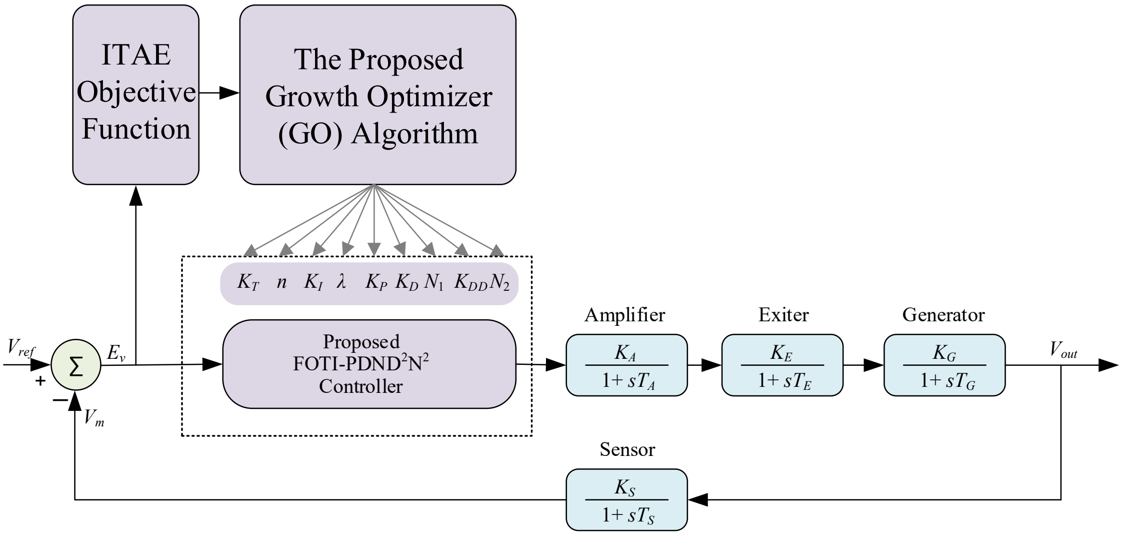


Figure 6. Schematic diagram of GO-based optimization of proposed MHFO AVR controller.

In this work, the ITAE objective function is selected for optimizing the proposed MHFO AVR controller. The error voltage signal e_V is utilized for evaluating the objective function in each iteration of the GO algorithm. The ITAE is preferred in this paper as it provides a better control response in AVR system. The ITAE integrates the absolute error, which is suitable for the case of the AVR system with error less than 1. Moreover, it makes the integration of time, which leads to having zero steady-state error. The process of the searching mechanism within the search space is determined by the GO optimizer, and finally, a set of the best nine parameters to minimize e_V is output from the algorithm. The parameter search space boundaries are set as the problem constraint as follows:

$$\begin{gathered} K_T^{min} \le K_T \le K_T^{max} \\ K_I^{min} \le K_I \le K_I^{max} \\ K_P^{min} \le K_P \le K_P^{max} \\ K_D^{min} \le K_D \le K_D^{max} \\ K_{DD}^{min} \le K_{DD} \le K_{DD}^{max} \\ n^{min} \le n \le n^{max} \\ \lambda^{min} \le \lambda \le \lambda^{max} \\ N_1^{min} \le N_1 \le N_1^{max} \\ N_2^{min} \le N_2 \le N_2^{max} \end{gathered} \tag{37}$$

in which $(f)^{min}$ and $(f)^{max}$ refer to the lower/upper limiting boundaries of each parameter, respectively. The parameters K_T^{min}, K_I^{min}, K_P^{min}, K_D^{min}, and K_{DD}^{min} are set at 0.0, and K_T^{max}, K_I^{max}, K_P^{max}, K_D^{max}, and K_{DD}^{max} are set at 3.0. Also, λ^{min} is set at 1, whereas λ^{max} is set at 2 within the GO-based algorithm. The values for n^{min} and n^{max} are set at 2 and 10, respectively. The filter coefficients N_1^{min}, and N_2^{min} are set at 50, whereas N_1^{max} is set at 500 and N_2^{max} is set at 1000 in the proposed optimization procedures.

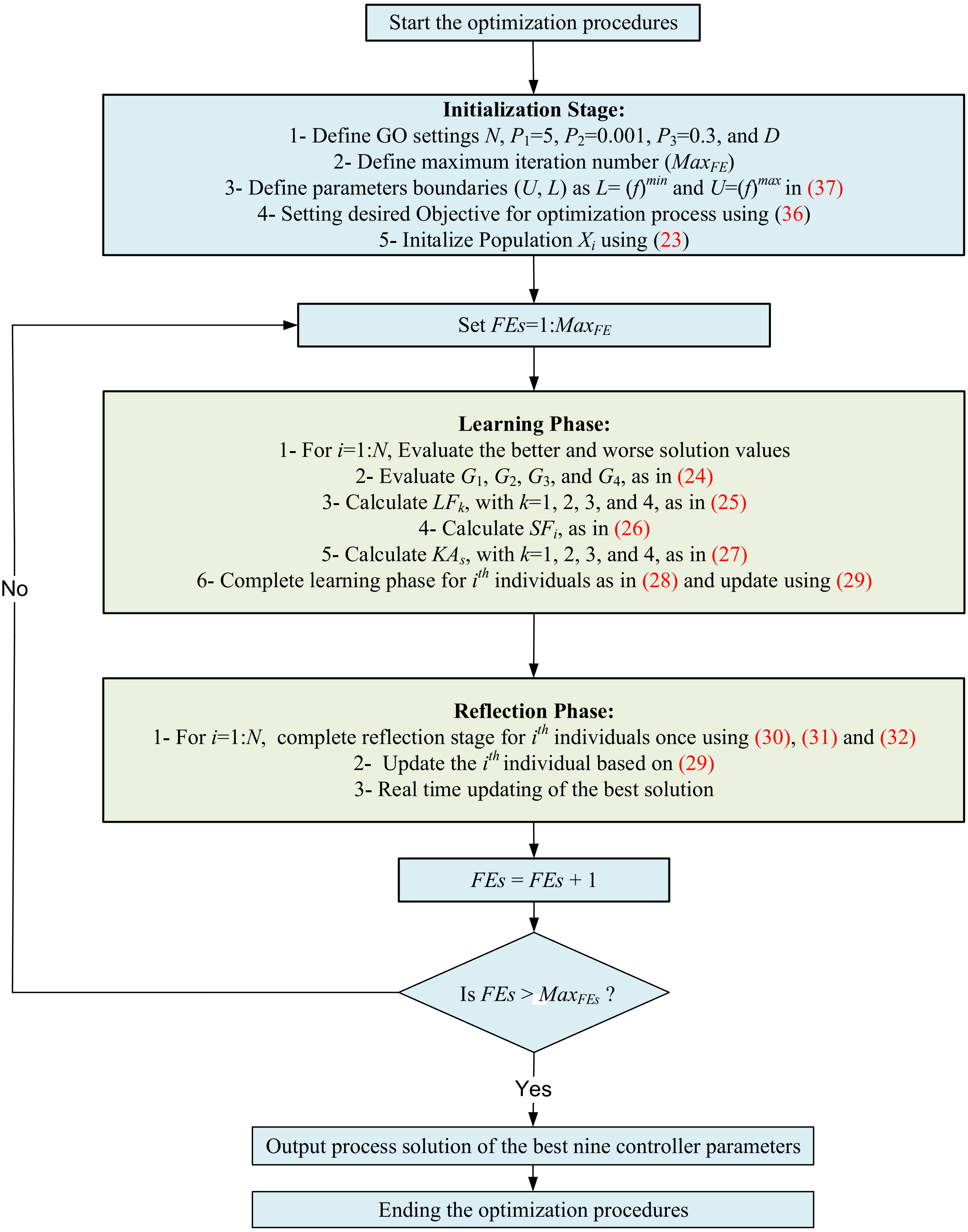

Start the optimization procedures
Initialization Stage:
1- Define GO settings N, P_1=5, P_2=0.001, P_3=0.3, and D
2- Define maximum iteration number (Max_{FE})
3- Define parameters boundaries (U, L) as $L= (f)^{min}$ and $U=(f)^{max}$ in (37)
4- Setting desired Objective for optimization process using (36)
5- Initalize Population X_i using (23)
Set FEs=1:Max_{FE}
Learning Phase:
1- For i=1:N, Evaluate the better and worse solution values
2- Evaluate G_1, G_2, G_3, and G_4, as in (24)
3- Calculate LF_k, with k=1, 2, 3, and 4, as in (25)
4- Calculate SF_i, as in (26)
5- Calculate KA_s, with k=1, 2, 3, and 4, as in (27)
6- Complete learning phase for i^{th} individuals as in (28) and update using (29)
Reflection Phase:
1- For i=1:N, complete reflection stage for i^{th} individuals once using (30), (31) and (32)
2- Update the i^{th} individual based on (29)
3- Real time updating of the best solution
FEs = FEs + 1
Is FEs > Max_{FEs} ?
No
Yes
Output process solution of the best nine controller parameters
Ending the optimization procedures

Figure 7. Main phases and entire operation of GO algorithm.

5. Simulation Results

This part discusses the performance testing of the proposed FOTI-PDND^{2}N^2 control scheme in controlling the AVR system by investigating its performance with different loading situations such as full load, no load, and multi-step load perturbations (MLP) against the uncertainties of the AVR system parameters. Moreover, the performance validation of the proposed controller is examined by comparing it with the conventional PID controller, which is tuned by the differential evolution (DE) technique as method (A). It is further compared to method (B) of the FOPID-based salp swarm algorithm (SSA), method (C) of PIDD^2-based particle swarm optimization (PSO), method (D) of the FOPID-based Manta Ray-Foraging Optimization (MRFO), and method (E) of the FOPID-based marine predator optimization algorithm (MPA) control methods. The optimized controller parameters for every method are shown in Table 3. Moreover, the discrete implementation realization of the proposed FOTI-PDND^{2}N^2 controller is tested utilizing the MATLAB *SIMULINK R2022a* software, which is interfaced with the programming GO M-file code to select the optimal parameters of the proposed controller using the integral time absolute error (ITAE) objective function via a personal computer with an Intel® i7 2.7GHz processor and 16 GB of RAM. The optimization process of the proposed GO has been executed using 50 iterations and 30 populations and compared with the MPA, MRFO, SSA, PSO, and DE, as depicted in the convergence curve of Figure 8, which proves the best and flat performance of GO against the other processes. The tested comparison scenarios are organized as follows:

Table 3. Controllers parameters.

Method	Controller Optimizer	K_P	K_t	K_i	K_{d1}	K_{d2}	n	λ	μ	N_1	N_2
A	PID-DE	1.9499	-	0.4430	0.3427	-	-	-	-	-	-
B	FOPID-SSA	1.9982	-	1.1706	0.5749	-	-	1.14	1.17	-	-
C	PIDD2-PSO	2.7784	-	1.8521	0.9997	0.0739	-	-	-	-	-
D	FOPID-MRFO	1.6506	-	0.7878	0.3932	-	-	1.21	1.21	-	-
E	FOPID-MPA	1.7061	-	0.8068	0.4	-	-	1.13	1.22	-	-
Proposed	MHFO-GO	2.5105	2.9582	2.7086	1.9033	0.1558	7.19	1.12	-	92.23	375.2

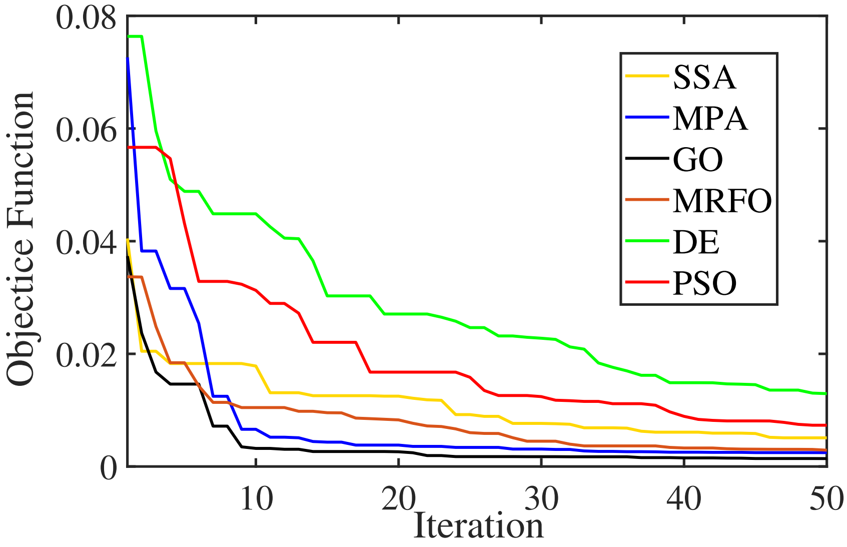


Figure 8. Convergence curve comparisons of GO algorithm with the literature.

5.1. Scenario 1: Full-Load Condition

The performance of the proposed method to enhance the AVR system output is examined under a step change in the reference voltage at the initial instant of the simulation time and the full-load condition at ($K_G = 1$) in this scenario. Figure 9 shows the voltage response of the AVR system in this case. It is shown in this figure that the proposed controller-based GO offers remarkable performance with a 1.039 p.u peak value compared to 1.053 p.u and 1.061 p.u with method (D) and method (E), respectively, while the voltage peak reaches to about 1.15 p.u when using the method (C) controller and 1.23 p.u with the method (B) controller. Despite method (A) having a peak of only 1.02 p.u, it does not settle down and is fixed at this value, whereas the methods (A), (B), (D), and (E) have settling time values of 3.5 s, 2.7 s, 1.1 s, and 1.05 s, respectively. However, the proposed method has the fastest settling time with t_s = 0.088 s and a rise time of t_r = 0.032 s. Table 4 summarizes all numerical results of this scenario, which proves that the proposed method has robust performance compared to other AVR controllers in terms of t_s, t_r, t_p, and M_p. The obtained results clarify the effectiveness of the second-order derivative and filter parts via the performance of the proposed controller.

Table 4. AVR time domain specifications of the proposed controller.

Scenario	Case	Peak Value (p.u.)	$T_P(s)$	$T_r(s)$	$T_s(s)$
(1)	Full Load $(K_g = 1)$	1.0545	0.0630	0.0293	0.0879
(2)	No Load $(K_g = 0.7)$	1.0007	0.3580	0.0514	0.0892
(3)	MLP	1.1004	8.0124	0.0293	8.0424

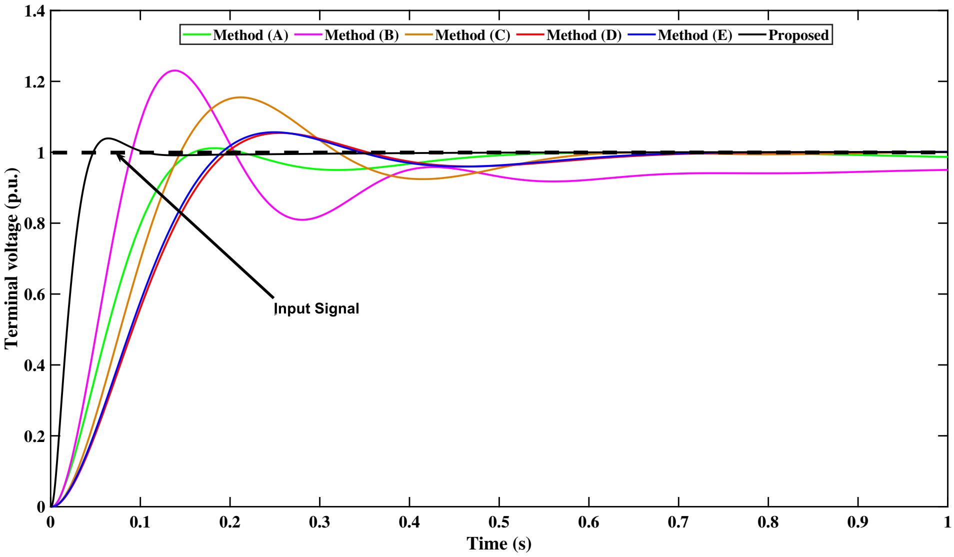


Figure 9. The AVR output in the full-load scenario with a step change in the reference voltage.

5.2. Scenario 2: No-Load Condition

This scenario investigates the AVR system performance under the same conditions as scenario 1 with the no-load case of (k_G = 0.7). Figure 10 shows the output voltage of the AVR system for different control techniques. It is clear that method (A) suffers from a steady-state error as the output voltage cannot reach the reference value, while method (B) exhibits the highest peak voltage value as it exceeds 1.1 p.u and its settling time to reach the reference value is greater than 3 s. Also, method (C) has a peak voltage at 1.054 p.u with a settling time of around 1.1 s. In addition, method (D) and method (E) do not exceed 0.992 p.u of the voltage; they take a settling time of around 1.15 s to reach the reference value. However, the proposed method with the double derivative and filter exhibits a reduced overshoot peak value to 0.998 p.u and a fast tracking of 0.51 s in this severe no-load step change scenario. Therefore, the proposed method has much better levels in terms of t_s, t_r, t_p, and M_p than the compared controllers, as tabulated in Table 4.

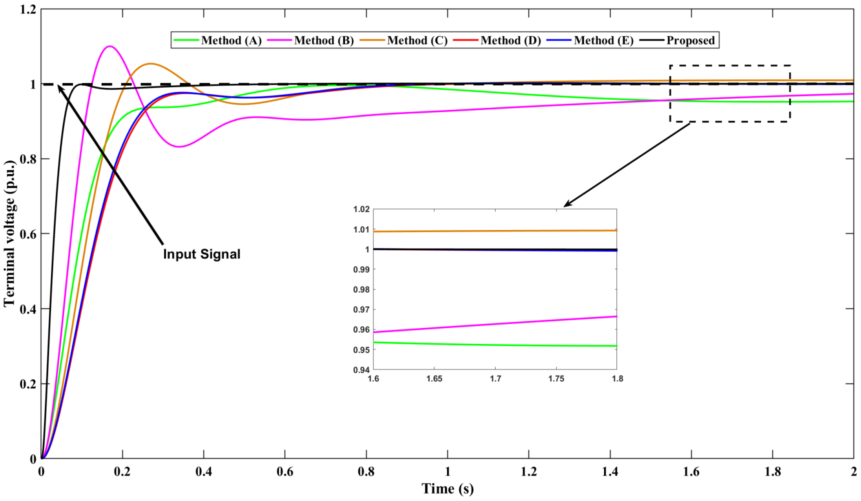


Figure 10. The AVR output in the no-load scenario with a step change in the reference voltage.

5.3. Scenario 3: Multi-Step Load Condition

The capability of the proposed method is investigated during this scenario against the load disturbance rejection, which represents an additional robustness criterion for the AVR output control. Therefore, the AVR system has been subjected to a load disturbance of ±10% of the rated terminal voltage through the interval of 10 s, as shown in Figure 11. Furthermore, this figure shows the impact of the load action on the output voltage considering different types of control techniques. It can be observed that the proposed method can maintain the peak value within ±5% of the allowable limits through the five steps of the load injection/rejection process. Method (A) has a lower peak value than the other methods, but with a fixed high value of the steady-state error until the end of the simulation time, while method (B) has a peak voltage overshoot of more than 10% of the reference value, particularly at the first step change accompanied by the steady-state error value. Method (C) exhibits a proper performance with reduced settling time (around 0.7 s) and decreased overshoot around (±10%). Methods (D) and (E) give a satisfactory performance with the settling time around 0.2 s and ±10% overshoot. From these results, it is obvious that the proposed method using MHFO-GO demonstrates reliable performance against load disturbance injections and rejections. The proposed method generates rapid and precise control inputs to maintain the AVR system stability at the rated value and within acceptable values of t_s, t_r, t_p, and M_p, as listed in Table 4.

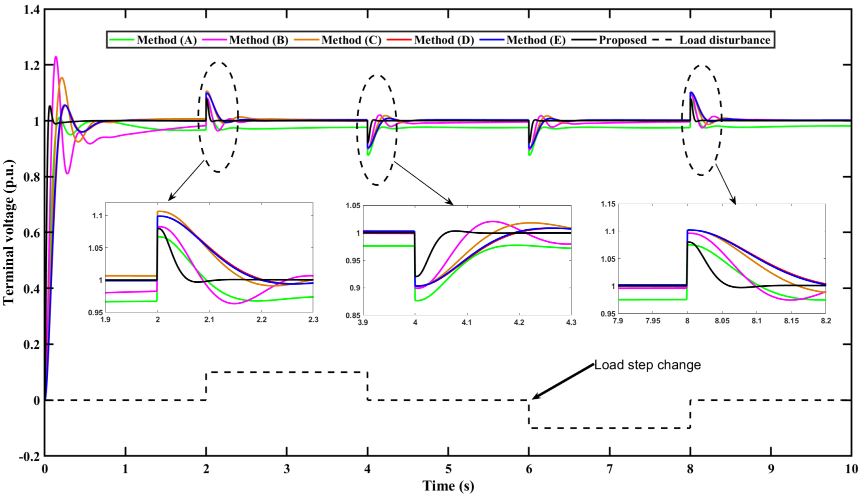


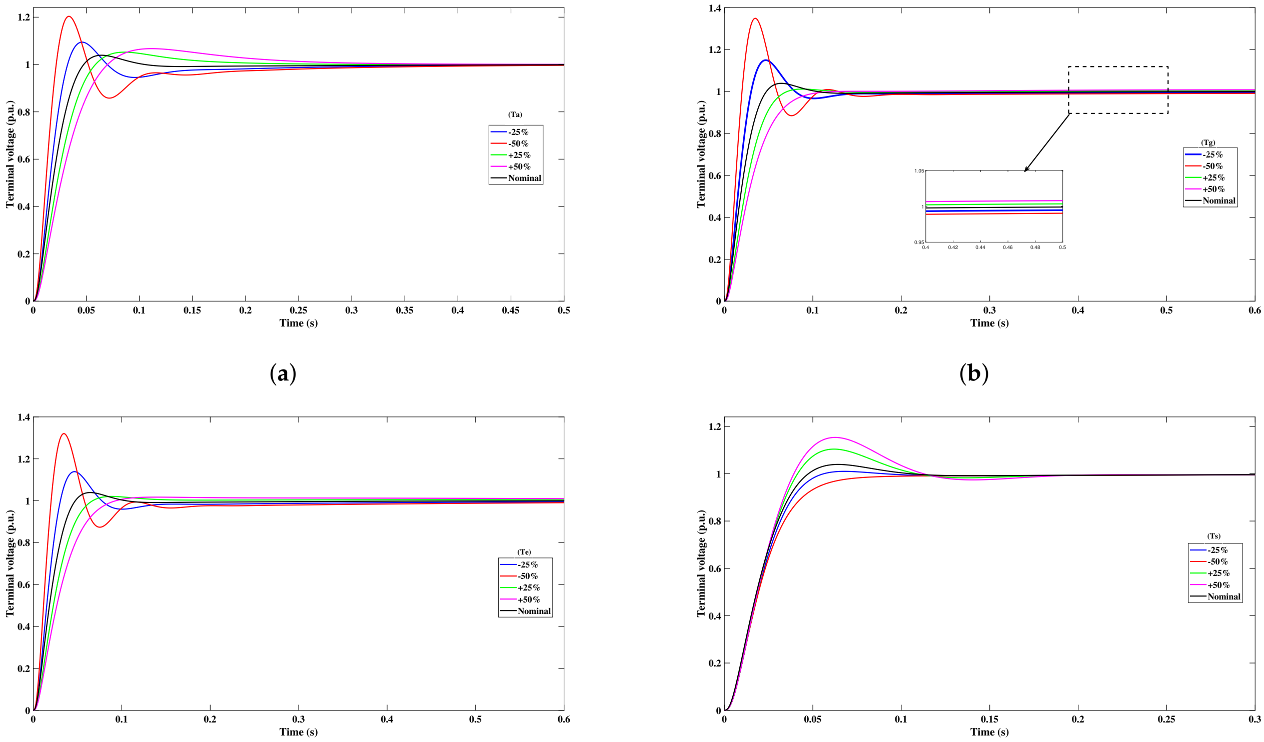
Figure 11. The AVR output in the multi-step load condition scenario.

5.4. Scenario 4: Sensitivity Analysis

Generally, load fluctuations and external factors may cause changes in the AVR system's design specifications. Therefore, it is important to investigate AVR system performance with parameter changes. To illustrate the behavior of the terminal voltage (v_t), the AVR parameters (t_a, t_e, t_g, and t_s) are allowed to be changed by both ±25% and ±50%. An evaluation of the proposed controller's efficacy under both the normal operating and modified system parameter scenarios is developed in this study as well. The performance parameters of the AVR system (i.e., peak value, t_s, t_r, and t_p) that are controlled by the proposed controller in accordance with the adjusted time-constant requirements are listed in Table 5. It is seen in the table that changes in T_A, T_E, and T_G increase the system overshoot by 20%, 25%, and 37% with respect to the nominal scenario, while decreasing T_S by 50% eliminates system overshoots without affecting the settling time, while increasing T_A by 50% decreases the system overshoot to 8% and extends the settling time to be 250% compared to the nominal scenario. On the other hand, increasing T_G and T_E by 50% accelerates the system response while eliminating system overshoots and extending the system settling time to be 0.45 s and 0.2 s, respectively. On the contrary, increasing T_S by 25% and 50% increases the system overshoots to 10% and 17%, respectively. The output of the AVR system step response when the T_A, T_G, T_E, and T_S parameters change is shown in Figure 12a–d. It is seen that the proposed controller has the capability and efficacy to maintain system stability even with parameter changes. On the other hand, the proposed controller has been compared with other control techniques under the parameter uncertainties to validate its effect on the AVR's performance. This can be observed from Figures 13–16, which show the system performance with the AVR parameter uncertainty with the different controllers in the literature. All the results show that the proposed controller has the best voltage stability result for the AVR system compared to all suggested methods under ±25% of all time constant values of the AVR system.

Table 5. AVR time domain specifications for scenario 4.

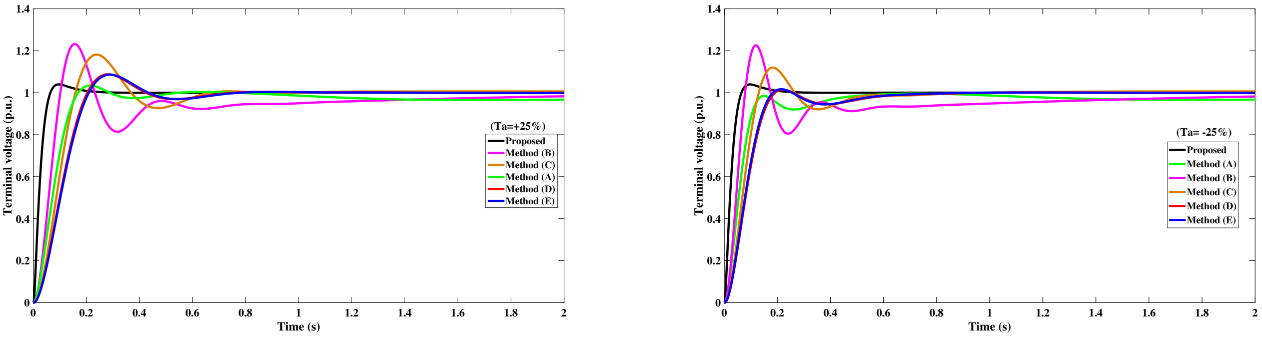
Parameters	Percentage Change	Peak Value (p.u.)	$T_P(s)$	$T_r(s)$	$T_s(s)$
Nominal	100%	1.0545	0.0630	0.0293	0.0879
T_A	+50%	1.0670	0.1110	0.0458	0.2214
	+25%	1.0521	0.0860	0.0377	0.1440
	−25%	1.0945	0.0460	0.0213	0.1844
	−50%	1.2035	0.0340	0.0142	0.2459
T_G	+50%	1.0084	0.6320	0.0542	0.0899
	+25%	1.0117	0.0880	0.0408	0.0628
	−25%	1.1498	0.0470	0.0202	0.1242
	−50%	1.3495	0.0350	0.0131	0.1706
T_E	+50%	1.0172	0.1390	0.0521	0.0814
	+25%	1.0206	0.0890	0.0401	0.0940
	−25%	1.1388	0.0470	0.0204	0.1361
	−50%	1.3203	0.0350	0.0133	0.2989
T_S	+50%	1.1532	0.0630	0.0268	0.1609
	+25%	1.1039	0.0620	0.0278	0.0980
	−25%	1.0099	0.0680	0.0321	0.0498
	−50%	1.0005	0.0580	0.0372	0.0722



(**a**) (**b**)

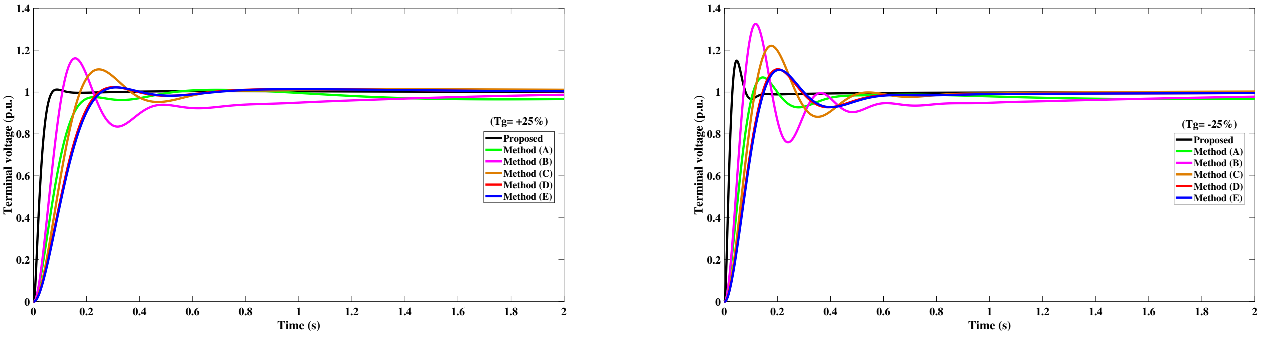
(**c**) (**d**)

Figure 12. AVR parameter uncertainty with the proposed controller. (**a**) Uncertainty in T_A. (**b**) Uncertainty in T_G. (**c**) Uncertainty in T_E. (**d**) Uncertainty in T_S.



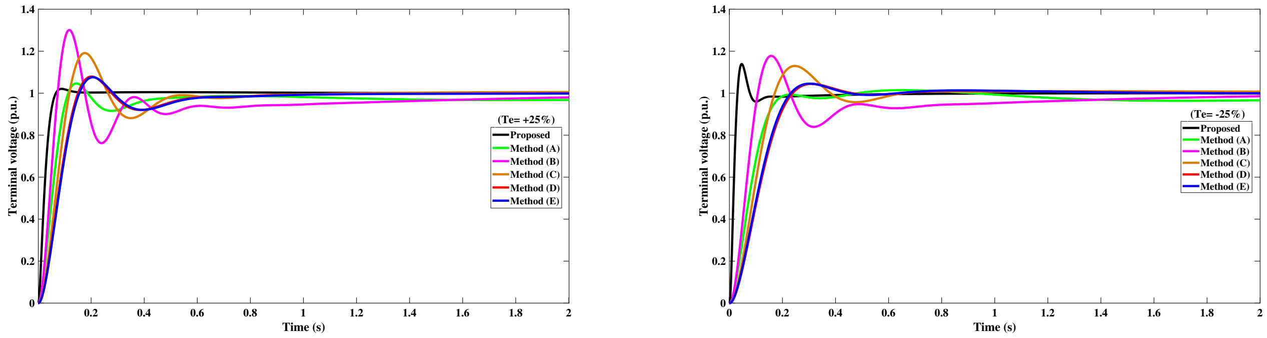
(**a**) (**b**)

Figure 13. Impact of changing T_A: (**a**) +25% T_A; (**b**) −25% T_A.



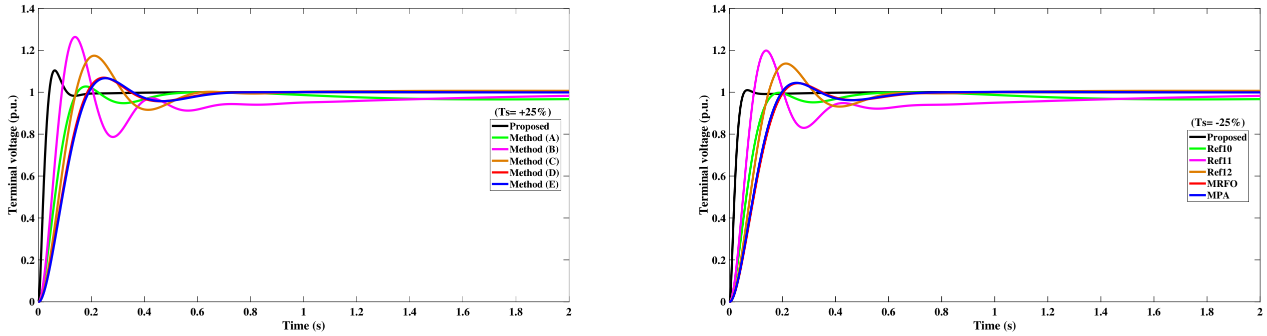
(**a**) (**b**)

Figure 14. Impact of changing T_G: (**a**) +25% T_G; (**b**) −25% T_G.



(**a**) (**b**)

Figure 15. Impact of changing T_E: (**a**) +25% T_E; (**b**) -25% T_E.



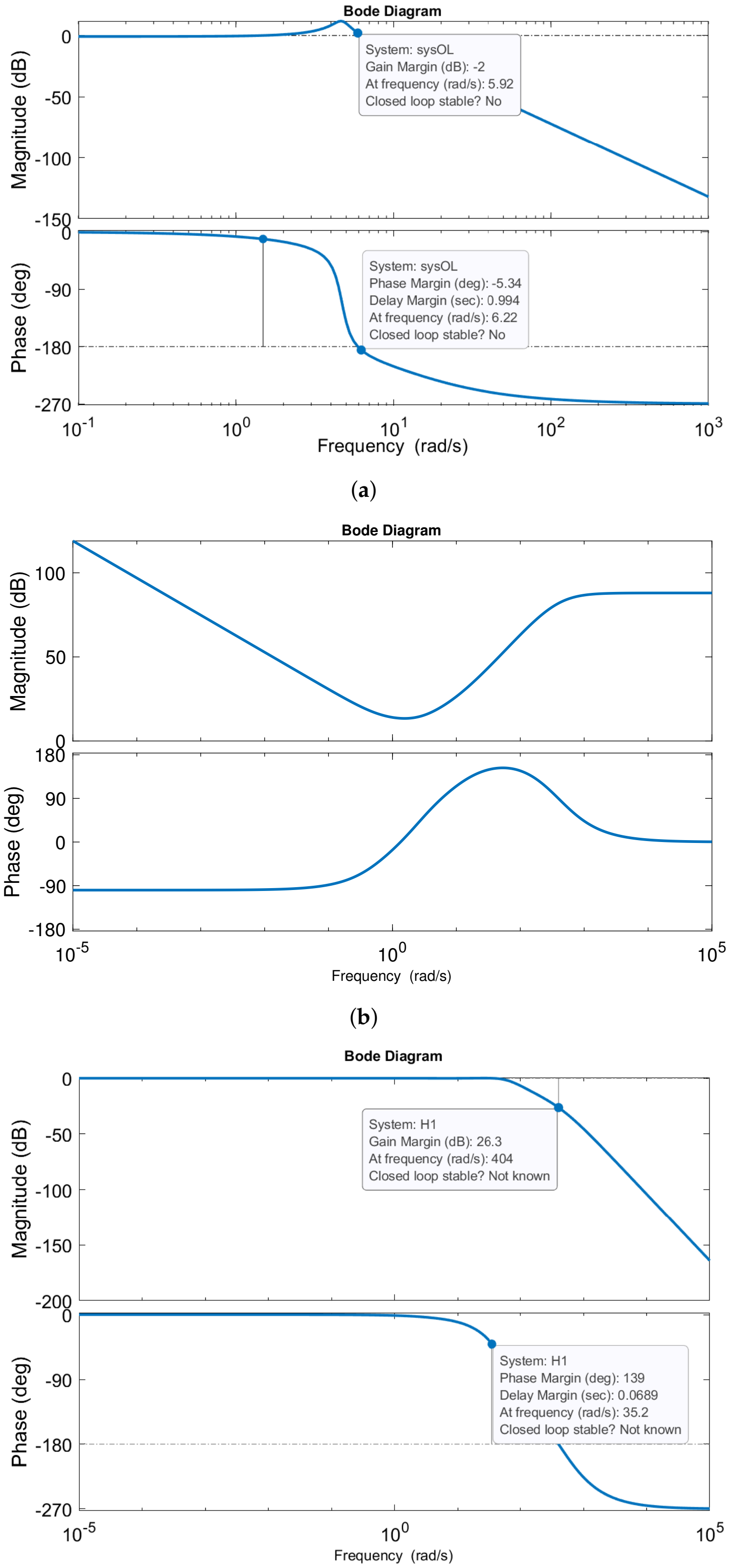
(**a**) (**b**)

Figure 16. Impact of changing T_S: (**a**) +25% T_S; (**b**) -25% T_S.

5.5. Scenario 5: Frequency Domain Analysis

This scenario presents the Bode analysis of the AVR system to demonstrate the realization of the proposed FOTI-PDND^{2}N^2 control structure in the frequency domain. The Bode plot, shown in Figure 17a for the open-loop AVR system, reveals that the open-loop system is marginally stable, as noticed by a negative gain margin of -2 dB and a negative phase margin of -5.34 degrees. The observed margins indicate that the system's amplification is already excessively high prior to reaching the phase crossover point, and the phase delay is severe, beyond the -180-degree threshold before the amplification decreases to 0 dB. Therefore, the system is susceptible to oscillations and instability. Moreover, the delay margin of 0.094 s indicates a vulnerability to further instability when more time delays are introduced. In another context, this figure clearly indicates that the open-loop system is unstable due to the negative margins shown. Therefore, the current state of this system requires the implementation of stabilizing techniques, such as feedback compensation, in order to rectify its reaction and guarantee optimal performance.

The proposed controller gain magnitude and phase compensation are shown in Figure 17b. The Bode plot of the closed-loop system equipped with the proposed controller is shown in Figure 17c. It shows that the closed-loop gain margin is a substantial 26.3 dB, which indicates that the system can tolerate a significant increase in gain before reaching instability. The phase margin is also very large at 139 degrees, which suggests that the system can withstand a considerable amount of additional phase lag without becoming unstable. Therefore, the proposed controller guarantees the AVR system's stability with a wide bandwidth, which enables the AVR system to accurately address the system parameter uncertainty and random disturbances as well.



(**a**)

(**b**)

(**c**)

Figure 17. Bode plots of AVR system and controller. (**a**) Open-loop-only response. (**b**) Controller-only response. (**c**) Closed-loop complete system response.

5.6. Scenario 6: Frequency Domain Performance Comparisons

A comprehensive analysis was conducted to compare the stability indices of the MHFO-GO (FOTI-PDND^{2}N^2) AVR controller proposed in this study with various methods found in the existing literature. The study examined the gain margin (GM), phase margin (PM), and controller bandwidth (BW) of the loop transfer function for both the AVR and the proposed controller. Table 6 presents a summary of the results obtained and the evaluation of the proposed FOTI-PDND^{2}N^2 controller. It is evident from the table that the proposed method exhibits a PM of 64.3°, surpassing the performance of PID controllers based on the DE, the FOPID controllers based on the SSA, the PID controllers based on the SCA, the FOPID controllers based on the MRFO, and the FOPID controllers based on the SMA methods. Furthermore, while the proposed FOTI-PDND^{2}N^2 controller demonstrates a slightly lower PM compared to the PIDD^2 controllers based on PSO, the PIDA controllers based on the WOA, and the PIDND^{2}N^2 controllers based on the AOA, it offers a significantly better bandwidth than these methods and all others examined in the literature. Additionally, the GM of the proposed method proves to be notably superior to that of the majority of the studied methods. These results collectively demonstrate the superior stability performance of the newly proposed method.

Table 6. Frequency domain response performance comparisons

Reference	Controller	Phase Margin PM (°)	Gain Margin GM (dB)	Bandwidth BW (rad/s)
[13]	PID based on DE	36.1	371.5	12.8
[21]	FOPID based on SSA	51.5	Inf.	21.3
[19]	PID based on SCA	52.6	20.3	14.8
[4]	FOPID based on MRFO	62.9	Inf.	16.7
[35]	PIDD2 based on PSO	79.6	Inf.	23.5
[36]	FOPID based on SMA	49.1	20.2	22.9
[16]	PIDA based on WOA	67.7	26.1	6.7
[48]	PIDND2N2 based on AOA	69.8	23.4	57.8
Proposed	MHFO-GO (FOTI-PDND^{2}N^2)	64.3	26.8	62.4

6. Conclusions

In this paper, a novel MHFO-GO controller including the fractional tilt integral combined with the proportional and first-/second-order derivative filter controller is constructed to improve the system control ability of the AVR. Moreover, the recent GO algorithm utilizing the integral of time multiplied by the absolute error (ITAE) performance criterion has been interfaced with the AVR *SIMULINK* system to optimize the nine control parameters of the proposed MHFO-GO (FOTI-PDND^{2}N^2) controller. The validation of the proposed GO algorithm was tested under 50 iterations and 30 populations with a comparison with the MPA, MRFO, SSA, PSO, and DE. In addition, the AVR performance based on the proposed controller was quantitatively compared with several conventional and fractional PID controllers in the literature. It can be revealed from the obtained analytical results that the proposed MHFO-GO controller achieves a superior response performance against the steady-state error, multi-step load disturbances, and system uncertainties and, hence, preserves the AVR system stability. This is due to the merits of the proposed combination of the FOTI and PDND^{2}N^2. Therefore, the proposed controller provides the best ITAE minimization with less settling time, rise time, and percentage maximum peak compared to other traditional and fractional PID control methods. Future work suggestions include making some online control parameter adjustment, presenting simplified design methods for the optimum parameter selection process, and performing further detailed stability analysis and comparisons.

Author Contributions: Conceptualization, E.A.M., M.A. and E.M.A.; methodology, E.A.M., M.A., W.A. and E.M.A.; software, E.A.M., M.A., W.A. and E.M.A.; validation, E.A.M. and E.M.A.; formal analysis, E.A.M. and M.A.; investigation, E.A.M., M.A. and W.A.; resources, E.A.M., W.A. and E.M.A.; data curation, E.A.M., M.A. and W.A.; writing—original draft preparation, E.A.M., M.A. and W.A.; writing—review and editing, E.A.M., M.A. and W.A.; visualization, W.A. and E.M.A.; supervision, E.M.A.; project administration, M.A., W.A. and E.M.A.; funding acquisition, W.A., E.M.A. and M.A. All authors have read and agreed to the published version of the manuscript.

Funding: The authors extend their appreciation to the Deputyship for Research & Innovation, the Ministry of Education in Saudi Arabia for funding this research work through project number 223202.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Said, S.M.; Aly, M.; Hartmann, B.; Mohamed, E.A. Coordinated fuzzy logic-based virtual inertia controller and frequency relay scheme for reliable operation of low-inertia power system. *IET Renew. Power Gener.* **2021**, *15*, 1286–1300. [CrossRef]
2. Amin, A.; Ebeed, M.; Nasrat, L.; Aly, M.; Ahmed, E.M.; Mohamed, E.A.; Alnuman, H.H.; Hamed, A.M.A.E. Techno-Economic Evaluation of Optimal Integration of PV Based DG with DSTATCOM Functionality with Solar Irradiance and Loading Variations. *Mathematics* **2022**, *10*, 2543. [CrossRef]
3. Furat, M.; Cücü, G.G. Design, Implementation, and Optimization of Sliding Mode Controller for Automatic Voltage Regulator System. *IEEE Access* **2022**, *10*, 55650–55674. [CrossRef]
4. Noman, A.M.; Almutairi, S.Z.; Aly, M.; Alqahtani, M.H.; Aljumah, A.S.; Mohamed, E.A. A Marine-Predator-Algorithm-Based Optimum FOPID Controller for Enhancing the Stability and Transient Response of Automatic Voltage Regulators. *Fractal Fract.* **2023**, *7*, 690. [CrossRef]
5. Ekinci, S.; Hekimoğlu, B. Improved Kidney-Inspired Algorithm Approach for Tuning of PID Controller in AVR System. *IEEE Access* **2019**, *7*, 39935–39947. [CrossRef]
6. Modabbernia, M.; Alizadeh, B.; Sahab, A.; Moghaddam, M.M. Robust control of automatic voltage regulator (AVR) with real structured parametric uncertainties based on H_∞ and μ-analysis. *ISA Trans.* **2020**, *100*, 46–62. [CrossRef]
7. Daraz, A.; Malik, S.A.; Basit, A.; Aslam, S.; Zhang, G. Modified FOPID Controller for Frequency Regulation of a Hybrid Interconnected System of Conventional and Renewable Energy Sources. *Fractal Fract.* **2023**, *7*, 89. [CrossRef]
8. Alilou, M.; Azami, H.; Oshnoei, A.; Mohammadi-Ivatloo, B.; Teodorescu, R. Fractional-Order Control Techniques for Renewable Energy and Energy-Storage-Integrated Power Systems: A Review. *Fractal Fract.* **2023**, *7*, 391. [CrossRef]
9. Alghamdi, S.; Sindi, H.F.; Rawa, M.; Alhussainy, A.A.; Calasan, M.; Micev, M.; Ali, Z.M.; Aleem, S.H.E.A. Optimal PID Controllers for AVR Systems Using Hybrid Simulated Annealing and Gorilla Troops Optimization. *Fractal Fract.* **2022**, *6*, 682. [CrossRef]
10. Ghasemi, M.; Rahimnejad, A.; Gil, M.; Akbari, E.; Gadsden, S.A. A self-competitive mutation strategy for Differential Evolution algorithms with applications to Proportional–Integral–Derivative controllers and Automatic Voltage Regulator systems. *Decis. Anal. J.* **2023**, *7*, 100205. [CrossRef]
11. Sahu, B.K.; Panda, S.; Mohanty, P.K.; Mishra, N. Robust analysis and design of PID controlled AVR system using Pattern Search algorithm. In Proceedings of the 2012 IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES), Bengaluru, India, 16–19 December 2012; pp. 1–6. [CrossRef]
12. Gaing, Z.L. A particle swarm optimization approach for optimum design of PID controller in AVR system. *IEEE Trans. Energy Convers.* **2004**, *19*, 384–391. [CrossRef]
13. Gozde, H.; Taplamacioglu, M. Comparative performance analysis of artificial bee colony algorithm for automatic voltage regulator (AVR) system. *J. Frankl. Inst.* **2011**, *348*, 1927–1946. [CrossRef]
14. Köse, E. Optimal Control of AVR System With Tree Seed Algorithm-Based PID Controller. *IEEE Access* **2020**, *8*, 89457–89467. [CrossRef]
15. Hekimoglu, B.; Ekinci, S. Grasshopper optimization algorithm for automatic voltage regulator system. In Proceedings of the 2018 5th International Conference on Electrical and Electronic Engineering (ICEEE), Istanbul, Turkey, 3–5 May 2018. [CrossRef]
16. Mosaad, A.M.; Attia, M.A.; Abdelaziz, A.Y. Whale optimization algorithm to tune PID and PIDA controllers on AVR system. *Ain Shams Eng. J.* **2019**, *10*, 755–767. [CrossRef]
17. Habib, S.; Abbas, G.; Jumani, T.A.; Bhutto, A.A.; Mirsaeidi, S.; Ahmed, E.M. Improved Whale Optimization Algorithm for Transient Response, Robustness, and Stability Enhancement of an Automatic Voltage Regulator System. *Energies* **2022**, *15*, 5037. [CrossRef]
18. Dogruer, T.; Can, M.S. Design and robustness analysis of fuzzy PID controller for automatic voltage regulator system using genetic algorithm. *Trans. Inst. Meas. Control* **2022**, *44*, 1862–1873. [CrossRef]
19. Hekimoğlu, B. Sine-cosine algorithm-based optimization for automatic voltage regulator system. *Trans. Inst. Meas. Control* **2018**, *41*, 1761–1771. [CrossRef]

20. Çelik, E.; Durgut, R. Performance enhancement of automatic voltage regulator by modified cost function and symbiotic organisms search algorithm. *Eng. Sci. Technol. Int. J.* **2018**, *21*, 1104–1111. [CrossRef]
21. Ekinci, S.; Hekimoğlu, B.; Kaya, S. Tuning of PID Controller for AVR System Using Salp Swarm Algorithm. In Proceedings of the 2018 International Conference on Artificial Intelligence and Data Processing (IDAP), Malatya, Turkey, 28–30 September 2018; pp. 1–6. [CrossRef]
22. Anbarasi, S.; Muralidharan, S. Enhancing the Transient Performances and Stability of AVR System with BFOA Tuned PID Controller. *J. Control Eng. Appl. Inform.* **2016**, *18*, 20–29.
23. Pradhan, R.; Majhi, S.K.; Pati, B.B. Design of PID controller for automatic voltage regulator system using Ant Lion Optimizer. *World J. Eng.* **2018**, *15*, 373–387. [CrossRef]
24. Zamani, M.; Karimi-Ghartemani, M.; Sadati, N.; Parniani, M. Design of a fractional order PID controller for an AVR using particle swarm optimization. *Control Eng. Pract.* **2009**, *17*, 1380–1387. [CrossRef]
25. Ortiz-Quisbert, M.E.; Duarte-Mermoud, M.A.; Milla, F.; Castro-Linares, R.; Lefranc, G. Optimal fractional order adaptive controllers for AVR applications. *Electr. Eng.* **2016**, *100*, 267–283. [CrossRef]
26. ZHANG, D.L.; TANG, Y.G.; GUAN, X.P. Optimum Design of Fractional Order PID Controller for an AVR System Using an Improved Artificial Bee Colony Algorithm. *Acta Autom. Sin.* **2014**, *40*, 973–979. [CrossRef]
27. Tang, Y.; Cui, M.; Hua, C.; Li, L.; Yang, Y. Optimum design of fractional order PIλDμ controller for AVR system using chaotic ant swarm. *Expert Syst. Appl.* **2012**, *39*, 6887–6896. [CrossRef]
28. Zeng, G.Q.; Chen, J.; Dai, Y.X.; Li, L.M.; Zheng, C.W.; Chen, M.R. Design of fractional order PID controller for automatic regulator voltage system based on multi-objective extremal optimization. *Neurocomputing* **2015**, *160*, 173–184. [CrossRef]
29. Sikander, A.; Thakur, P.; Bansal, R.; Rajasekar, S. A novel technique to design cuckoo search based FOPID controller for AVR in power systems. *Comput. Electr. Eng.* **2018**, *70*, 261–274. [CrossRef]
30. Ayas, M.S.; Sahin, E. FOPID controller with fractional filter for an automatic voltage regulator. *Comput. Electr. Eng.* **2021**, *90*, 106895. [CrossRef]
31. Pan, I.; Das, S. Frequency domain design of fractional order PID controller for AVR system using chaotic multi-objective optimization. *Int. J. Electr. Power Energy Syst.* **2013**, *51*, 106–118. [CrossRef]
32. Khan, I.A.; Alghamdi, A.S.; Jumani, T.A.; Alamgir, A.; Awan, A.B.; Khidrani, A. Salp Swarm Optimization Algorithm-Based Fractional Order PID Controller for Dynamic Response and Stability Enhancement of an Automatic Voltage Regulator System. *Electronics* **2019**, *8*, 1472. [CrossRef]
33. Shayeghi, H.; Younesi, A.; Hashemi, Y. Optimal design of a robust discrete parallel FP+FI+FD controller for the Automatic Voltage Regulator system. *Int. J. Electr. Power Energy Syst.* **2015**, *67*, 66–75. [CrossRef]
34. Mohd Tumari, M.Z.; Ahmad, M.A.; Mohd Rashid, M.I. A fractional order PID tuning tool for automatic voltage regulator using marine predators algorithm. *Energy Rep.* **2023**, *9*, 416–421. [CrossRef]
35. Sahib, M.A. A novel optimal PID plus second order derivative controller for AVR system. *Eng. Sci. Technol. Int. J.* **2015**, *18*, 194–206. [CrossRef]
36. Izci, D.; Ekinci, S.; Zeynelgil, H.L.; Hedley, J. Fractional Order PID Design based on Novel Improved Slime Mould Algorithm. *Electr. Power Components Syst.* **2021**, *49*, 901–918. [CrossRef]
37. Oziablo, P.; Mozyrska, D.; Wyrwas, M. Fractional-variable-order digital controller design tuned with the chaotic yellow saddle goatfish algorithm for the AVR system. *ISA Trans.* **2022**, *125*, 260–267. [CrossRef] [PubMed]
38. Eke, I.; Saka, M.; Gozde, H.; Arya, Y.; Taplamacioglu, M.C. Heuristic optimization based dynamic weighted state feedback approach for 2DOF PI-controller in automatic voltage regulator. *Eng. Sci. Technol. Int. J.* **2021**, *24*, 899–910. [CrossRef]
39. Ali Hussien Mary, A.H.M.; Miry, M.H. An Optimal Robust State Feedback Controller for the AVR System-Based Harris Hawks Optimization Algorithm. *Electr. Power Components Syst.* **2020**, *48*, 1684–1694. [CrossRef]
40. Elsisi, M.; Tran, M.Q.; Hasanien, H.M.; Turky, R.A.; Albalawi, F.; Ghoneim, S.S.M. Robust Model Predictive Control Paradigm for Automatic Voltage Regulators against Uncertainty Based on Optimization Algorithms. *Mathematics* **2021**, *9*, 2885. [CrossRef]
41. Elsisi, M. Optimal design of non-fragile PID controller. *Asian J. Control* **2021**, *23*, 729–738. [CrossRef]
42. Das, S.; Pan, I. On the Mixed H_2/H_∞ Loop-Shaping Tradeoffs in Fractional-Order Control of the AVR System. *IEEE Trans. Ind. Inform.* **2014**, *10*, 1982–1991. [CrossRef]
43. Aguila-Camacho, N.; Duarte-Mermoud, M.A. Fractional adaptive control for an automatic voltage regulator. *ISA Trans.* **2013**, *52*, 807–815. [CrossRef]
44. Elsisi, M. Design of neural network predictive controller based on imperialist competitive algorithm for automatic voltage regulator. *Neural Comput. Appl.* **2019**, *31*, 5017–5027. [CrossRef]
45. Yin, L.; Zhang, C.; Wang, Y.; Gao, F.; Yu, J.; Cheng, L. Emotional Deep Learning Programming Controller for Automatic Voltage Control of Power Systems. *IEEE Access* **2021**, *9*, 31880–31891. [CrossRef]
46. Ayas, M.S.; Sahin, A.K. A reinforcement learning approach to Automatic Voltage Regulator system. *Eng. Appl. Artif. Intell.* **2023**, *121*, 106050. [CrossRef]
47. Ekinci, S.; Izci, D.; Eker, E.; Abualigah, L. An Effective Control Design Approach Based on Novel Enhanced Aquila Optimizer for Automatic Voltage Regulator. *Artif. Intell. Rev.* **2022**, *56*, 1731–1762. [CrossRef]
48. Ekinci, S.; Çetin, H.; Izci, D.; Köse, E. A Novel Balanced Arithmetic Optimization Algorithm-Optimized Controller for Enhanced Voltage Regulation. *Mathematics* **2023**, *11*, 4810. [CrossRef]

49. Pan, I.; Das, S. Chaotic multi-objective optimization based design of fractional order PIλDμ controller in AVR system. *Int. J. Electr. Power Energy Syst.* **2012**, *43*, 393–407. [CrossRef]
50. Bakir, H.; Guvenc, U.; Kahraman, H.T.; Duman, S. Improved Lévy flight distribution algorithm with FDB-based guiding mechanism for AVR system optimal design. *Comput. Ind. Eng.* **2022**, *168*, 108032. [CrossRef]
51. Micev, M.; Ćalasan, M.; Oliva, D. Fractional Order PID Controller Design for an AVR System Using Chaotic Yellow Saddle Goatfish Algorithm. *Mathematics* **2020**, *8*, 1182. [CrossRef]
52. Zhang, Q.; Gao, H.; Zhan, Z.H.; Li, J.; Zhang, H. Growth Optimizer: A powerful metaheuristic algorithm for solving continuous and discrete global optimization problems. *Knowl.-Based Syst.* **2023**, *261*, 110206. [CrossRef]
53. Fatani, A.; Dahou, A.; Abd Elaziz, M.; Al-qaness, M.A.A.; Lu, S.; Alfadhli, S.A.; Alresheedi, S.S. Enhancing Intrusion Detection Systems for IoT and Cloud Environments Using a Growth Optimizer Algorithm and Conventional Neural Networks. *Sensors* **2023**, *23*, 4430. [CrossRef]
54. Aribia, H.B.; El-Rifaie, A.M.; Tolba, M.A.; Shaheen, A.; Moustafa, G.; Elsayed, F.; Elshahed, M. Growth Optimizer for Parameter Identification of Solar Photovoltaic Cells and Modules. *Sustainability* **2023**, *15*, 7896. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.