Design and Optimization of an Intelligent Automatic Voltage Regulator Using the Sine-Generalized Hybrid Optimization Operator

Divyansh Singh

Department of Electrical Engineering, Madhav Institute of Technology & Science, Gwalior, Madhya Pradesh, India

Email: singhdivyansh8520@gmail.com | ORCID: https://orcid.org/0009-0000-5761-7819

Kuldeep Kumar Swarnkar

Assistant Professor, Department of Electrical Engineering, Madhav Institute of Technology & Science, Gwalior, Madhya Pradesh, India

Email: kuldeepkumarsony@mitsgwalior.in | ORCID: https://orcid.org/0000-0002-8446-4130

How to Cite this Article:

Singh, D. (2026). Design and Optimization of an Intelligent Automatic Voltage Regulator Using the Sine-Generalized Hybrid Optimization Operator. International Journal of Creative and Open Research in Engineering and Management, <i>02</i>(05). https://doi.org/10.55041/ijcope.v2i5.491

https://doi.org/ 10.55041/ijcope.v2i5.491

Abstract—
This paper presents a simulation-based approach for tuning automatic voltage regulator (AVR) controllers using a Sine-Generalized Hybrid Optimization Operator (SGHOO). The AVR benchmark plant is modeled by first-order amplifier, exciter, generator, and sensor blocks, and PI, PD, PID, and FOPID controllers are optimized in Python. SGHOO combines the exploration behavior of the Sine Cosine Algorithm with a generalized-normal local search step so that the search can move across the parameter space and then refine promising regions. A weighted fitness function based on rise time, settling time, overshoot, steady-state error, IAE, ITAE, ISE, and ISTE is minimized for each controller. The simulation results show stable closed-loop responses, with the PID controller giving the best overall compromise between speed and error. Compared with the SCA and GNDO baselines, SGHOO improves convergence quality and provides a practical framework for AVR voltage regulation.

Keywords— Automatic voltage regulator, SGHOO, Hybrid metaheuristic optimization, Sine Cosine Algorithm, Generalized Normal Distribution Optimization, PID controller, FOPID controller.

I. INTRODUCTION

Voltage regulation is a fundamental requirement in power systems because the terminal voltage of a synchronous generator must remain within acceptable limits under varying load and disturbance conditions. The automatic voltage regulator (AVR) plays a critical role in maintaining this voltage by adjusting the excitation system, thereby improving power quality, protecting equipment, and enhancing overall system stability.

Classical PI, PD, and PID controllers are widely used in AVR systems because they are simple, practical, and easy to implement. However, their performance depends strongly on proper tuning, and conventional trial-and-error methods often lead to a compromise among rise time, overshoot, settling time, and steady-state accuracy. Fractional-order controllers provide additional flexibility and can improve dynamic performance, but they also introduce more tuning parameters, making the design problem more complex.

To address this challenge, many studies have adopted optimization-based tuning methods, including genetic algorithms, teaching-learning-based optimization, sine–cosine-based methods, artificial bee colony variants, Archimedes optimization, and hybrid simulated annealing approaches [1]–[12]. These approaches have demonstrated improved transient response and convergence characteristics compared to conventional tuning techniques.

Despite these advances, a practical limitation remains: most existing studies focus on optimizing a single controller structure or a specific problem formulation. A unified framework capable of tuning multiple controller types within the same AVR environment under a common performance criterion is less frequently addressed.

This work addresses that gap by proposing SGHOO as a unified tuning framework for PI, PD, PID, and FOPID controllers on the same AVR plant, with the goal of improving both transient and steady-state performance through a single optimization approach.

II. AVR SYSTEM MODELLING

The AVR benchmark model contains four first-order subsystems: amplifier, exciter, generator, and sensor. Each block is represented as:

$$G(s) = \frac{K}{\tau s + 1}$$

, where K is the gain and τ is the time constant.

Specifically, the amplifier is modeled as

$$G_A(s) = \frac{K_A}{\tau_A s + 1}$$

, the exciter as

$$G_E(s) = \frac{K_E}{\tau_E s + 1}$$

, the generator as

$$G_G(s) = \frac{K_G}{\tau_G s + 1}$$

, and the sensor as

$$G_R(s) = \frac{K_R}{\tau_R s + 1}$$

The overall open-loop transfer function of the AVR system is obtained by cascading these individual transfer functions:

$$G_{\text{plant}}(s) = G_A(s) \cdot G_E(s) \cdot G_G(s) \cdot G_R(s)$$

$$G_{\text{plant}}(s) = \frac{K_A}{\tau_A s + 1} \cdot \frac{K_E}{\tau_E s + 1} \cdot \frac{K_G}{\tau_G s + 1} \cdot \frac{K_R}{\tau_R s + 1}$$

The closed-loop transfer function is given by:

$$G_{\text{CL}}(s) = \frac{C(s) G_{\text{plant}}(s)}{1 + C(s) G_{\text{plant}}(s)}$$

$$G_{\text{plant}}(s) = \frac{10.0 \times 1.0 \times 0.8 \times 1.0}{(0.1s + 1)(0.4s + 1)(1.4s + 1)(0.05s + 1)}$$

$$e(t) = r(t) - y(t)$$

For the simulation used in the report, $K_A = 10.0$, $\tau_A = 0.1$, $K_E = 1.0$, $\tau_E = 0.4$, $K_G = 0.8$, $\tau_G = 1.4$, $K_R = 1.0$, and $\tau_R = 0.05$.

The control error is $e(t) = r(t) - y(t)$, where $r(t)$ is the reference signal and $y(t)$ is the output.

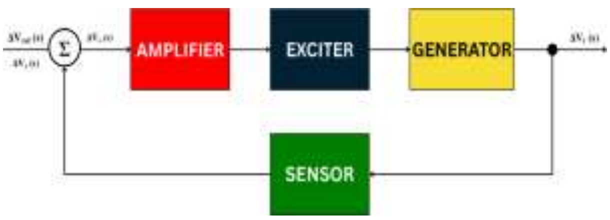


Fig. 1 – *AVR benchmark model used in the study.*

III. CONTROLLER DESIGN

The controller structures considered are PI, PD, PID, and FOPID. Their transfer functions are defined as follows:

For the PI controller:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau$$

$$U(s) = K_p E(s) + \frac{K_i}{s} \cdot E(s) = \left(K_p + \frac{K_i}{s}\right) E(s)$$

$$C_{PI(s)} = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s}$$

For the PD controller:

$$u(t) = K_p e(t) + K_d \frac{de(t)}{dt}$$

$$U(s) = K_p E(s) + K_d s\, E(s) = (K_p + K_d s) E(s)$$

$$C_{PD}(s) = K_p + K_d s$$

For the PID controller:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau + K_d \frac{de(t)}{dt}$$

$$U(s) = K_p E(s) + \frac{K_i}{s} \cdot E(s) + K_d s\, E(s) = \left(K_p + \frac{K_i}{s} + K_d s\right) E(s)$$

$$C_{PID}(s) = K_p + \frac{K_i}{s} + K_d s$$

For the FOPID controller:

$$u(t) = K_p e(t) + K_i D^{(-\lambda)} e(t) + K_d D^{\mu} e(t)$$

$$\mathcal{L}\{D^{(-\lambda)} e(t)\} = \frac{E(s)}{s^{\lambda}}$$

$$\mathcal{L}\{D^{\mu} e(t)\} = s^{\mu} E(s)$$

$$U(s) = \left(K_p + \frac{K_i}{s^{\lambda}} + K_d s^{\mu}\right) E(s)$$

$$C_{FOPID}(s) = K_p + \frac{K_i}{s^{\lambda}} + K_d s^{\mu}$$

The FOPID controller introduces two additional orders, λ and μ. Since standard control libraries do not directly support fractional-order operators, the Oustaloup approximation [13] is employed to approximate s^{λ} as a rational transfer function, given by:

$$s^{\alpha} \approx K \prod_{k=-N}^{N} \frac{1 + \frac{s}{\omega_k}}{1 + \frac{s}{\omega'_k}}$$

IV. PROPOSED SGHOO-BASED OPTIMIZATION

Present The proposed SGHOO algorithm integrates the global exploration capability of the Sine Cosine Algorithm (SCA) [11] with the local exploitation mechanism of the Generalized Normal Distribution Optimization (GNDO) [12]. The objective is to achieve a balanced trade-off between exploration and exploitation, enabling efficient convergence toward the optimal controller parameters.

Initially, a population of candidate solutions is randomly generated within predefined bounds. For each candidate, the closed-loop AVR response is evaluated and a fitness value is computed. Based on this evaluation, the best solution is updated iteratively.

At each iteration, the position of each candidate is updated according to the hybrid SGHOO rule. The position update mechanism of SGHOO is mathematically expressed as follows:

$$x^{(t+1)} = \begin{cases} \text{SCA Update:} & x_j + r_{1,t} = \begin{cases} sin\,(r_2)\left|r_3 x_j^* - x_j\right|, & r_4 < 0.5, \\ cos\,(r_2)\left|r_3 x_j^* - x_j\right|, & otherwise, \end{cases} \\ & \text{(applied for each } j = 1, \dots, d); \\ \text{GNDO Update:} & \begin{cases} \text{Local Exploitation:} & \begin{cases} \mu = \dfrac{x + x^* + M}{3}, \delta = \sqrt{\dfrac{\|x-\mu\|^2 + \|x^*-\mu\|^2 + \|M-\mu\|^2}{3}}, \\ x^{(t+1)} = \mu + \delta\eta,\ \eta \sim \mathcal{N}(0,1) \end{cases} \\ \text{Global Exploration:} & \text{Select } p_1, p_2, p_3, \\ & v_1 = \begin{cases} x - p_1, & f(x) < f(p_1) \\ p_1 - x, & otherwise \end{cases}, v_2 = \begin{cases} p_2 - p_3, & f(p_2) < f(p_3) \\ p_3 - p_2, & otherwise \end{cases}, \\ & \text{with } \beta \sim \text{ Uniform } (0,1), \lambda_3 = |\mathcal{N}(0,1)|, \lambda_4 = |\mathcal{N}(0,1)|, \\ & x^{(t+1)} = x + \beta\lambda_3 v_1 + (1-\beta)\lambda_4 v_2. \end{cases} \end{cases}$$

With a predefined probability P_{SCA}, the SCA-based update is employed, where the position of each dimension (x_j) is modified using sinusoidal functions controlled by random parameters r_1, r_2, r_3, and r_4. These parameters regulate the step size, direction, and stochastic behavior of the search, allowing candidate solutions to move toward or away from the current best solution (x_j^*), thereby enhancing global exploration.

Otherwise, the GNDO-based update is applied, which consists of both local and global search mechanisms. In the local exploitation phase, a mean position (μ) is computed using the current solution, the best solution, and the population mean M, while the spread (δ) defines the search radius. A normally distributed random variable is then used to refine the solution around this region. In the global exploration phase, the current solution is perturbed using randomly selected individuals from the population to maintain diversity and avoid premature convergence.

This probabilistic switching between SCA and GNDO updates enables the algorithm to maintain an effective balance between exploration and exploitation throughout the optimization process, thereby improving convergence speed and solution accuracy.

The overall SGHOO-based optimization framework applied to the AVR system is illustrated in Fig. 2.

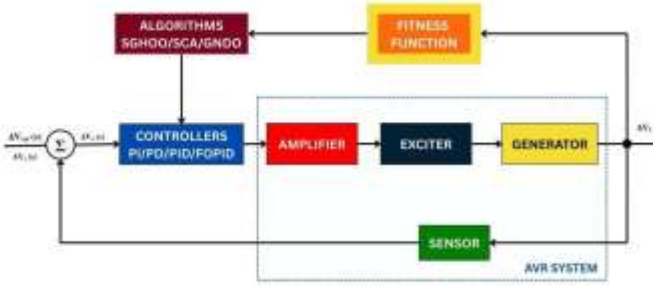


Fig. 2 – *SGHOO optimization loop and fitness-evaluation structure.*

Fig. 2 illustrates the overall SGHOO-based optimization framework for the AVR system. The error signal, obtained from the difference between reference and actual output voltage, is fed to the controller block (PI/PD/PID/FOPID), whose parameters are optimized using the SGHOO algorithm.

The optimized controller drives the AVR system, consisting of the amplifier, exciter, generator, and sensor components. The system output is continuously fed back to form a closed-loop system.

The fitness function evaluates system performance based on transient and steady-state characteristics, and this evaluation guides the SGHOO algorithm in updating controller parameters iteratively until optimal performance is achieved.

V. FITNESS FUNCTION

The objective is to minimize a single score that combines transient-response and error measures:

$$J = w_1 T_r + w_2 T_s + w_3 M_p + w_4 U_s + w_5 e_{ss} + w_6 IAE + w_7 ITAE + w_8 ISE + w_9 ISTE$$

where T_r is the rise time, T_s is the settling time, M_p is the maximum overshoot, U_s is the undershoot, e_{ss} is the steady-state error, and IAE, ITAE, ISE, and ISTE are the integral performance indices.

The code uses the weight vector [1, 1, 1, 1, 5, 1, 1, 1, 1], which places higher emphasis on steady-state error while still penalizing slow or oscillatory responses.

VI. OPTIMIZATION PROCEDURE

1. Initialize a population of candidate controller parameters within the prescribed bounds.
2. Evaluate each candidate by simulating the AVR closed-loop step response and computing the fitness value.
3. At each iteration, update candidates using either the SCA exploration rule or the generalized normal local-search rule.

International Journal of Creative and Open Research in Engineering and Management
ISSN: **3108-1754 (Online)**
Volume 02 Issue 05 May-2026 | Impact Factor: 3.5

4. Clip updated parameters to their bounds, re-evaluate fitness, and retain the better solution.
5. Update the global best solution and repeat until the maximum number of iterations is reached.

This procedure is repeated separately for PI, PD, PID, and FOPID controllers so that the best parameter set for each structure can be obtained under the same evaluation rule.

VII. PERFORMANCE METRICS

The performance of the optimized AVR system is evaluated using standard time-domain specifications, including rise time (T_r), settling time (T_s), overshoot (M_p), and steady-state error (e_{ss}). These metrics provide insight into both the transient and steady-state behaviour of the system.

To assess cumulative error performance, integral-based indices are also considered, defined as:

$$IAE = \int_0^\infty |e(t)|dt$$

$$ITAE = \int_0^\infty t|e(t)|dt$$

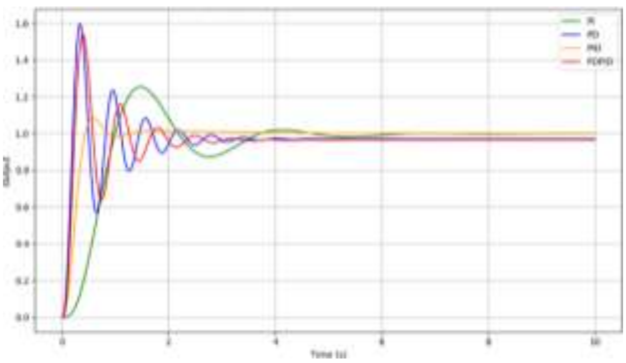
$$ISE = \int_0^\infty e^2(t)dt$$

$$ISTE = \int_0^\infty te^2(t)dt$$

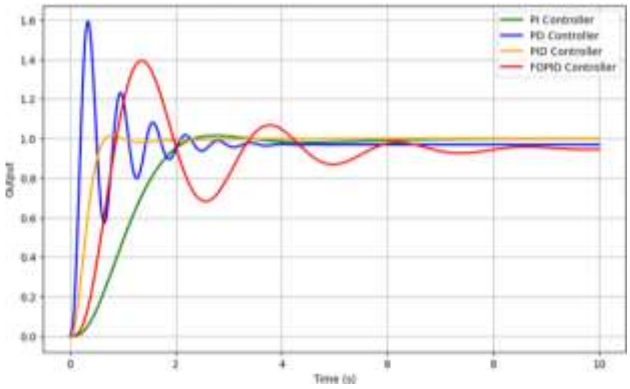
These indices provide a quantitative measure of system accuracy over time and are widely used for controller performance evaluation. Lower values of these indices indicate improved tracking performance and reduced error.

VIII. RESULTS AND DISCUSSION

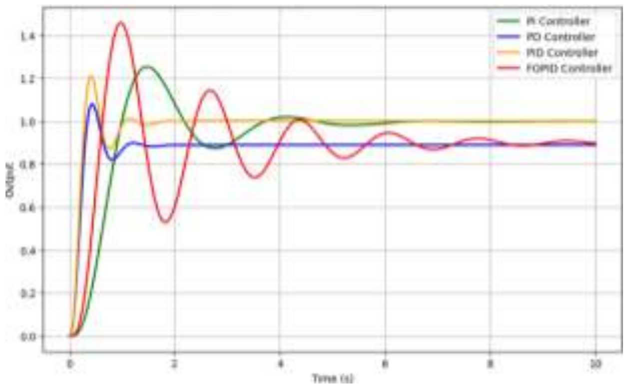
The comparative step responses of the optimized controllers are shown in Figure 3.



(a)



(b)



(c)

Fig. 3 – (a) *Step response of optimized controllers using the proposed SGHOO algorithm;* (b) *Step response of optimized controllers using the SCA algorithm;* (c) *Step response of optimized controllers using the GNDO algorithm.*

The curves indicate that all controller types stabilize the AVR plant, while the proposed hybrid optimizer produces smoother convergence and improved response shaping relative to the baseline algorithm variants.

Across the considered controllers, SGHOO achieves a balanced response. The PID case provides the best overall compromise between speed and accuracy, whereas the FOPID case retains greater flexibility at the cost of a longer settling time, which is expected due to the additional fractional-order parameters. The PI and PD controllers are simpler and easier to tune, but their integral-error performance is generally less favourable than that of the optimized PID case.

The numerical results are summarized in Tables 1 and 2. The reported values show that SGHOO achieves competitive rise times and often lower cumulative error measures than the SCA-and GNDO-based baselines. In particular, the SGHOO-tuned PID controller yields a rise time of 0.16 s, a settling time of 0.962 s, an overshoot of 0.222, and zero steady-state error. The corresponding integral indices are also low, indicating strong tracking performance.

Overall, the results suggest that the hybrid search strategy is effective because SCA contributes broad exploration in the early stages of search, while the generalized-normal update refines the controller parameters near promising regions. This combination is especially suitable for AVR tuning, where fast transient response and small steady-state error must be achieved simultaneously.

Table 1
Time-domain performance metrics of optimized controllers

Controller/ Algorithm	Rise Time [s]	Settling Time [s]	Overshoot	Steady-State Error
SGHOO - PI	0.601	3.547	0.253	0.001
SGHOO-PD	0.120	10	0.596	0.031
SGHOO-PID	0.160	0.962	0.222	0
SGHOO-FOPID	0.261	10	0.314	0.089
SCA-PI	1.343	2.164	0.014	0.001
SCA-PD	0.120	10	0.593	0.031
SCA-PID	0.381	0.601	0.016	0
SCA-FOPID	0.521	10	0.393	0.055
GNDO-PI	0.601	3.547	25.28	0.001
GNDO-PD	0.200	10	7.940	0.111
GNDO-PID	0.160	0.982	21.10	0
GNDO-FOPID	0.361	10	45.83	0.103

Table 2
Integral error indices of optimized controllers

Controller/ Algorithm	IAE	ITAE	ISE	ISTE
SGHOO - PI	0.975	1.068	0.552	0.231
SGHOO-PD	0.715	1.758	0.227	0.113
SGHOO-PID	0.276	0.109	0.155	0.017
SGHOO-FOPID	1.264	4.662	0.358	0.482
SCA-PI	1.150	1.018	0.801	0.386
SCA-PD	0.714	1.763	0.226	0.113
SCA-PID	0.347	0.121	0.235	0.032
SCA-FOPID	1.520	3.832	0.652	0.589
GNDO-PI	0.975	1.068	0.552	0.231
GNDO-PD	1.280	5.593	0.282	0.635
GNDO-PID	0.275	0.102	0.157	0.017
GNDO-FOPID	1.756	5.766	0.649	N/A

IX. CONCLUSION

This paper presented a software-based approach for tuning AVR controllers using the Sine–Generalized Hybrid Optimization Operator. The benchmark AVR model was represented using first-order amplifier, exciter, generator, and sensor blocks, and the controller parameters were optimized in Python.

The proposed SGHOO framework combines the exploration strength of SCA with a generalized normal local-search step, enabling a practical balance between diversification and intensification during optimization. The method was applied to PI, PD, PID, and FOPID controllers under a common fitness function based on transient and error-performance measures.

The results indicate that the optimized controllers achieve stable closed-loop behavior, with the PID configuration giving the most favorable compromise between speed and accuracy in the reported simulations. The study confirms that a unified hybrid optimization framework can effectively support AVR controller tuning and can serve as a foundation for future work on real-time and large-scale power-system applications.

REFERENCES

[1] S. Panda and N. P. Padhy, "Design Optimization of PID Controller in Automatic Voltage Regulator System Using Taguchi Combined Genetic Algorithm Method," IEEE Systems Journal, vol. 7, no. 4, pp. 825–831, 2013.

[2] S. A. Nowdeh, M. G. Monfared, M. J. H. Moghaddam, and M. Bigdeli, "Design optimisation of PID controller in automatic voltage regulator system using teaching-

learning based optimisation algorithm," International Journal of Power and Energy Conversion, vol. 7, no. 4, pp. 377–393, 2016.

[3] M. Soliman, M. Abdel-Rahman, and M. Badr, "Optimal FOPID/PID controller parameters tuning for the AVR system based on sine–cosine algorithm," Evolutionary Intelligence, vol. 12, no. 2, pp. 145–160, 2019.

[4] J. K. Bokam, N. Patnana, T. Varshney, and V. P. Singh, "Sine Cosine Algorithm Assisted FOPID Controller Design for Interval Systems Using Reduced-Order Modeling Ensuring Stability," Algorithms, vol. 13, no. 12, p. 317, 2020.

[5] M. Micev, M. Ćalasan, and D. Oliva, "Fractional Order PID Controller Design for an AVR System Using Chaotic Yellow Saddle Goatfish Algorithm," Mathematics, vol. 8, no. 7, p. 1182, 2020.

[6] A. Agwa, S. Elsayed, and M. Ahmed, "Design of Optimal Controllers for Automatic Voltage Regulation Using Archimedes Optimizer," Intelligent Automation & Soft Computing, vol. 33, no. 2, pp. 1–12, 2022.

[7] H. Altınkaya and D. Ekmekci, "Tuning of PID Controller in PLC-Based Automatic Voltage Regulator System Using Adaptive Artificial Bee Colony–Fuzzy Logic Algorithm," Electronics, vol. 13, no. 24, p. 5039, 2024.

[8] S. Alghamdi et al., "Optimal PID Controllers for AVR Systems Using Hybrid Simulated Annealing and Gorilla Troops Optimization," Fractal and Fractional, vol. 6, no. 11, p. 682, 2022.

[9] A. K. Ali, "An optimal design for an automatic voltage regulation system using a multivariable PID controller based on hybrid simulated annealing–white shark optimization," Scientific Reports, vol. 14, p. 30218, 2024.

[10] Y. Zhou, J. Zhang, X. Yang, and Y. Ling, "Optimization of PID Controller Based on Water Wave Optimization for an Automatic Voltage Regulator System," Journal of Information Technology and Control, vol. 48, no. 1, pp. 160–171, 2019.

[11] S. Mirjalili, "SCA: A Sine Cosine Algorithm for solving optimization problems," Knowledge-Based Systems, vol. 96, pp. 120–133, 2016.

[12] A GNDO paper/source such as "Generalized Normal Distribution Optimization Algorithm for Economic Dispatch with Renewable Resources Integration," Journal of Energy and Power Technology, 2023, DOI: 10.21926/jept.2303030, or another direct GNDO paper if you prefer the exact original algorithm source.

[13] A. Oustaloup, "La Commande CRONE: Commande Robuste d'ordre non entier," Hermes, Paris, 1991.