2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

Designing Heuristic-Based Tuners for PID Controllers in Automatic Voltage Regulator Systems Using an Automated Hyper-Heuristic Approach*

Daniel F. Zambrano-Gutierrez*, Alberto C. Molina-Porras†, Juan G. Avina-Cervantes†,
Rodrigo Correa‡, and Jorge M. Cruz-Duarte*,
*Tecnologico de Monterrey, Monterrey 64849, Mexico, E-mails: {A00836756, jorge.cruz}@tec.mx
†University of Guanajuato, Salamanca 36885, Mexico, E-mails: {a.molinaporras, avina}@ugto.mx
‡Universidad Industrial de Santander, Bucaramanga 680002, Colombia, E-mail: crcorrea@saber.uis.edu.co

***Abstract*—Engineering processes often require optimizing model variables for satisfactory solutions. Reliable approaches exist in literature but are application-dependent. In that sense, metaheuristics have been proven to deliver outstanding results while imposing a low computing burden. However, choosing the most suitable one from the many available can overwhelm even experts. This study implements a methodology that automatically tailors a problem-based metaheuristic through a hyper-heuristic approach. We select the tuning problem of a Proportional Integral Derivative controller as a case study for achieving the best stable features in an Automatic Voltage Regulator system. The numerical results demonstrate the reliability and potential of the implemented methodology in solving control system tuning. Plus, we conduct an in-depth quantitative comparison with recent works in the literature that support those conclusions.**

***Index Terms*—Automatic Algorithm design, Metaheuristics, Control Engineering, Parameter Tuning, Complex Systems.**

I. Introduction

OPTIMIZATION methods are crucial in solving practical design engineering problems [1, 2]. These algorithms seek, at least for approximate, optimal solutions to complex problems, searching in the space of potential solutions. Deterministic approaches and Metaheuristics (MHs) have been developed in the literature to tackle real problems [3].

However, gradient-based deterministic methods may lose effectiveness in highly nonlinear scenarios due to continuity issues [4]. Conversely, MHs constitute a valuable alternative due to their flexibility and low computing burden. They use searching techniques inspired by natural principles that mimic the behavior of biological, social, or physical processes to find efficient solutions [5, 6]. Some examples of classical MHs are Genetic Algorithms [7], Simulated Annealing [8], and Particle Swarm Optimization [9]. Recall that an MH may not always offer the best solutions for all optimization problems, as indicated by the No-Free-Lunch theorem [10]. Therefore, selecting the appropriate MH for a specific case can take significant time due to the colorful palette of approaches available in the literature [11, 12].

An efficient way to get the suitable MH for the problem is through the automatic and intelligent generation of MHs via a Hyper-Heuristic (HH) model [13]. Recently, several proposals have been reported in the Automatic Algorithm Design (AAD), primarily for dealing with combinatorial optimization problems [14, 15]. In the case of continuous problems, it was reported in [16] a framework that focuses on customizing MHs by searching on the heuristic space. According to the performance tests, the presented numerical results demonstrate that the automatically generated MHs efficiently solve classical benchmark functions from the literature. Moreover, such a framework has opened a wide gamut of opportunities for MHs that can solve engineering tuning problems, such as those related to electrical and robotic systems [17, 18]. In this context, a particular case is the Automatic Voltage Regulator (AVR) systems [19]. AVRs are generally designed to maintain a constant output voltage in generators or power generation systems independently of load fluctuations or energy demand [20]. Still, AVRs require a well-designed and tuned controller to ensure the stability and performance of the power grid face disturbances [21, 22]. Some studies have employed Proportional, Integral, and Derivative (PID) controllers due to their versatility and easy implementation [23, 24]. However, proper PID controller tuning is crucial in the precise operation of the AVR system. In this study, we implement the HH framework reported in [16] and evaluate it in the tuning problem of PID controllers for AVR systems. So, we automatically generate a customized population-based MH for tuning a PID controller, achieving the AVR system with the desired output voltage stability features. In addition, the main contributions of this work are summarized as follows:

i) We demonstrate the potential of an automated methodology to generate an MH for tuning a PID controller and improving the robustness of an AVR system against disturbances.

*Daniel F. Zambrano-Gutierrez and Alberto Carlos Molina-Porras thank the Consejo Nacional de Humanidades, Ciencia y Tecnología (CONAHCyT) for the financial support provided through their fellowships 1046000 and 1156648. Daniel F. Zambrano-Gutierrez and Jorge M. Cruz-Duarte also thank the Tecnológico de Monterrey for their support under grant FAP 2275.

ii) We confirm the exceptional performance of the custom MH compared with two approaches recently proposed in the literature.
iii) We corroborate that the PID control tuned by the tailored MH can compensate for induced disturbances, keeping the output voltage stable.

II. Foundations

This section describes the most relevant concepts we employed in this work, such as AVR, PID, and AAD.

A. Automatic Voltage Regulator System (AVR)

An AVR system is an energy technological solution of vital importance in electricity distribution and generation. It adjusts the voltage in one or more phases of an electrical system in response to changes in input voltage, load fluctuations, and other external factors [19]. Its primary purpose is maintaining the voltage at stable and constant levels, avoiding fluctuations detrimental to the connected devices, which can have adverse consequences. Hence, an AVR must ensure reliable and optimal power quality in electrical supply systems.

A basic AVR system can be modeled as four interconnected subsystems: an amplifier, exciter, generator, and sensor. Each subsystem performs specific functions in accurately regulating the output voltage [25]. Fig. 1 depicts these subsystems represented by their transfer function,

$$G_k(s) = \frac{K_k}{1 + s\tau_k}, \tag{1}$$

since the subscript $k = \{a, e, g, s\}$ corresponds to the initial letter of each subsystem's name. These first-order systems are represented by a gain K_k and a time constant τ_k. The values presented in TABLE I were selected based on [23, 24] to ensure a fair comparison using the same plant setup.

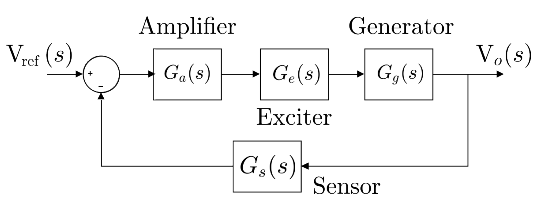

Fig. 1. AVR system model without a controller.

TABLE I
Gain and time constant values used for the AVR subsystems' transfer functions.

Subsystem:	Amplifier	Exciter	Generator	Sensor
Gain:	$K_a = 10.0$	$K_e = 1.0$	$K_g = 1.0$	$K_s = 1.0$
Time Constant:	$\tau_a = 0.1$ s	$\tau_e = 0.4$ s	$\tau_g = 1.0$ s	$\tau_s = 0.01$ s

The AVR system's behavior can be evaluated through its step response, as Fig. 2 depicts. However, it suffers from a non-zero steady-state error, high overshoot, and long settling times. For future comparisons, the AVR system without a controller exhibits an Overshoot (M_p) of 67.42%, a Settling Time (T_s) of 6.971 s, a Rise Time (T_r) of 0.754 s, and a Steady-State Error (E_{ss}) of 0.090 p.u. With these poor features, the need for implementing a controller that guides the system dynamics to a desired behavior is evident.

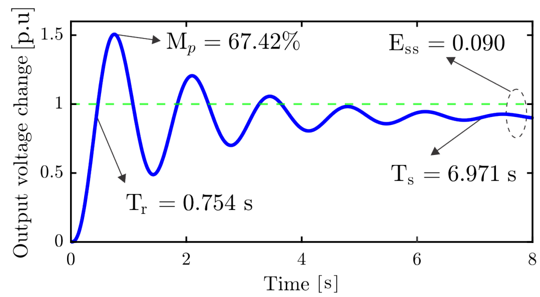

Fig. 2. Output voltage of the AVR system without a controller.

B. AVR System with a PID Controller

This study uses a Proportional, Integral, and Derivative (PID) controller due to its simple structure and easy implementation. As its name indicates, this controller combines three fundamental control actions in response to the error signal, which compares the reference and the sensor's output [26]. The controller's transfer function is given by

$$G_{\text{PID}}(s) = K_p + K_i \frac{1}{s} + K_d\, s, \tag{2}$$

where K_p, K_i, and K_d are the proportional, integral, and derivative gains, respectively.

These parameters must be tuned appropriately to guarantee that the PID controller regulates and stabilizes the AVR output voltage efficiently. This task may seem simple, but it is not trivial. It could require expertise and knowledge about the process, an extensive analytical procedure, or sophisticated algorithms. Hence, the tuning process involves balancing system dynamic response and stability, considering the specific plant features and operating conditions. In this work, we implement a heuristic-based algorithm tailored for tuning this controller for the AVR system. The tailoring procedure is described in the following sections.

C. Automatic Algorithm Design (AAD)

AAD is a well-known field focused on high-level techniques that select or generate algorithms for dealing with a problem family [27]. This designing task can be defined using the General Combinatorial Optimization Problem (GCOP) [28] for heuristic-based algorithms such as metaheuristics. For the particular case of continuous optimization problems, a specific variant is referred to as the Metaheuristic Composition Optimization Problem (MCOP) [29], given by

$$(\mathrm{MH}_*; \vec{x}_*) = \underset{\mathrm{MH}\in\mathfrak{H}^{\varpi},\, \vec{x}\in\mathfrak{X}}{\operatorname{argmax}} \{Q(\mathrm{MH}\,|\mathfrak{X})\}, \tag{3}$$

since $Q(\mathrm{MH}\,|\mathfrak{X})$ corresponds to a performance metric value associated to a Metaheuristic (MH) implemented on a particular problem $(\mathfrak{X}, f)$. Consider that $\mathfrak{X}$ corresponds to the feasible problem domain and f to the objective function. Now, a high-level algorithm that deals with the MCOP is commonly called Hyper-Heuristic (HH), no matter its nature.

A HH searches within the heuristic space $\mathfrak{H}^{\varpi}$ to find the best heuristic sequence that composes the optimal MH (MH_*) rendering a maximal performance $Q(\mathrm{MH}_* \,|\mathfrak{X})$ [30]. (ϖ stands for the number of operators in the MH.)

Moreover, to better understand the HH process, it is necessary to review the term '*heuristic*.' In a broad sense, a heuristic is a procedure that generates or modifies one or more candidates for a solution in a given problem [31]. In particular, heuristics can be classified into three categories based on their level of abstraction and interaction with the problem domain [29]. At the lowest level of abstraction, there is the Simple Heuristics (SHs), which interact directly with the problem domain. Then, MHs interact indirectly with the problem by controlling the search procedure done by a fixed sequence of SHs. At the highest level, HHs search in the heuristic space for either a combination of SHs or an MH that tackles the problem.

Considering the heuristic interaction nature with the search domain (either the problem or heuristic one), these can be classified according to [32, 33] as follows: A *Constructive Heuristic* (h_i) creates a candidate solution from scratch; a *Perturbative Heuristic* (h_p) modifies a candidate solution; and a *Selective Heuristic* (h_s) evaluates a candidate solution and decides whether to accept it or look for another one.

Keeping this information in mind, an MH can be described using the three essential components: an initialization heuristic h_i; at least one Search Operator (SO) h_o, which is composed by a perturbative heuristic h_p succeeded by a selective one h_s; and a selective heuristic operating as master strategy or finalizer (h_f). Thus, $\mathrm{MH} = \langle h_i, h_o, h_f \rangle$, where h_o can be a SO composed of many SOs given in a sequence, *i.e.*, $h_o = h_{\varpi} \circ \cdots \circ h_1, \forall\, h_k \in \vec{h} \in \mathfrak{H}^{\varpi}$. Further information can be found in [31, 32].

III. Proposed Approach

This section details the methodology proposed to generate heuristic-based tuners for a PID controller in an AVR system. We organize the following sections w.r.t. the low-level and high-level domains corresponding to the controller tuning problem and the MCOP, respectively. Figure 3 shows an overview of this methodology.

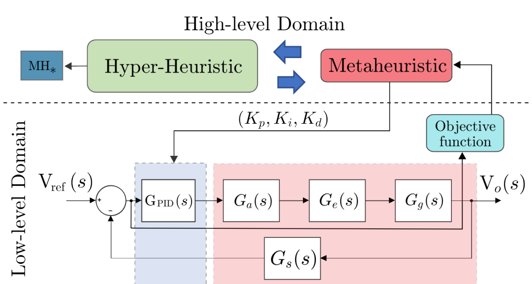

Fig. 3. Overview of the hyper-heuristic procedure to generate a metaheuristic for tuning a PID controller for an AVR system.

A. Low-Level Domain

First of all, the low-level problem domain, corresponding to the PID controller tuning, is a three-dimensional space where is the design vector $(K_p, K_i, K_d)^{\intercal} \in \mathfrak{X} \subseteq \mathbb{R}^3$. In the literature, it is common to find several approaches to evaluate the performance of a controller once coupled to the dynamic system [34, 35]. However, traditional performance metrics, such as Integral Time Absolute Error (ITAE), Integral Square Error (ISE), Integral of Time multiplied Squared Error (ITSE), and Integral Absolute Error (IAE), are of particular importance in controller tuning. These metrics are defined as

$$\begin{aligned} \mathrm{IAE} &= \textstyle\int_0^T |e(\tau)| d\tau, & \mathrm{ISE} &= \textstyle\int_0^T e^2(\tau) d\tau, \\ \mathrm{ITAE} &= \textstyle\int_0^T \tau |e(\tau)| d\tau, & \mathrm{ITSE} &= \textstyle\int_0^T \tau e^2(\tau) d\tau. \end{aligned} \tag{4}$$

We consider them as objective functions in the low-level problem domain and also regard other functions incorporating specific system characteristics [23, 24, 36], given by

$$\begin{aligned} &\mathrm{OF}k = \alpha_1 \times g_k + \alpha_2 \times \mathrm{T_s} + \alpha_3 \times \mathrm{M}_p, \\ &\mathrm{ZLG} = (\mathrm{M}_p + E_{ss}) \times \left(1 - e^{-\beta}\right) + (\mathrm{T_s} - t_r) \times e^{-\beta}, \end{aligned} \tag{5}$$

since $g_k \in \{\mathrm{ITAE}, \mathrm{IAE}, \mathrm{ITSE}, \mathrm{ISE}\}, \forall\, k = 1, \ldots, 4$. Plus, $\alpha_1 = \alpha_2 = \alpha_3 = 0.33$, and $\beta = 0.8$ were determined for this study. Each objective function evaluates the controller's performance from different dynamic perspectives. However, the effect of each of these objective functions on the PID controller tuning is only detailed. Besides, we prioritize implementing the traditional ITSE metric to generate a customized MH through an HH process.

B. High-Level Domain

In this case, the high-level domain corresponds to the heuristic space $\mathfrak{H}^{\varpi}$, where ϖ is the number of SOs used for generating an MH. For this space, we employ a collection composed of 205 SOs, obtained by considering different combinations of SOs extracted from ten well-known MHs [13]. In addition, for evaluating the performance of a candidate sequence of SOs as an MH, we use

$$\mathrm{Q}(\mathrm{MH}\,|\mathfrak{X}) = -(\mathrm{med}(F_h) + \mathrm{iqr}(F_h)), \tag{6}$$

where F_h is given by $F_h = \{f(\vec{x}_{r,*}) \mid \forall\, \vec{x}_{r,*} \in X_*\}$, which stands for a set of fitness values $f(\vec{x}_{r,*})$ achieved after implementing the candidate MH on N_r independent runs; thus, $X_* = \{\vec{x}_{1,*}, \ldots, \vec{x}_{N_r,*}\}$. In this work, we set $N_r = 20$. med and iqr are the median and inter-quartile range operators.

Lastly, to solve the MCOP in (3), we implemented the Simulated Annealing Hyper-Heuristic (SAHH) reported in [16] to facilitate the experimentation. We employed 20 for the population size of SOs, 30 for the MH's maximum iteration number, and 10 for the HH's maximum step number.

IV. Experiments and Results

The proposed strategy to tune a PID controller for an AVR system and achieve a suitable heuristic-based tuner for this problem was tested on a two stages methodology. First, we implemented the HH process to tailor this MH according to Section III. Then, we conducted several experiments to study the influence of different objective functions on the generated MH by examining the features of the AVR system. In the second stage, we analyzed the controller tuned with the achieved metaheuristic regarding stability and performance and its robustness to perturbations. We also compared our resulting controller against others tuned using two state-of-the-art techniques such as the Tree Seed Algorithm (TSA) [24] and Improved Kidney Algorithm (IKA) [23]. Furthermore, we employ a population size of 20 individuals for the SOs, a maximum number of 30 iterations as a finalization criterion of candidate MHs, and a maximum number of 10 steps for the hyper-heuristic search.

All the numerical case studies were conducted in Python v3.9 running on an ASUS TUF Gaming F17 with AMD Ryzen 7 Processor 5700G-8 CPU Cores, 16GB RAM, and using Microsoft Windows 10-64 bit. For implementing the high-level procedures, we utilized the stable CUSTOMHyS v1.1.2 framework, which is freely available at https://pypi.org/project/customhys/ [16]. For numerically simulating the AVR system and each PID controller in the low-level domain, we employed Matlab R2022b. Moreover, all data this work achieved is freely available at https://github.com/Danielfz14/AutoDesign_AVR_PID_Controller.

The first stage focused on tackling the MCOP to find the optimal tuner for a PID controller operating in an AVR system. Fig. 4 displays the HH evolution process, where each box represents the fitness results obtained after repeating 20 times the current candidate MH. At the sixth step of SAHH, we notice that MH_6 is a great algorithm that outperforms the previous candidates. Fig. 4 also shows the performance tendency (orange dashed strokes) characterized by the lowest value of the Q metric from (6).

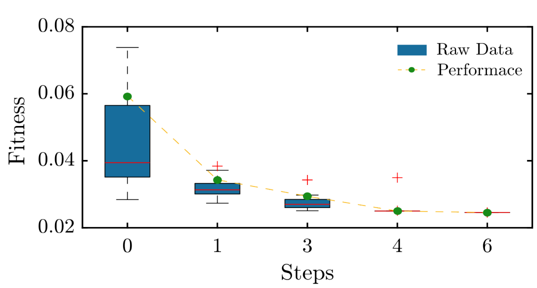

Fig. 4. Evolution tendency of the fitness function achieved by each candidate MH during the HH procedure.

To analyze the heuristic search carried out by SAHH, Fig. 5 depicts the fitness evolution per iteration generated by the candidate MHs at the zeroth (pink strokes), third (green strokes), and sixth (blue strokes) steps. At step zero, the behavior of the initial candidate (MH_0) is expectedly poor, with a considerable dispersion and stagnation far from optimal. In step three, MH_3 presented evident performance improvement with reduced dispersion in all fitness evolution curves. However, some stagnation problems can still be observed. Finally, the sixth and final candidate metaheuristic $\mathrm{MH}_6 \equiv \mathrm{MH}_*$ corresponds to the tailored one (MH_*). It rendered the best performance during the SAHH implementation with a metric value of $2.4{\times}10^{-2}$. However, recall that the HH process is constrained to conduct only ten steps, so considering more HH steps can still be improved. We can now be interested in knowing what is inside this heuristic-based. It is composed of two search operators, *i.e.*, $\mathrm{MH}_* = \langle h_i, h_o, h_f \rangle$, with $\bar{h}_o = h_{18} \circ h_{186}$, following the MH standard model detailed in [32]. The more detailed structure of these operators can be seen as follows:

h_{18}: Differential Mutation with a scale factor of 1.0 as the perturbation heuristic and Greedy as the selection heuristic.

h_{186}: Swarm Dynamic with Gaussian distribution as the perturbation heuristic and Greedy as the selection heuristic.

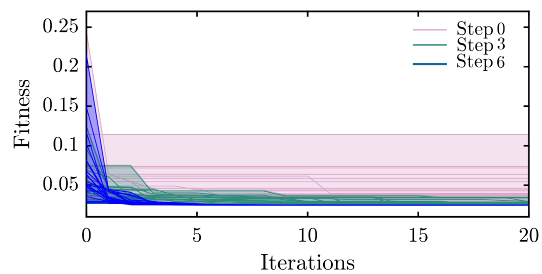

Fig. 5. Fitness evolution generated per the candidate metaheuristics MH_0, MH_3, and MH_6 during the steps 0, 3, and 6 of the hyper-heuristic procedure.

Now, let us examine the effect of the objective function once the customized metaheuristic (MH_*) is generated. To do so, we evaluated its performance on various objective functions using (5) to identify the target function that would provide the best performance for the system. We implemented the MH_* using these functions and employing a maximum of 50 iterations and a population of 30 individuals.

TABLE II
RESULTS OF THE TRANSIENT RESPONSE OF THE AVR SYSTEM TO A STEP INPUT FOR DIFFERENT TARGET FUNCTIONS.

Fobj.	**Fitness**	M_p **[%]**	$\mathbf{T_s}$ **[s]**	$\mathbf{T_r}$ **[s]**	$\mathbf{E_{ss}}$ **[p.u.]**	K_p	K_i	K_d
OF1	0.162	1.83	**0.38**	0.26	$8.62{\times}10^{-5}$	0.680	0.577	0.257
OF2	0.336	27.08	1.99	**0.09**	$5.92{\times}10^{-4}$	1.370	1.814	0.911
OF3	0.152	1.79	0.43	0.28	$2.47{\times}10^{-4}$	0.629	0.560	0.560
OF4	0.303	9.38	0.69	0.21	$6.04{\times}10^{-5}$	0.875	0.784	0.285
ZGL	**0.077**	**0.52**	0.49	0.32	$\mathbf{2.17}{\times}10^{-5}$	0.612	0.410	0.200
ZGL+ITSE	0.081	1.57	0.39	0.26	$8.20{\times}10^{-5}$	0.669	0.571	0.253

TABLE II details the results from this evaluation, where the best-obtained values are highlighted. The ZLG function provides remarkable performance, achieving the lowest overshoot (0.52%) and steady-state error (2.17×10^{-5}). OF1 provided the shortest settling time (0.38 s), and OF2 enabled the fastest rise time (0.09 s). These results exhibit that each function focuses on a particular feature. Therefore, ZLG was selected for the remainder of the analysis as it balances overshoot, settling time, rise time, and steady-state error.

In the second stage, we focus on analyzing the PID controller tuned by the tailored MH and comparing its results with those generated by similar controllers tuned with TSA [24] and IKA [23]. These algorithms were employed to calculate the PID control system parameters based on ITSE objective function criteria, and performances were also explored using other objective functions such as ZLG. TABLE III presents the values for K_p, K_i, and K_d of the PID controllers estimated using the MH$_*$, TSA, and IKA. We can quickly notice that the MH$_*$-tuned controller performs better than the IKA- and TSA-tuned controllers. The overshoot obtained by the MH$_*$-tuned controller was 0.52%, 29.8 and 28.8 times lower than those obtained by the IKA- and TSA-tuned controllers. Plus, the MH$_*$-tuned controller achieved a settling time of 0.49 s, which was 53% and 59% faster than the controllers tuned with IKA and TSA. In the case of the steady-state error, all algorithms improved this metric, but the MH$_*$-tuned controller obtained the lowest value among the others. Fig. 6 summarizes the above analysis, showing the evolution of the output voltage from the AVR system for each controller adjusted via the three different algorithms. The zoom-in square allows for a detailed study of overshoot (M_p) and settling time (T_s) values.

TABLE III
RESULTS OF TRANSIENT RESPONSE ANALYSIS OF THE AVR SYSTEM FOR DIFFERENT CONTROLLERS TUNED BY MH$_*$, IKA, AND TSA.

Alg.	**Fobj.**	M_p **[%]**	**T$_s$ [s]**	**T$_r$ [s]**	**E$_{ss}$ [p.u.]**	K_p	K_i	K_d
MH$_*$	ZGL	**0.520**	**0.490**	0.32	$\mathbf{2.17{\times}10^{-5}}$	0.612	0.410	0.200
IKA [23]	ZGL+ITSE	15.00	0.753	**0.12**	$2.08{\times}10^{-4}$	1.128	0.956	0.567
TSA [24]	ITSE	15.57	0.758	0.13	$8.86{\times}10^{-4}$	1.042	1.009	0.599

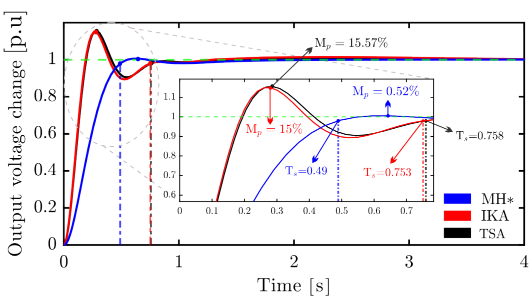

Fig. 6. Step response of the AVR system in a closed-loop and with a PID controller tuned with the tailored metaheuristic MH$_*$, TSA, and IKA.

Then, we perform a root locus and Bode analysis to verify the stability of the AVR system using each controller tuned by MH$_*$, IKA, and TSA. TABLE IV shows the system features for closed-loop poles and damping corresponding to the tuned controllers connected to the AVR system. It is essential to mention that the system is stable for all three methods used since the closed-loop poles are located on the left side of the complex s plane. After comparing the results obtained for each controller, we observed that the conjugate poles of AVR systems are generally more stable. However, we also noticed that the conjugate poles obtained from IKA and TSA are slightly located farther to the left than those from MH$_*$. Despite this, it is paramount to emphasize that the AVR system, tuned using MH$_*$, has a much higher damping ratio of 72% and 68.9% compared to the damping ratios achieved by using IKA and TSA, respectively. This confirms the quick neutralization of the oscillations shown in Fig. 6.

TABLE IV
CLOSED LOOP POLES AND DAMPING RATIOS OF AVR SYSTEM FOR DIFFERENT CONTROLLERS TUNED BY MH$_*$, IKA, AND TSA.

MH$_*$		**IKA**		**TSA**	
Closed Loop Poles	**Damping Ratio**	**Closed Loop Poles**	**Damping Ratio**	**Closed Loop Poles**	**Damping Ratio**
-100.55	1	-101	1	-101	1
-0.9935	1	-0.8-0.9i	0.652	-0.9+0.8i	0.748
-1.8801	1	-0.8+0.9i	0.652	-0.9-0.8i	0.748
-5.04+5.42i	0.681	-5.13+11.9i	0.395	-5.05+11.5i	0.403
-5.04-5.42i	0.681	-5.13-11.9i	0.395	-5.05-11.5i	0.403

Lastly, we conducted a disturbance test to analyze the AVR system controlled by a PID controller. We modified the load of synchronous generators using Heaviside function inputs. The disturbances appeared at 3 and 5 s with 10% and -20% amplitudes, respectively. Fig. 7 exhibits how each controlled system effectively manages these disturbances. However, the MH$_*$-tuned system remarkably performs by quickly stabilizing and smoothing the response.

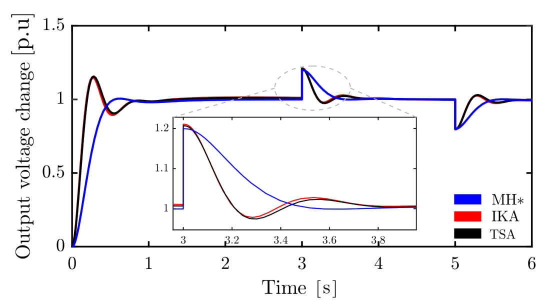

Fig. 7. AVR system responses to disturbances for different controllers tuned by MH$_*$, IKA, and TSA.

V. CONCLUSIONS

This work implemented an Automatic Algorithm Design methodology based on Simulated Annealing to generate a population-based Metaheuristic (MH$_*$) for tuning a PID controller in an Automatic Voltage Regulator System (AVR) system, which is present in various electrical applications.

The MH tailored for this work comprises two SOs, such as a *Differential Mutation* from Differential Evolution followed by a *Greedy selection* and a *Swarm Dynamic with Gaussian distribution* from Particle Swarm Optimization also followed by a *Greedy selection*. With this MH$_*$, we analyzed the effect of the different objective functions from the literature on the system's dynamic responses. We noticed that ZGL is a balanced function in speed response and overshoot of the controlled system. We also compared MH$_*$ against two state-of-the-art algorithms, such as TSA [24] and IKA [23]. From numerical results, we proved that MH$_*$ exhibited superior performance compared to the IKA and TSA algorithms. The controller tuned using MH$_*$ obtained an overshoot of 0.52%, at least 29 times lower than that obtained by the IKA- and TSA-tuned systems. For the settling time, the system controlled by MH$_*$ achieved a value of 0.49 s, at least 53% lower than those obtained by the systems adjusted by IKA and TSA. Plus, we observed that our system achieved a damping ratio at least 69% higher than that obtained by the system using the other algorithms. Finally, we evaluated the robustness of the MH_*-tuned system in a scenario involving changes in the AVR's generator load due to disturbances. We found that the MH_*-tuned system performed superior to those obtained using the IKA and TSA algorithms. Consequently, the response of the MH_*-tuned system evidenced the controller's robustness and efficiency.

We are confident that professionals from any field can implement the proposed methodology for tailoring MHs capable of dealing with complex problems, allowing them to focus on their specific applications.

This work opens room for several future research areas. We shall explore a range of disturbances, evaluate the controller's robustness, and improve its adaptability. We also plan to analyze various electrical system applications to identify patterns for more effective heuristic exploration. Lastly, we shall add more mathematical operators to the HH framework.

REFERENCES

[1] R. T. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering," *Structural and multidisciplinary optimization*, vol. 26, pp. 369–395, 2004.

[2] A. R. Parkinson, R. Balling, and J. D. Hedengren, "Optimization methods for engineering design," *Brigham Young University*, vol. 5, no. 11, 2013.

[3] J. Kim, S. A. Wilkerson, and S. A. Gadsden, "Comparison of gradient methods for gain tuning of a pd controller applied on a quadrotor system," in *Unmanned Systems Technology XVIII*, vol. 9837, pp. 278–287, SPIE, 2016.

[4] S. Shalev-Shwartz, O. Shamir, and S. Shammah, "Failures of gradient-based deep learning," in *International Conference on Machine Learning*, pp. 3067–3075, PMLR, 2017.

[5] N. Razmjooy, M. Ashourian, and Z. Foroozandeh, *Metaheuristics and optimization in computer and electrical engineering*. Springer, 2021.

[6] G. H. Valencia-Rivera, I. Amaya, J. M. Cruz-Duarte, J. C. Ortíz-Bayliss, and J. G. Avina-Cervantes, "Hybrid controller based on lqr applied to interleaved boost converter and microgrids under power quality events," *Energies*, vol. 14, no. 21, p. 6909, 2021.

[7] O. Kramer and O. Kramer, *Genetic algorithms*. Springer, 2017.

[8] D. Bertsimas and J. Tsitsiklis, "Simulated annealing," *Statistical science*, vol. 8, no. 1, pp. 10–15, 1993.

[9] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-international conference on neural networks*, vol. 4, pp. 1942–1948, IEEE, 1995.

[10] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE transactions on evolutionary computation*, vol. 1, no. 1, pp. 67–82, 1997.

[11] M. Abdel-Basset, L. Abdel-Fatah, and A. K. Sangaiah, "Metaheuristic algorithms: A comprehensive review," *Computational intelligence for multimedia big data on the cloud with engineering applications*, pp. 185–231, 2018.

[12] A. Gogna and A. Tayal, "Metaheuristics: review and application," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 25, no. 4, pp. 503–526, 2013.

[13] J. M. Cruz-Duarte, I. Amaya, J. C. Ortiz-Bayliss, S. E. Conant-Pablos, H. Terashima-Marín, and Y. Shi, "Hyper-heuristics to customise metaheuristics for continuous optimisation," *Swarm and Evolutionary Computation*, vol. 66, p. 100935, 2021.

[14] M. A. L. Silva, S. R. de Souza, M. J. F. Souza, and M. F. de Franca Filho, "Hybrid metaheuristics and multi-agent systems for solving optimization problems: A review of frameworks and a comparative analysis," *Applied Soft Computing*, vol. 71, pp. 433–459, 2018.

[15] F. Da Ros and L. Di Gaspero, "Exploring the potential of jules: A white box framework for local search metaheuristics," in *Genetic and Evolutionary Computation Conference Companion (GECCO '23 Companion), July 15–19, 2023, Lisbon, Portugal*, p. 4, ACM, New York, NY, USA, 2023.

[16] J. M. Cruz-Duarte, I. Amaya, J. C. Ortiz-Bayliss, H. Terashima-Marín, and Y. Shi, "CUSTOMHyS: Customising Optimisation Metaheuristics via Hyper-heuristic Search," *SoftwareX*, vol. 12, p. 100628, July 2020.

[17] D. F. Zambrano-Gutierrez, J. M. Cruz-Duarte, J. G. Avina-Cervantes, J. C. Ortiz-Bayliss, J. J. Yanez-Borjas, and I. Amaya, "Automatic design of metaheuristics for practical engineering applications," *IEEE Access*, vol. 11, pp. 7262–7276, 2023.

[18] D. F. Zambrano-Gutierrez, J. Cruz-Duarte, and H. Castañeda, "Automatic hyper-heuristic to generate heuristic-based adaptive sliding mode controller tuners for buck-boost converters," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 1482–1489, 2023.

[19] T. Li, W. Bai, Q. Liu, Y. Long, and C. P. Chen, "Distributed fault-tolerant containment control protocols for the discrete-time multiagent systems via reinforcement learning method," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[20] S. M. Hietpas and M. Naden, "Automatic voltage regulator using an ac voltage-voltage converter," *IEEE Transactions on Industry Applications*, vol. 36, no. 1, pp. 33–38, 2000.

[21] I. Eke, M. Saka, H. Gozde, Y. Arya, and M. C. Taplamacioglu, "Heuristic optimization based dynamic weighted state feedback approach for 2dof pi-controller in automatic voltage regulator," *Engineering Science and Technology, an International Journal*, vol. 24, no. 4, pp. 899–910, 2021.

[22] P. Sirsode, A. Tare, and V. Pande, "Design of robust optimal fractional-order pid controller using salp swarm algorithm for automatic voltage regulator (avr) system," in *2019 Sixth Indian Control Conference (ICC)*, pp. 431–436, 2019.

[23] S. Ekinci and B. Hekimoğlu, "Improved kidney-inspired algorithm approach for tuning of pid controller in avr system," *IEEE Access*, vol. 7, pp. 39935–39947, 2019.

[24] E. Köse, "Optimal control of avr system with tree seed algorithm-based pid controller," *IEEE Access*, vol. 8, pp. 89457–89467, 2020.

[25] G. A. Salman, A. S. Jafar, and A. I. Ismael, "Application of artificial intelligence techniques for lfc and avr systems using pid controller," *International Journal of Power Electronics and Drive Systems*, vol. 10, no. 3, p. 1694, 2019.

[26] M. A. Sahib and B. S. Ahmed, "A new multiobjective performance criterion used in pid tuning optimization algorithms," *Journal of advanced research*, vol. 7, no. 1, pp. 125–134, 2016.

[27] J. Stork, A. E. Eiben, and T. Bartz-Beielstein, "A new taxonomy of global optimization algorithms," *Natural Computing*, pp. 1–24, 2020.

[28] R. Qu, G. Kendall, and N. Pillay, "The general combinatorial optimisation problem: Towards automated algorithm design," *IEEE Comput. Intell. Mag.*, vol. 15, no. 2, pp. 14–23, 2020.

[29] J. M. Cruz-Duarte, J. C. Ortiz-Bayliss, I. Amaya, and N. Pillay, "Global optimisation through hyper-heuristics: Unfolding population-based metaheuristics," *Applied Sciences*, vol. 11, no. 12, p. 5620, 2021.

[30] J. Cruz-Duarte, I. Amaya, J. Ortiz-Bayliss, and N. Pillay, "Automated design of unfolded metaheuristics and the effect of population size," in *IEEE Congress on Evol. Comp. (CEC)*, pp. 1155–1162, 2021.

[31] J. H. Drake, A. Kheiri, E. Özcan, and E. K. Burke, "Recent advances in selection hyper-heuristics," *European Journal of Operational Research*, vol. 285, no. 2, pp. 405–428, 2020.

[32] J. Cruz-Duarte, J. Ortiz-Bayliss, I. Amaya, Y. Shi, H. Terashima-Marín, and N. Pillay, "Towards a Generalised Metaheuristic Model for Continuous Optimisation Problems," *Mathematics*, vol. 8, p. 2046, nov 2020.

[33] N. Pillay and R. Qu, *Hyper-Heuristics: Theory and Applications*. Springer, 2018.

[34] S. Panda, B. K. Sahu, and P. K. Mohanty, "Design and performance analysis of pid controller for an automatic voltage regulator system using simplified particle swarm optimization," *Journal of the Franklin Institute*, vol. 349, no. 8, pp. 2609–2625, 2012.

[35] B. Hekimoğlu and S. Ekinci, "Grasshopper optimization algorithm for automatic voltage regulator system," in *2018 5th international conference on electrical and electronic engineering (ICEEE)*, pp. 152–156, IEEE, 2018.

[36] P. K. Mohanty, B. K. Sahu, and S. Panda, "Tuning and assessment of proportional–integral–derivative controller for an automatic voltage regulator system employing local unimodal sampling algorithm," *Electric Power Components and Systems*, vol. 42, no. 9, pp. 959–969, 2014.