Proceedings of the 21st International Conference on Global Research and Education (Inter-Academia 2024)

Nature-Inspired Metaheuristics for Tuning the PI Controller of a High-Voltage Pulse Generator

Z. Kubrak[a] and M. Bartyś[b,*]

[a] *Faculty of Mechatronics, Warsaw University of Technology, św. A. Boboli 8, 02-525 Warsaw, Poland*

[b] *Institute of Automatic Control and Robotics, Warsaw University of Technology, św. A. Boboli 8, 02-525 Warsaw*

Doi: 10.12693/APhysPolA.146.340 *e-mail: michal.bartys@pw.edu.pl

Metaheuristics are currently playing an increasingly important role in the tuning of industrial controllers. In particular, this paper presents the results of implementing various nature-inspired metaheuristics for the tuning of a proportional–integral controller intended for use in a high-voltage pulse generator. This paper analyses and compares the results of tuning obtained using both classical metaheuristics, such as simulated annealing, genetic algorithms, particle swarm optimisation, and differential evolution, and newer approaches, such as sand cat swarm optimisation and sea lion optimisation. An original, complex multi-criteria cost function is constructed in this paper for optimising and ranking nature-inspired metaheuristics for the tuning of the proportional-integral controller. The results show that sand cat swarm optimisation outperforms other optimisation approaches according to the adopted multi-criteria optimisation criterion.

topics: metaheuristics, controller tuning, high-voltage generator, pulse generator

1. Introduction

Proportional–integral (PI) controllers are fundamental components in industrial control loop systems, renowned for their simple structure and effectiveness in the control of multiple processes.

The tuning of PI controllers plays a pivotal role in achieving expected performance, stability, response to disturbances, and robustness across varying operational conditions. Generally, the proper tuning of the controller is not a trivial task. This is particularly true if the control quality criteria are highly sensitive to changes in controller parameters. Numerous tuning approaches have been developed [1–4]. Among these, heuristics [5] and nature-inspired metaheuristics are now playing an increasingly important role [6–13].

This paper aims to compare and rank a set of nature-inspired metaheuristics in the context of their application to the tuning of a PI controller intended for a high-voltage, high-power pulse generator. Such generators are used, among others, in experiments with plasma [14, 15].

The high-voltage pulse generator delivers a series of switched *direct current* (DC) voltage pulses across its output terminals (Fig. 1). The averaged output voltage of an ideal generator is a rectangular wave with amplitude U_s, frequency f_s, and duty cycle d_s. The binary-valued control signal V_c allows for discrete two-level adjustment of the generator output voltage. This signal has a square-like shape with frequency f_c, as shown in Fig. 1.

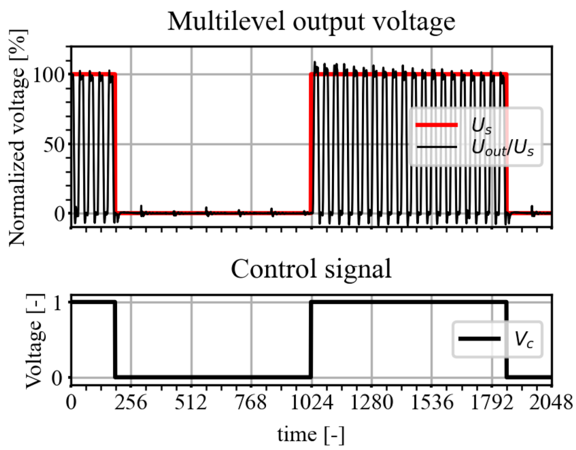

Fig. 1. Output voltage of a pulse generator.

The duty of the signal is determined by the parameter d_c. PI controller indirectly governs the output voltage of high-voltage pulse generator U_{out} by the changes in duty d of the PWM modulator. The block structure of the high-voltage control loop is shown in Fig. 2.

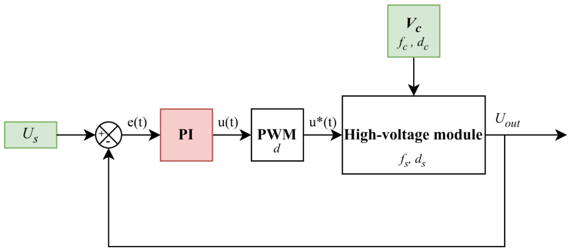

Fig. 2. Block schematics of the high-voltage control system.

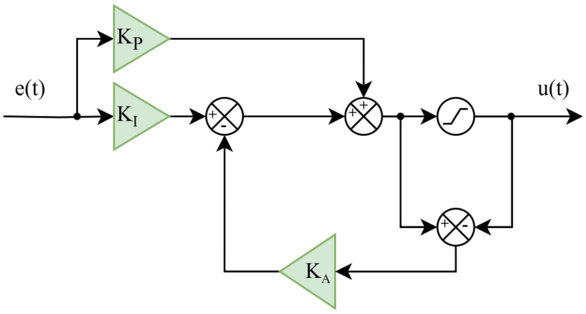

Fig. 3. PI controller block diagram.

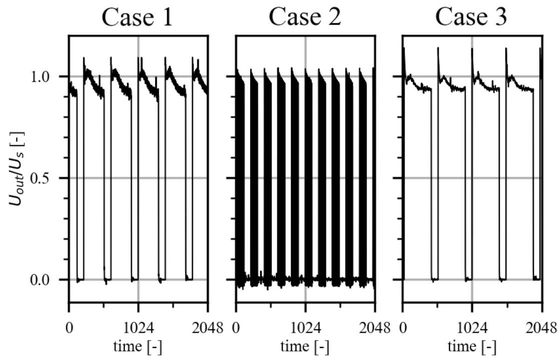

Fig. 4. Illustration of output voltages for reference cases.

TABLE I

Reference control parameters used for benchmark purposes.

Parameter	Unit	P1	P2	P3
f_s	kHz	400	400	400
d_s	%	75	50	50
$U_s/U_{\max}$	–	0.5	0.5	0.75
f_c	kHz	1	10	0.8
d_c	%	75	50	80

The block diagram of the PI controller discussed in this paper is shown in Fig. 3. The structure of this controller is classic. The controller output voltage $u(t)$ is limited due to the need to match the controller output with the range of the input of the pulse-width modulator (PWM). In addition, the controller is equipped with an anti-windup that moderates the controller output $u(t)$ in the event of saturation.

The anti-windup compensates for the output of the integrator K_I on $u(t)$ via negative feedback with integrator K_A. The time constant of this integrator is fixed, and therefore its tuning is not necessary.

Due to the high voltages and high power delivered by the generator, conducting plasma experiments is very risky and requires constant supervision. Therefore, manual tuning of the controller should not be considered the best option.

On the other hand, model-based research requires advanced knowledge of the system under control or at least exhaustive process data. A compromise solution is the idea of using an auxiliary model of the generator that is built using the same electronic components as the real one but with scaled-down output power, currents, and voltages. Such a solution reflects the dynamics of the generator, enabling experimental studies over a wide range of parameters without compromising the safety of the experimenter.

In the case of nonlinear systems being controlled, or those following a setpoint, it is necessary to tune the controller close to the operating point. Therefore, to compare the effectiveness of the selected metaheuristic tuning approaches in seeking optimal or suboptimal PI controller settings, a set of three benchmark cases was arbitrarily defined. The parameters of these cases are presented in Table I, while the waveforms reflecting output voltages of the generator are shown in Fig. 4.

2. Tuning of PI controller

Some simple and industrially proven experimental tuning approaches for PID controllers, such as Ziegler–Nichols or Cohen–Coon, are neither appropriate nor effective in the case of high-voltage generators. The reason is that, as experiments have shown, they do not guarantee the required quality of control.

In recent years, there has been a growing interest in leveraging nature-inspired metaheuristic algorithms to address the challenges of PI controller tuning. Metaheuristic algorithms, drawing inspiration from natural phenomena or biological behaviours, offer a promising alternative by efficiently exploring complex search spaces and identifying near-optimal controller parameters.

The appeal of metaheuristic algorithms lies in their ability to adapt and optimise controller parameters without requiring detailed system models or explicit knowledge of system dynamics. This raises hope for solving the problem of proper tuning for high-voltage generators.

The problem of finding near-optimal PI controller settings can be formulated as seeking the constrained minimum of a two-variable function, where the arguments are the gains of the proportional and integral terms, and the constraints are the permissible ranges of their settings.

Nowadays, different indices are used for the evaluation of the control quality. All of them estimate how well the controller performs. Some of them can be interpreted as integrated cost functions. Commonly used cost functions include the *integral of absolute error* (IAE), *integral of squared error* (ISE), and *integral of time-weighted absolute error* (ITAE). Selecting the proper cost function for the controller settings optimisation is crucial [8].

Obviously, the use of integral performance indices does not directly allow for the control of some desired criteria such as overshoot or settling time [8]. Therefore, to account for multiple criteria, it is necessary to propose a cost function that incorporates these quantities [10, 16–18].

In the case of high-voltage generator, the major problem encountered is the inability to bring the system to a steady state using controller settings resulting from manual tuning. Therefore, in this paper, an optimisation approach was applied to address this problem.

The optimisation aims to improve the following performance indices:

(i) mean of maximum overshoots (e_m),

(ii) mean settling time (t_s) — the time required for the transient's damped oscillations to reach and stay within $\pm 5\%$ of the steady-state value,

(iii) averaged steady-state error (e_s).

The values of these indices are calculated in sliding time windows. It should be emphasised that the performance indices are strongly interdependent. For example, shortening the settling time often comes at the expense of increased overshoots.

Let us define a simple cost function C considering together five performance indices

$$C = M\Phi M^{\mathrm{T}} + O, \tag{1}$$

where M is vector of performance indices, Φ — diagonal matrix of non-negative weights of performance indices, O — offset, and

$$M = \begin{bmatrix} e_m & e_{st} & t_s & MSE & IAE \end{bmatrix}^{\mathrm{T}}. \tag{2}$$

Originally, the matrix Φ was chosen arbitrarily in the form

$$\Phi = \begin{bmatrix} 1.5 & 0 & 0 & 0 & 0 \\ 0 & 1.5 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0 & 0.001 \end{bmatrix}. \tag{3}$$

Additionally, to discourage the optimisation algorithm from selecting controller gains that lead to system instability, an arbitrary chosen value of an offset value O was added to the cost function.

By all experiments, the default settings for PI controller were used as elements of the initial population. In particular, the values $K_P = 0.5$, $K_I = 0.003$ were chosen as the starting points. Together six optimisation algorithms were examined, i.e., *simulated annealing* (SA), *genetic algorithms* (GA), *particle swarm optimisation* (PSO), *differential evolution* (DE), *sand cat swarm optimisation* (SCSO), and *sea lion optimisation* (SLnO).

SA mimics the annealing process used in metallurgy, exploring the solution space by probabilistically accepting even inferior solutions to eliminate the effect of the algorithm getting stuck around a local optimum.

GA is inspired by natural selection, evolving a population of candidate solutions through crossover and mutation.

DE is a population-based algorithm that achieves better solutions by using mutation, crossover, and selection. The optimisation strategy is to select the best individuals from the current population and use them in the mutation process.

PSO simulates the social behaviour of swarms. The AIW-PSO (AIW — *adaptive inertia weight*) variant adjusts the inertia weight dynamically during the optimisation process to balance exploration and exploitation effectively.

SCSO is inspired by the hunting behaviour of sand cats, optimizing solutions by exploring and exploiting the search space through adaptive strategies based on the cats' social and hunting dynamics.

SLnO simulates the social and predatory behaviour of sea lions. It utilizes mechanisms such as spiral updating positions and encircling prey to balance exploration and exploitation in the search process.

It should be added that for all optimisation algorithms, the original implementation of the algorithm was used, except for the PSO algorithm, where the AIW-PSO variant was used.

Table II presents the set of initial parameters chosen for the optimisation.

The ranking of the algorithms was carried out based on the final value of the objective function achieved.

Surprisingly, even a small change in the gains of the PI controller significantly affects control quality. This confirms the observation that the search for optimal PI controller settings is a difficult task. Moreover, the search space for the gains of the PI controller is quite large. Therefore, searching for near-optimal settings can be time-consuming. To reduce this time, a preliminary analysis of the data was conducted to narrow down the search space. For this purpose, a data clustering method was used. Figure 5 illustrates the collected data for all the studied reference cases. For each case, exactly

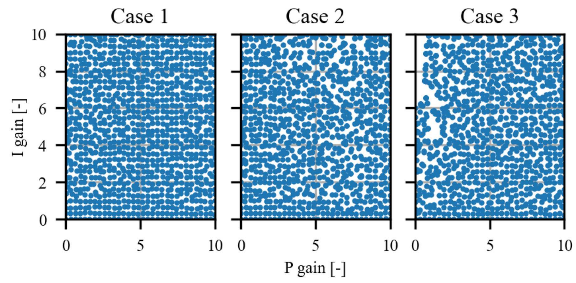

Fig. 5. Representation of acquired data.

TABLE II

Parameters of used algorithms.

Algorithm	Parameter	Symbol	Value
SA	start temperature	$T_{\max}$	25000
	end temperature	$T_{\min}$	2.5
GA	crossover probability	p_c	0.95
	mutation probability	p_m	0.0025
	population size	p_s	30
AIW-PSO	acceleration constant	c_1	2.05
	acceleration constant	c_2	2.05
	inertia factor	α	0.4
	population size	p_s	30
DE	mutation constant	F	0.5
	population size	p_s	30
SCSO	population size	p_s	30
SLnO	population size	p_s	30

TABLE III

Results of GMM clustering in Case 1.

Index	K_P range	K_I range
C	$[0, 10]$	$[0, 0.33507]$
		$[4.80, 9.99]$
IAE	$[0, 10]$	$[0, 0.34632]$
MSE	$[0, 10]$	$[0, 0.34632]$
e_m	$[0, 10]$	$[0, 5.62669]$
e_{st}	$[0, 10]$	$[0, 0.52697]$
t_s	$[0, 10]$	$[0, 0.52697]$

900 measurement data points were collected, evenly covering the plane constrained by the permissible ranges of the PI controller gain settings.

For each case, clustering was carried out using the *Gaussian mixture model* (GMM) algorithm, allowing commonalities to be identified. GMM was useful for recognising the relationships between the performance indices and individual controller settings. Surprisingly, no relationship was found for the proportional gain. The results are summarised in Figs. 6–8. In these figures, the cluster with the lowest average performance index value is marked in green.

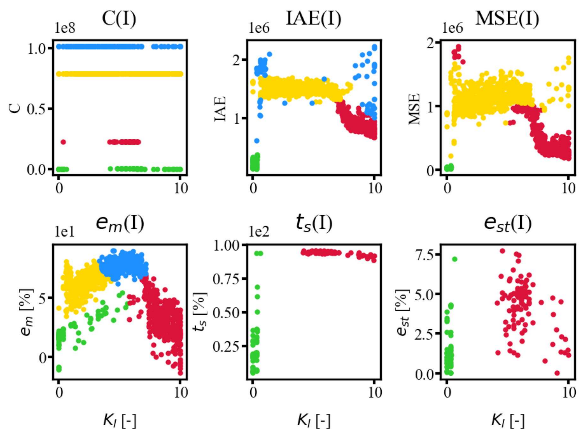

Fig. 6. Results of clustering experimental data in Case 1.

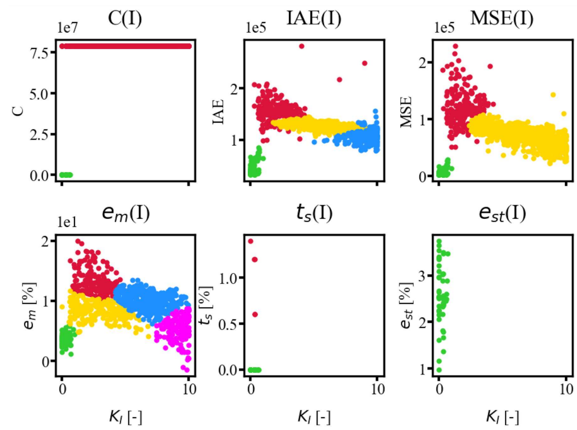

Fig. 7. Results of clustering experimental data in Case 2.

TABLE IV

Results of GMM clustering in Case 2.

Index	K_P range	K_I range
C	$[0, 10]$	$[0, 0.66729]$
IAE	$[0, 10]$	$[0, 1]$
MSE	$[0, 10]$	$[0, 1]$
e_m	$[0, 10]$	$[0, 1]$
e_{st}	$[0, 10]$	$[0, 0.63525]$
t_s	$[0, 10]$	$[0, 0.63525]$

Table III presents the results of clustering achieved for K_I gain.

Based on the data presented in Table III, the search space was narrowed to the range of $[0, 0.52697]$ for the K_I gain.

Figure 7 presents the clusters of experimental data obtained in Case 2.

Table IV presents the results of clustering achieved for K_I gain values indicated by the green-marked clusters.

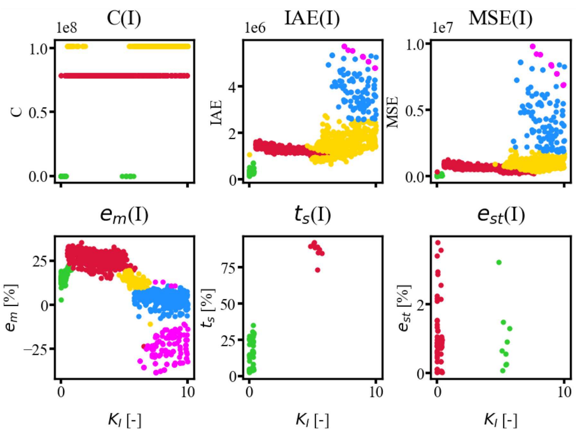

Fig. 8. Results of clustering experimental data in Case 3.

TABLE V

Results of GMM clustering in Case 3.

Index	K_P range	K_I range
C	$[0, 10]$	$[0, 0.3218]$ $[4.848, 5.727]$
IAE	$[0, 10]$	$[0, 0.39845]$
MSE	$[0, 10]$	$[0, 0.52697]$
e_m	$[0, 10]$	$[0, 0.72994]$
e_{st}	$[0, 10]$	$[0, 0.39845]$
t_s	$[0, 10]$	$[4.848, 5.727]$

Based on the data presented in Table IV, the search space was narrowed to the range of $[0, 0.63525]$ for the K_I gain.

Figure 8 shows the clusters of experimental data obtained in Case 3.

Table V presents the results of clustering achieved for the K_I gain.

In this case, the search space for the gain K_I was narrowed to $[0, 0.52697]$.

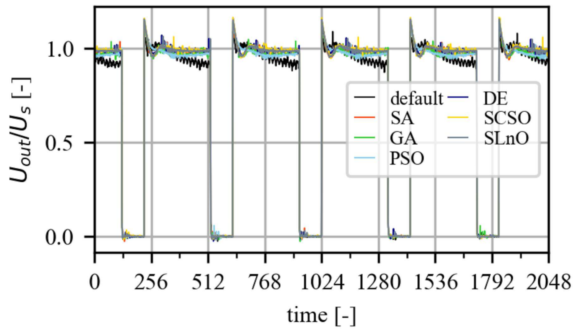

Fig. 9. Generator output voltage waveform in Case 1.

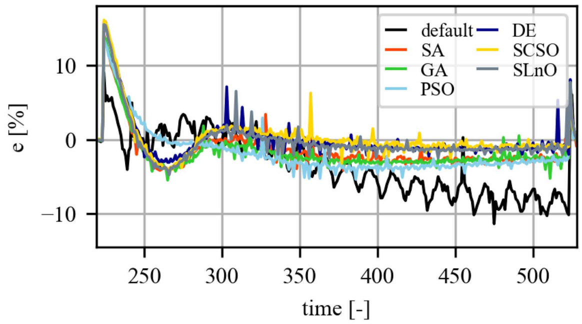

Fig. 10. Relative control error obtained in Case 1.

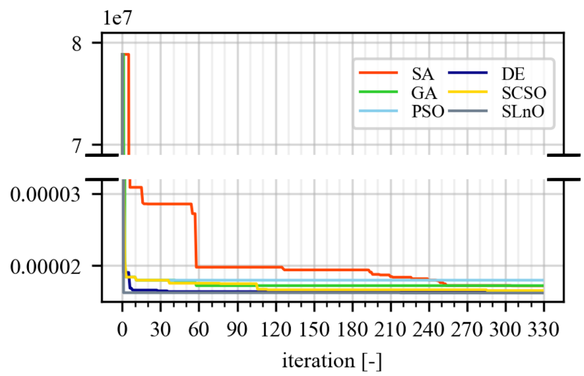

Fig. 11. Convergence of cost functions in Case 1.

3. Experimental findings

The output voltage of the generator was measured in each experiment. The experiments were performed for each set of PI controller settings generated by all the optimisation algorithms tested.

The output voltage and control error for the studied reference Case 1 are shown in Figs. 9 and 10, respectively.

The collected values of performance indices obtained experimentally in Case 1 are depicted in Table VI.

As can be seen in Figs. 9–10 and Table VI, the SLnO algorithm achieves the best performance in terms of minimizing the cost function. It is worth mentioning that the SCSO and DE algorithms also provide a significant improvement over manual tuning.

Moreover, as can be seen in Fig. 11, the SLnO convergence curve of the cost function shows the greatest rate of decrease among all others and falls to the lowest value, thus exhibiting its efficiency in the studied case.

The output voltage and control error for the examined reference Case 2 are shown in Figs. 12 and 13, respectively.

The collected values of performance indices obtained experimentally in Case 2 are depicted in Table VII. Notably, the obtained values differ significantly from those obtained in Case 1.

TABLE VI

Experimental results of the reference Case 1.

Parameter	Unit	Manual	SA	GA	PSO	DE	SCSO	SLNO
K_P	–	0.5	6.6941	6.33648	5.42544	6.86145	7.72135	7.48437
K_I	–	0.003	0.01362	0.07965	0.07492	0.00915	0	0.00125
C	–	78849583	1720	1723	1799	1631	1654	1620
e_m	%	9.1	14.19	13.83	13.44	15.87	16.38	15.45
e_{st}	%	–	−2.499	−2.373	−2.81	−0.94	−0.781	−1.156
t_s	%	–	4.416	4.416	5.584	5.167	5.0	4.584

TABLE VII

Experimental results of the reference Case 2.

Parameter	Unit	Manual	SA	GA	PSO	DE	SCSO	SLNO
K_P	–	0.5	0.69505	0.5	0.57478	0.2926	0.38109	0.05947
K_I	–	0.003	0.34373	0.23699	0.02094	0.01065	0.00562	0.20234
C	–	949	477	479	474	484	466	480
e_m	%	3.62	4.88	2.71	3.17	2.97	4.85	2.75
e_{st}	%	−2.195	−0.716	−2.74	−2.328	−2.648	−0.668	−2.743
t_s	%	0	0	0	0	0	0	0

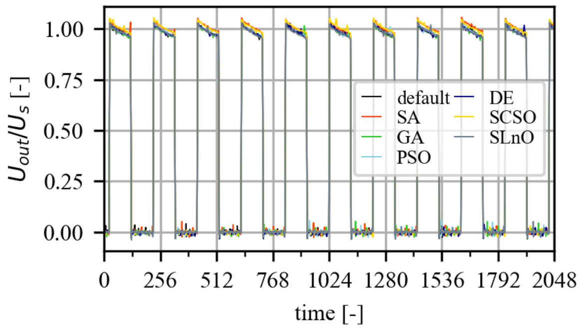

Fig. 12. Generator output voltage waveform in Case 2.

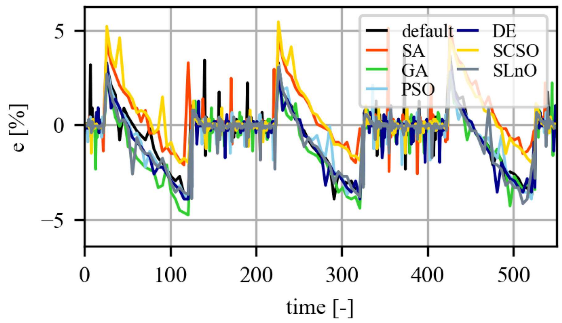

Fig. 13. Relative control error obtained in Case 2.

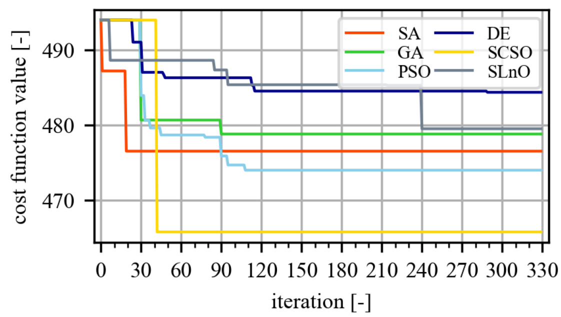

Fig. 14. Convergence of cost functions in Case 2.

As can be seen in Figs. 12–13 and Table VII, the SCSO algorithm achieves the best performance in terms of the cost function and the lowest steady-state error. It is worth mentioning that the SA metaheuristic algorithm also provides similar results, but with a worse steady-state error.

As can be seen in Fig. 14, the SCSO convergence curve of the cost function shows the greatest rate of decrease among all others and falls to the lowest value compared to the other algorithms, thus demonstrating its efficiency in the studied case. The SA algorithm also demonstrates rapid convergence in the initial stages; however, it becomes trapped in a local minimum, preventing it from achieving a globally optimal final result.

The output voltage and control error for the studied reference Case 3 are shown in Figs. 15 and 16, respectively.

The collected values of performance indices obtained experimentally in Case 3 are depicted in Table VIII. The obtained values in this case show that the SCSO once again achieves the best performance in terms of cost function.

TABLE VIII

Experimental results of the reference Case 3.

Parameter	Unit	Manual	SA	GA	PSO	DE	SCSO	SLNO
K_P	–	0.5	5.04991	5.42544	5.42544	6.07837	6.26104	6.45068
K_I	–	0.003	0.00106	0.01094	0.07492	0.0008	0	0.00209
C	–	78851313	2300	2593	2686	2438	2249	2318
e_m	%	13.73	14.41	11.85	12.99	14.19	14.49	15.39
e_{st}	%	–	−0.365	−3.246	−2.393	−1.9047	−0.289	−0.065
t_s	%	–	7.501	4.499	5.299	5.6	7.299	7.2

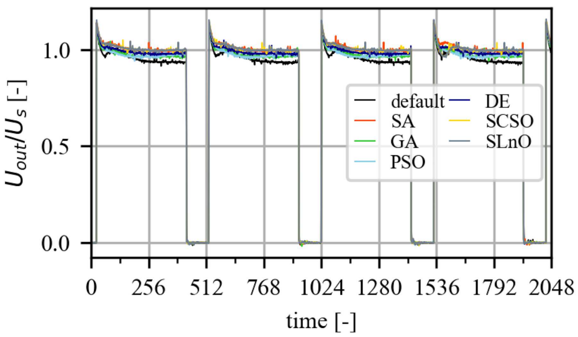

Fig. 15. Generator output voltage waveform in Case 3.

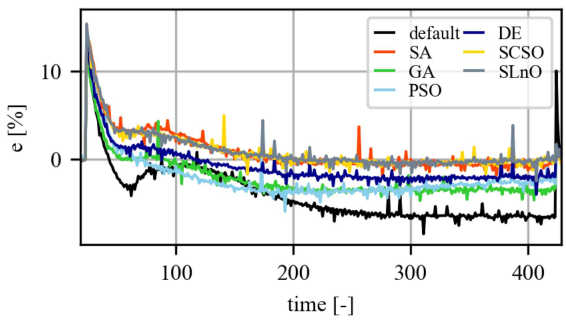

Fig. 16. Relative control error obtained in Case 3.

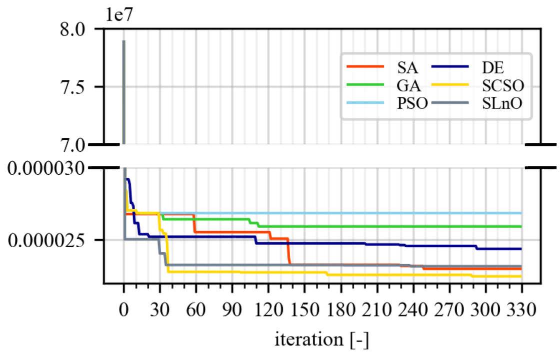

Fig. 17. Convergence of cost functions in Case 3.

As can be seen in Fig. 17, the SCSO convergence curve of the cost function falls to the lowest value compared with the other algorithms, thus exhibiting its efficiency in the studied case. The SLnO and SA algorithms demonstrate rapid convergence however, they get worse results in terms of settling time.

4. Discussion of results

This section aims to compare and analyse the control quality indices obtained for all six studied metaheuristic nature-inspired algorithms applied for tuning the PI controller of a high-voltage generator.

Figure 18 reveals the relation between cost function and achievable overshoot. It turns out that the lower the cost function, the lower the average overshoot e_m is. Specifically, the cost function tends to balance the performance and reduce e_m more effectively compared to IAE and MSE.

The MSE cost function results in moderate e_m values, while the IAE tends toward high averaged overshoots. Therefore, in order to minimize the maximum overshoot, the analysis of the cost function C may be advantageous.

Figure 19 shows the correlations between the values of the cost function and steady-state error. The IAE consistently displays the lowest average steady-state error. This indicates that the IAE function is particularly useful when striving for minimizing steady-state error. In contrast, the MSE and C cost functions generally result in higher average e_{st} values. Therefore, to minimize the average steady-state error, the analysis of the IAE cost function may be the preferred choice.

Figure 20 shows the correlations between values of cost functions and settling time t_s. Here, the cost function C stands out as the most effective, when aiming for the shortest average settling time values. Therefore, to minimize the average settling time, the analysis of the cost function C may be the best choice.

The analysis revealed that the C cost function generally brings the most balanced performance across the control quality indices e_m, e_{st}, and t_s.

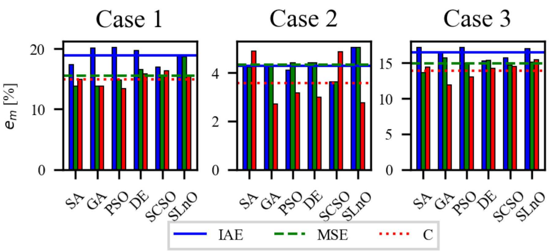

Fig. 18. Summary of maximum overshoot obtained in the performed experiments.

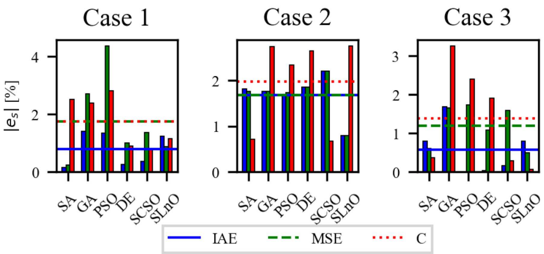

Fig. 19. Summary of steady-state error obtained in the performed experiments.

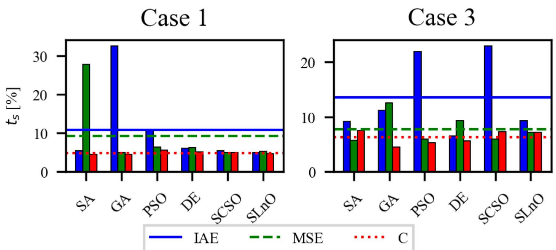

Fig. 20. Summary of settling times obtained in the performed experiments.

Specifically, the C function is effective in minimizing the e_m and t_s parameters, but performs worse with e_{st}. To achieve the lowest e_{st}, the IAE cost function is recommended.

5. Conclusions

The comparative study of nature-inspired metaheuristic algorithms for tuning the PI controller settings of a high-voltage generator demonstrates the superiority of these techniques over manual tuning. The SCSO and SLnO algorithms consistently showed superior performance across the three different benchmark cases studied in this paper. The SLnO algorithm exhibited the fastest convergence rate in Case 1, indicating its potential for rapid optimisation in similar scenarios.

The comparative analysis indicates that the choice of cost function and optimisation algorithm significantly impacts the control quality indices. The cost function generally provided the most balanced performance across control quality indices, effectively minimizing both overshoot and settling time. In turn, the IAE cost function was particularly effective in minimizing the steady-state error, making it a preferred choice when this parameter is critical.

These findings underscore the importance of selecting appropriate optimisation algorithms tailored to specific processes. For problems similar to those studied, SCSO and SLnO are recommended for their consistent performance in minimizing cost functions and rapid convergence, respectively. When balancing overall performance across multiple indices, the C cost function defined in this paper is advantageous, while the IAE cost function is preferable for applications where steady-state error minimization is paramount.

Future research should include multiple runs of each algorithm to provide statistical validation of the findings. Additionally, exploring hybrid metaheuristic approaches and more sophisticated parameter tuning methods hopefully could further enhance optimisation performance. Incorporating real-time adaptive mechanisms might also address the dynamics of high-voltage generator control, leading to more robust and reliable solutions.

References

[1] J.G. Ziegler, N.B. Nichols, *Trans. ASME* **64**, 759 (1942).

[2] G.H. Cohen, G.A. Coon, *Trans. ASME* **75**, 827 (1953).

[3] K.J. Åström, T. Hägglund, *PID Controllers: Theory, Design, and Tuning*, 2 ed., Instrument Society of America Research, Triangle Park (NC) 1995.

[4] S. Bharat, A. Ganguly, R. Chatterjee, B. Basak, D.K. Sheet, A. Ganguly, *AJCT* **5**, 1 (2019).

[5] M. Bartyś, *Energies* **13**, 4604 (2020).

[6] J. Pearl, *Heuristics: Intelligent Search Strategies for Computer Problem Solving*, Addison-Wesley Longman Publishing, 1984.

[7] A. Jayachitra, V. Rajendran, *Adv. Artif. Intell.* **2014**, 791230 (2014).

[8] M.J. Neath, A.K. Swain, U.K. Madawala, D.J. Thrimawithana, *IEEE Trans. Power Electron.* **29**, 1523 (2014).

[9] M.I. Solihin, L.F. Tack, L.K. Moey, in: *Proc. of the Int. Conf. on Advanced Science, Engineering and Information Technology*, 2011.

[10] I. Chiha, J. Ghabi, N. Liouane, in: *5th Int. Symp. on Communications Control and Signal Processing, (ISCCSP 2012)*, 2012.

[11] V.T. Aghaei, A.S. Abbasi, J. Rasheed, A.M. Abu-Mahfouz, *Heliyon* **9**, e13885 (2023).

[12] R. Masadeh, B. Mahafzah, A. Sharieh, *Int. J. Adv. Comput. Sci. Appl.* **10**, 388 (2019).

[13] A.Y. Jaen-Cuellar, R. de J. Romero-Troncoso, L. Morales-Velazquez, R.A. Osornio-Rios, *Int. J. Adv. Robot. Syst.* **10**, 1 (2013).

[14] D. Ochs T. Rettich, *Vak. Forsch. Prax.* **18**, 32 (2006).

[15] Y. Jin, Y. Li, C. Jiang, H. Yu, *IEEE Access* **11**, 141394 (2023).

[16] M. Abachizadeh, M.R.H. Yazdi, A. Yousefi-Koma, in: *2010 IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, 2010, p. 379.

[17] A. Karimi, H. Eskandari, M. Sedighizadeh, A. Rezazadeh, A. Pirayesh, *Int. J. Tech. Phys. Prob. Eng.* **5**, 123 (2013).

[18] V. Rajinikanth, K. Latha, *Appl. Computat. Intell. Soft Comput.* **2012**, 214264 (2012).