Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2017) 25: 1300 – 1310

doi:10.3906/elk-1511-252

Comparison of using the genetic algorithm and cuckoo search for multicriteria optimisation with limitation

Ryszard KLEMPKA[1,*], **Bogusław FILIPOWICZ**[2]
[1]Department of Power Electronics and Energy Control Systems, Faculty of Electrical Engineering, Automatics, Computer Science, and Biomedical Engineering, AGH University of Science and Technology, Krakow, Poland
[2]Polytechnic Institute, State Higher Vocational School in Tarnow, Tarnow, Poland

Received: 26.11.2015 • **Accepted/Published Online:** 26.04.2016 • **Final Version:** 10.04.2017

Abstract: The article presents an example of using two optimisation methods, a genetic algorithm and cuckoo search, to identify parameters of electric drive controllers using some quality criteria and by applying a limitation to the maximum values of signals in the controlled facility. The results for both optimisation methods are compared. The impact of the probability that the nest host discovers the laid eggs on the speed of finding the optimum solution is investigated.

Key words: Genetic algorithms, cuckoo search, optimisation, DC drive control

1. Introduction

The basic goal of control is to obtain a response of an object according to the design assumptions, i.e. to have the object response error as small as possible against the requested response. Making a design that works in a real system is an important aspect of designing. This means designing controls for a mathematical model of a system that will work correctly with a real object despite making assumptions simplifying the design and when the object contains nonmodelled dynamics. It is possible to achieve this goal by designing a controller according to the H_∞ norm [1]. Unfortunately, these goals tend to exclude one another. Reaching the first goal makes the other one unachievable and vice versa. That is why it is necessary to find a compromise between the goals, i.e. to use multicriteria optimisation.

It is a relatively easy task to design a control system for linear objects. It becomes more complicated when the controlled object is nonlinear and additionally if it is necessary to take into account different limitations for real signals.

This all makes designing an optimum control a complex and difficult process, where artificial intelligence methods have been used more and more often [2,3]. The need to find the optimum of a function that tends to be nonlinear, noncontinuous, and composed of several quality criteria as well as considering additional limitations makes the search for the solutions extremely extensive, and there are either no tools for an analytical search for the solution or the calculation time is unacceptable. In such cases it is sensible to use artificial intelligence methods, such as genetic algorithms (GAs) or a cuckoo search (CS) algorithm.

*Correspondence: klempka@agh.edu.pl

2. Controlled object

A DC drive with two secondary control loops with PI type controllers was selected as the test object (Figure 1). Both the control and the controller structure can be selected using different methods [4,5].

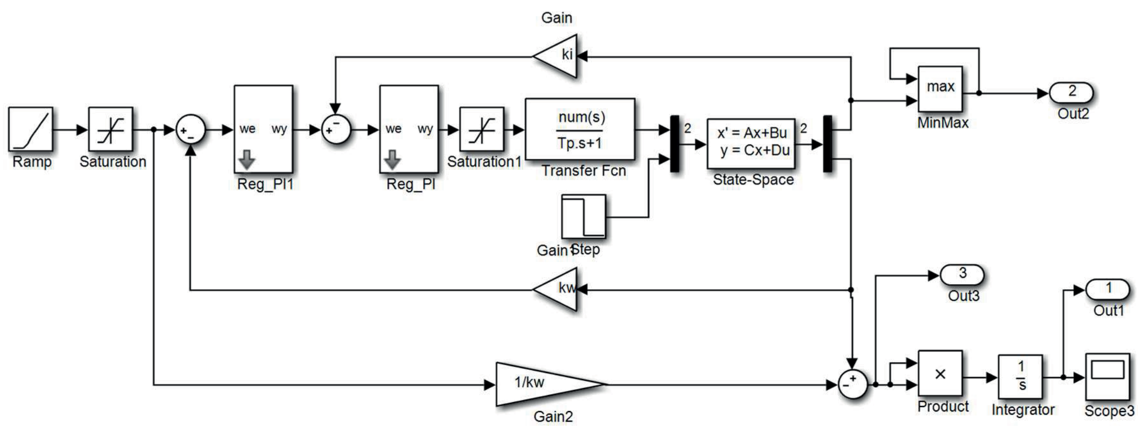

Figure 1. Control system with a DC motor.

The purpose of this paper is to compare two optimisation methods from the group of artificial intelligence methods. That is why a classic control system was selected.

The motor is described by the following system of equations.

$$
\begin{aligned}
\begin{bmatrix} \frac{di}{dt} \\ \frac{d\omega}{dt} \end{bmatrix} &= \begin{bmatrix} -\frac{R}{L} & -\frac{k}{L} \\ \frac{k}{J} & 0 \end{bmatrix} \cdot \begin{bmatrix} i \\ \omega \end{bmatrix} + \begin{bmatrix} \frac{1}{L} & 0 \\ 0 & -\frac{1}{J} \end{bmatrix} \cdot \begin{bmatrix} u \\ M \end{bmatrix} \\
\begin{bmatrix} i \\ \omega \end{bmatrix} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} i \\ \omega \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} u \\ M \end{bmatrix}
\end{aligned}
\tag{1}
$$

Where:

u – armature voltage

R – armature resistance

i – armature current

L – armature inductance

k – constant coefficients dependent on the motor structure and excitation current in the SI units system that make the flux linkage

M – mechanical torque

J – moment of inertia referred to the motor shaft

ω – angular speed of the motor

The motor is powered from a 6-pulse converter, which can be regarded (making a simplification) as a first-order inertial element with the following transmittance:

$$G_p = \frac{K_p}{1 + sT_p} \tag{2}$$

Where:

K_p – converter gain

T_p – converter time constant

The supply voltage of the motor must not be higher than the voltage corresponding to full controlling of the converter, and that is why the maximum value of the converter controlling signal is limited.

Both controllers are of the PI type. The current controller (G_I(s)), which is controlled by the difference between the output signal from the speed controller and the real current signal, develops a signal sent to the converter supplying the motor. The speed controller (G_Ω(s)) is controlled by the difference between the setting signal and the measured speed signal. The transmittances of the controllers are described by Eq. (3).

$$\begin{array}{l} G_I(s) = \frac{k_{Ip} \cdot s + k_{Ii}}{s} \\ G_\Omega(s) = \frac{k_{\Omega p} \cdot s + k_{\Omega i}}{s} \end{array} \tag{3}$$

The setting signal completes the function in Eq. (4).

$$u_{control}(t) = \begin{cases} \alpha \cdot t + u_0 & \text{for} \quad u_{control}(t) < u_{\max} \\ u_{\max} \end{cases} \tag{4}$$

The motor torque is regarded as an interference that has an adverse impact on the motor speed. During simulation of the model, the torque is changed in steps after 7 s, from 140 Nm to 60 Nm.

The parameters of the motor and auxiliary elements are as follows:

$P_N = 22$ kW, $U_N = 440$ V, $I_N = 56.2$ A, $J = 2.7$ kgm^2, $R = 0.465\ \Omega$, $L = 15.345$ mH,

$n_N = 1500$ rpm, $k = 2.62$, $\omega_N = 157$ s^{-1}, $K_p = 100$, $T_p = 1.67$ ms, $k_i = 0.1$, $k_w = 0.05$,

$\alpha = 2.7$ V/s, $u_0 = 0.01$ V, $u_{\max} = 7.85$ V.

Figure 1 additionally contains the MinMax block with feedback, detecting the maximum value of the motor current and the speed. During start-up under full load, the current must not exceed twice the nominal value of the current, i.e. 112.4 A. The speed is checked due to the need to reach the rated speed during simulation.

Another additional element in Figure 1 is a collection of blocks calculating one of the optimisation criteria, i.e. an integral from the square value of the set and measured motor speed error (Eq. (5)).

$$y_1 = \int_0^{10} \left(\omega - \frac{u_{control}}{k_w} \right)^2 dt \tag{5}$$

In order to identify the second criterion, the maximum value is calculated using the Bode characteristics determined for the lines from the interfering inputs, i.e. from the load torque and from two inputs measuring the speed and current. The values correspond to the value of the H$_\infty$ norm (Eq. (6)).

$$y_2 = \max \left(\| G \left(j \cdot \omega \right) \| \right) \tag{6}$$

3. Selecting parameters of controllers using the genetic algorithm and cuckoo search

The purpose of the optimisation is to find four parameters of controllers (two parameters for the current controller and two for the speed controller), for which the functions of the objective will successively reach the following minimum:

$$y_1 = \int_0^{10} \left(\omega - \frac{u_{control}}{k_w} \right)^2 dt \tag{7}$$

$$y_2 = \max\left(\left\| G\left(j \cdot \omega\right) \right\|\right) \tag{8}$$

$$y_3 = 10 \cdot y_1 + y_2 \tag{9}$$

Owing to the third function (Eq. (9)), it will be possible to reach a compromise between function one (Eq. (7)) and two (Eq. (8)), which exclude one another. In every optimisation case the maximum value of the current has to be limited and the rated speed achieved. Each function will be minimised both by the genetic algorithm and the cuckoo search.

3.1. Genetic algorithm

The GA is one of the tools proposed for performing optimisation; it belongs to the group of approximate methods [6]. The method is inspired by nature. It is based on a population of solutions that evolves in every subsequent generation, striving to ensure continuous improvement of its individuals. A better individual will have more offspring, which means that an individual that is better adapted will survive. Individuals are improved just as in nature by selection, crossover, and mutations.

Figure 2 shows the basic block diagram of the genetic algorithm.

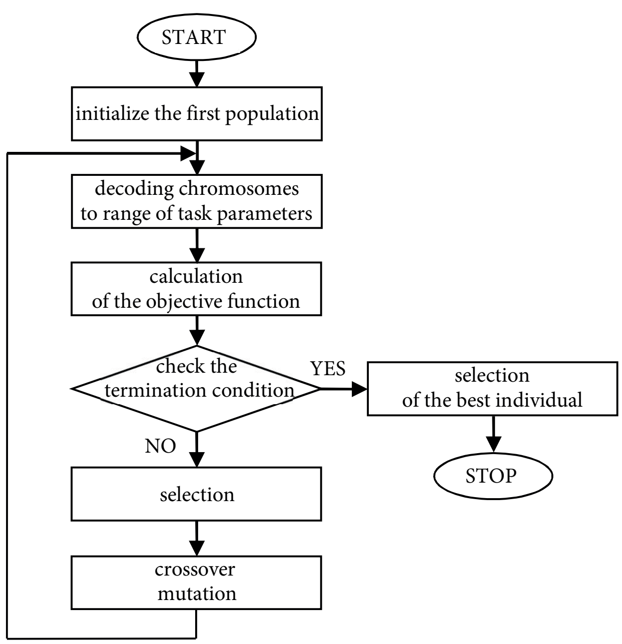

Figure 2. Block diagram of the basic genetic algorithm.

Since the initial research on the form of GAs and components, they have undergone modifications and the current form of the basic GA includes:

- initialisation of the first population,

- decoding the chromosomes to the task parameters,
- determining the value of the objective function for each individual,
- checking the condition of ending the algorithm,
- selection,
- crossover and mutation.

Optimisations of the parameters of controllers for DC drives with a GA were already described [7], whereby the limitation for the maximum value of the current was not taken into consideration.

All optimisations using the GA were made with the following configuration:

- range of variation of decision variables under the terms of the task:
  $k_{\Omega p}$, $k_{\Omega i}$, k_{Ip}, k_{Ii} = (0–2000),
- parameters are coded in a string of 15 bits for each,
- populations of 200 individuals,
- probability of crossover $p_k = 0.95$,
- probability of mutation $p_m = 0.01$,
- condition of termination of work AG - 100 generations,
- ranking with the factors: $C_{min} = 0$, $C_{\max} = 2$,
- selection method - stochastic universal sampling,
- crossing the shuffle.

4. Cuckoo search algorithm

The CS algorithm is the second optimisation method under consideration. Similarly to the GA, it is inspired by nature and belongs to the group of algorithms based on the population of solutions. It should be highlighted that the bee algorithm, the firefly algorithm, and the cockroach optimisation algorithm also belong to the same class of algorithms [8,9]. Cuckoos lay their eggs in the nests of other birds so that they hatch their eggs and feed their nestlings. It can happen that the host recognises the alien egg. Then the egg is thrown out of the nest or the host abandons the whole nest and builds a new one. Some cuckoos specialise in making their eggs similar in colour and size to the eggs of specific species of birds, which makes it difficult for the host to recognise the alien egg.

Every egg identifies one potential solution for the set optimisation task. Information on all decision variables are recorded in an egg.

The following rules are applicable for the CS algorithm:

- every cuckoo lays one egg and lays it in a randomly selected nest,
- only the best egg in the nest will survive by the next generation,

- the number of nests is fixed,
- the laid egg is detected by the host with probability $p_a \in [0,\ 1]$. If the laid egg is detected, the host either throws it away or leaves the whole nest and randomly builds a new nest.

The CS algorithm was first presented by Yang and Deb in 2009 [10].

The Cuckoo Search algorithm is presented in Figure 3.

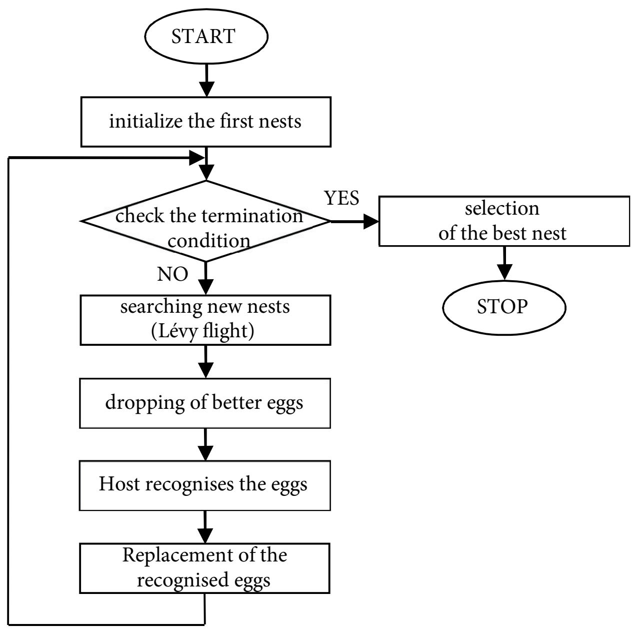

Figure 3. Block diagram of the cuckoo search.

At the beginning of the algorithm, a random set of eggs is generated. Each egg is evaluated for how good the solution it represents is.

Then new random cuckoo eggs are generated. For the needs of this article Lévy flights [11] by Mantegna's algorithm (Eq. (10)) were used.

$$\nu = \frac{x \cdot \sigma\left(\beta\right)}{\left|y\right|^{\frac{1}{\beta}}} \tag{10}$$

Where:

$$\sigma\left(\beta\right) = \left(\frac{\Gamma\left(1+\beta\right) \cdot \sin\left(\frac{\pi \cdot \beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \cdot \beta \cdot 2^{\frac{\beta-1}{2}}}\right)^{\frac{1}{\beta}} \tag{11}$$

$$\Gamma\left(\alpha\right) = \int\limits_0^\infty \left(e^{-t} \cdot t^{x-1}\right) dt \tag{12}$$

$$A = 0.01 \cdot z \cdot \nu \cdot \left(s - best\right) \tag{13}$$

$$s^{new} = s + A \tag{14}$$

x, y, z – random numbers with normal distribution,

s – nest currently under consideration,

$best$ – nest with the best egg,

A – identified cuckoo's flight route,

s^{new} – new nest of the cuckoo.

Eqs. (13) and (14) identify the new nest found by the cuckoo. Owing to the algorithm, the new egg can potentially generate a better solution. If so, the cuckoo replaces the host's egg in the nest with its own one.

Now there is time for the nest host, which tries to recognise the laid eggs. The host does it with probability p_a. New random solutions similar to the existing ones are generated to replace the recognised eggs if they demonstrate a better solution.

$$s^{new} = s + k \cdot \left(s^{rand} - s\right) \tag{15}$$

Where:

k – random number from the [0, 1] range,

s^{rand} – randomly selected nest.

When the whole cycle is finished, another generation of cuckoos is generated, coming back to the generation of new eggs using Lévy flights.

The optimisation algorithm with the CS method was developed quite recently. Work on improving the algorithm is under progress [12]. Despite the above, the CS was tested in practical applications [13–19].

The CS algorithm has been configured in the following way for this paper:

- range of variation of decision variables under the terms of the task:

  $k_{\Omega p}$, $k_{\Omega i}$, k_{Ip}, k_{Ii} = (0–2000),

- condition of termination of work CS - 200 generations,

- number of nests - 200,

- probability of detecting a laid egg p_a = 0.25,

- $\beta = 3/2$.

The quality of the proposed solution in both the GA and CS is identified using a computer simulation of the operation of the system presented in Figure 1 in the MATLAB/Simulink package, which significantly extends the process of optimisation.

4.1. Results

Optimisation using the GA and CS was performed, where Eqs. (7)–(9) were minimised one after another. One should remember that the minimisations were performed with limiting of the maximum motor starting current and the condition that the motor has to reach the rated speed. The Table presents the obtained results.

The presented results confirm that the criteria in Eqs. (7) and (8) exclude one another and their combination in the criterion of Eq. (9) helps find a compromise between them.

Table. List of obtained optimisation results.

Minimisation	Method	$k_{\Omega p}$	$k_{\Omega i}$	k_{Ip}	k_{Ii}	y_1	y_2	y
y_1	AG	312.64	1532.2	4.39	5.87	0.0680	450.43	0.0680
	CS	325.98	1637.9	1.13	58.56	0.0602	317.44	0.0602
y_2	AG	6.16	0.488	1.96	38.12	3106	4.76	4.76
	CS	11.82	0.0758	2.01	92.28	1269.7	9.76	9.76
y_3	AG	119.93	214.22	0.49	0.78	1.60	62.95	78.49
	CS	113.82	199.91	0.51	12.72	1.436	64.73	79.09

The results obtained using both methods are not identical despite many minimisation attempts. This is mainly due to the fact that they are stochastic methods.

Figures 4 and 5 present the curves for the speed error and motor current, respectively, resulting from the solutions found by both algorithms.

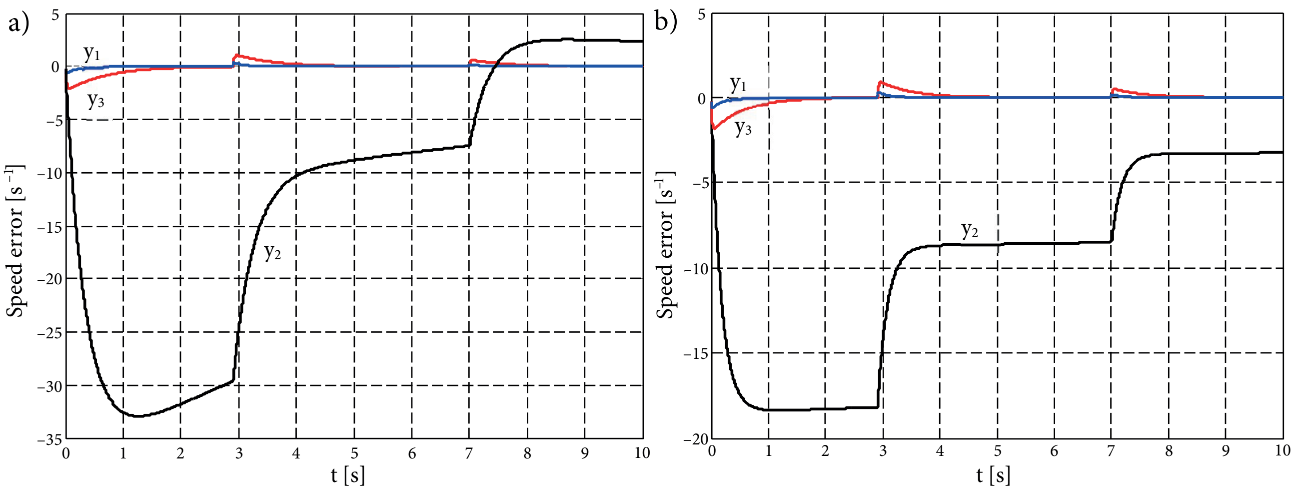

Figure 4. Speed error curves: a) AG, b) CS.

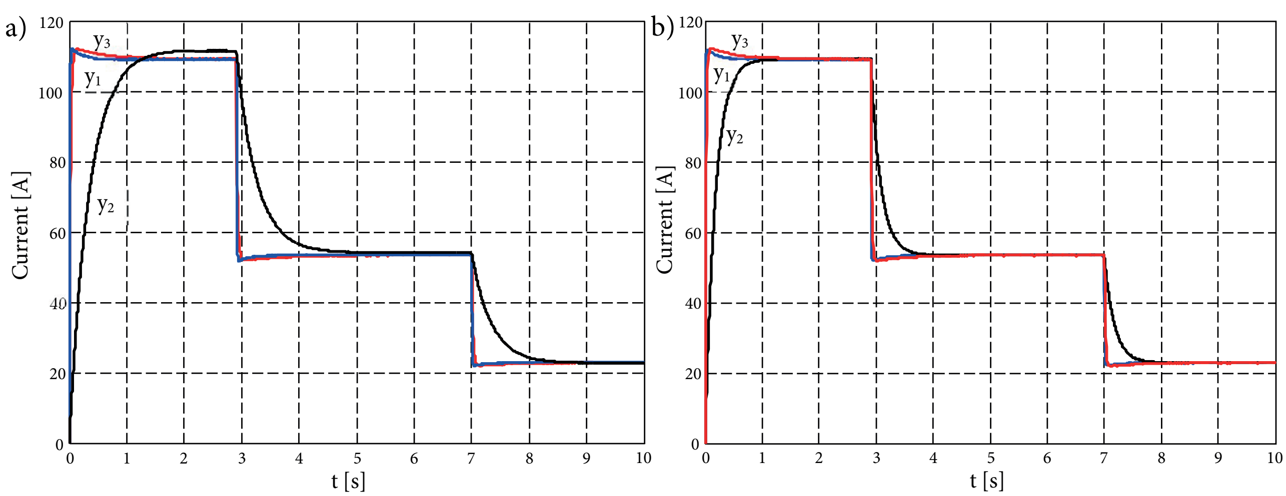

Figure 5. Motor current curves: a) AG, b) CS.

One should pay attention to the fact that optimisation with the CS algorithm takes twice as much time as for the GA. This results from twice the number of simulations carried out in Simulink to evaluate the quality of the proposed solutions during one generation, that is from the structure of the algorithm itself. A second reason for the longer computational process is twice the number of generations.

Figure 6 presents a diagram of the minimum values during objective function optimisation (Eqs. (7)–(9)) for both methods. The curves were plotted for the best case recorded during multiple optimisation.

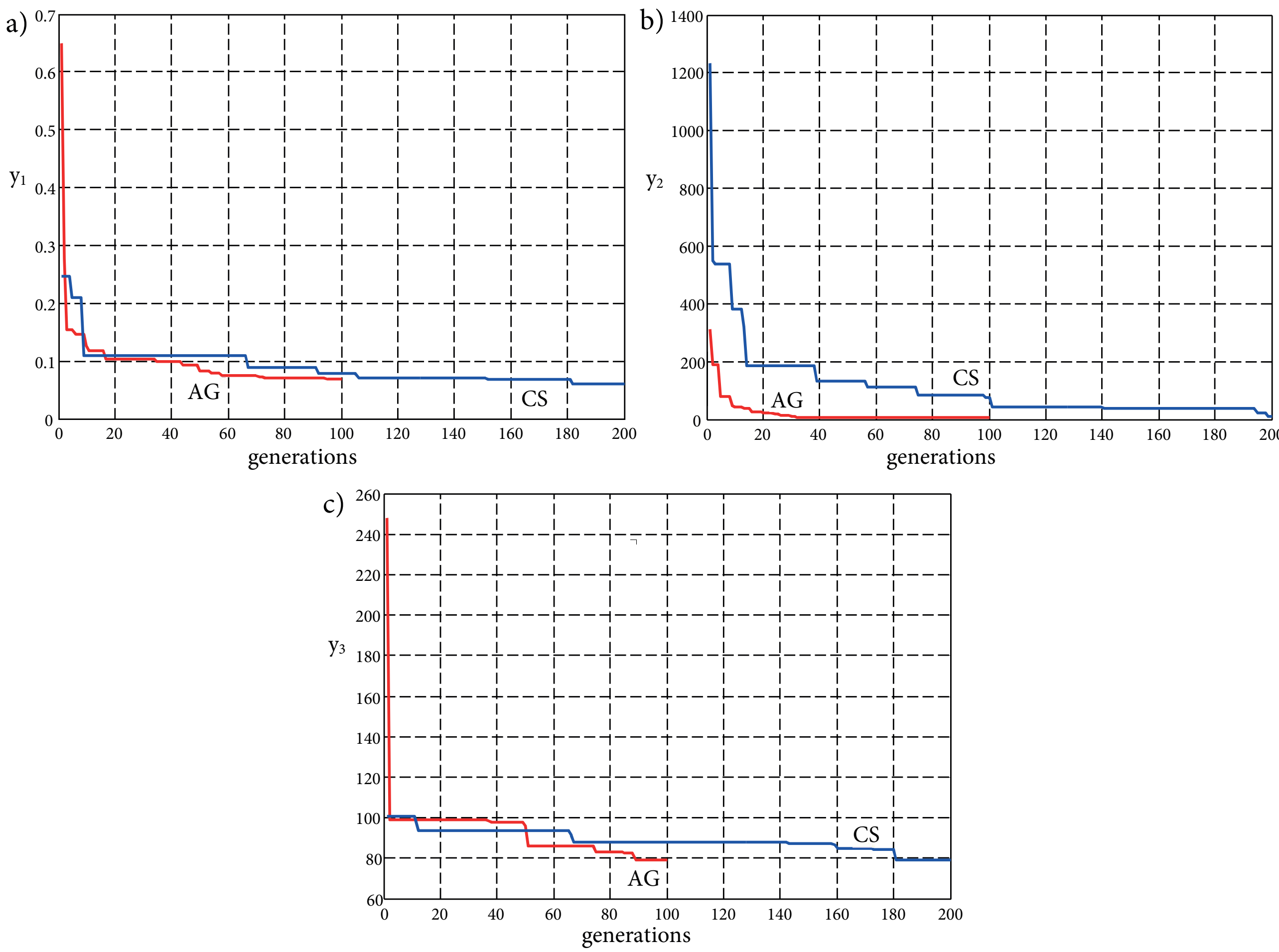

Figure 6. Minimum values of the objective function: a) y_1, b) y_2, c) y_3.

Figure 7 presents a process of searching for a solution with the CS algorithm for the objective function of Eq. (9) at different values of the probability of discovering the laid eggs. Registration was made using the CS algorithm for the searched four variables from the (0–250) range.

5. Conclusions

The paper presents an example of searching parameters for electric drive controllers. Optimisation of two criteria excluding one another has been made, as well as a criterion that is a compromise of the first two factors. Additionally, the optimisations were performed using limitations on the basic signals of the object. Two optimisation methods from the group of artificial intelligence methods were employed, i.e. the GA and

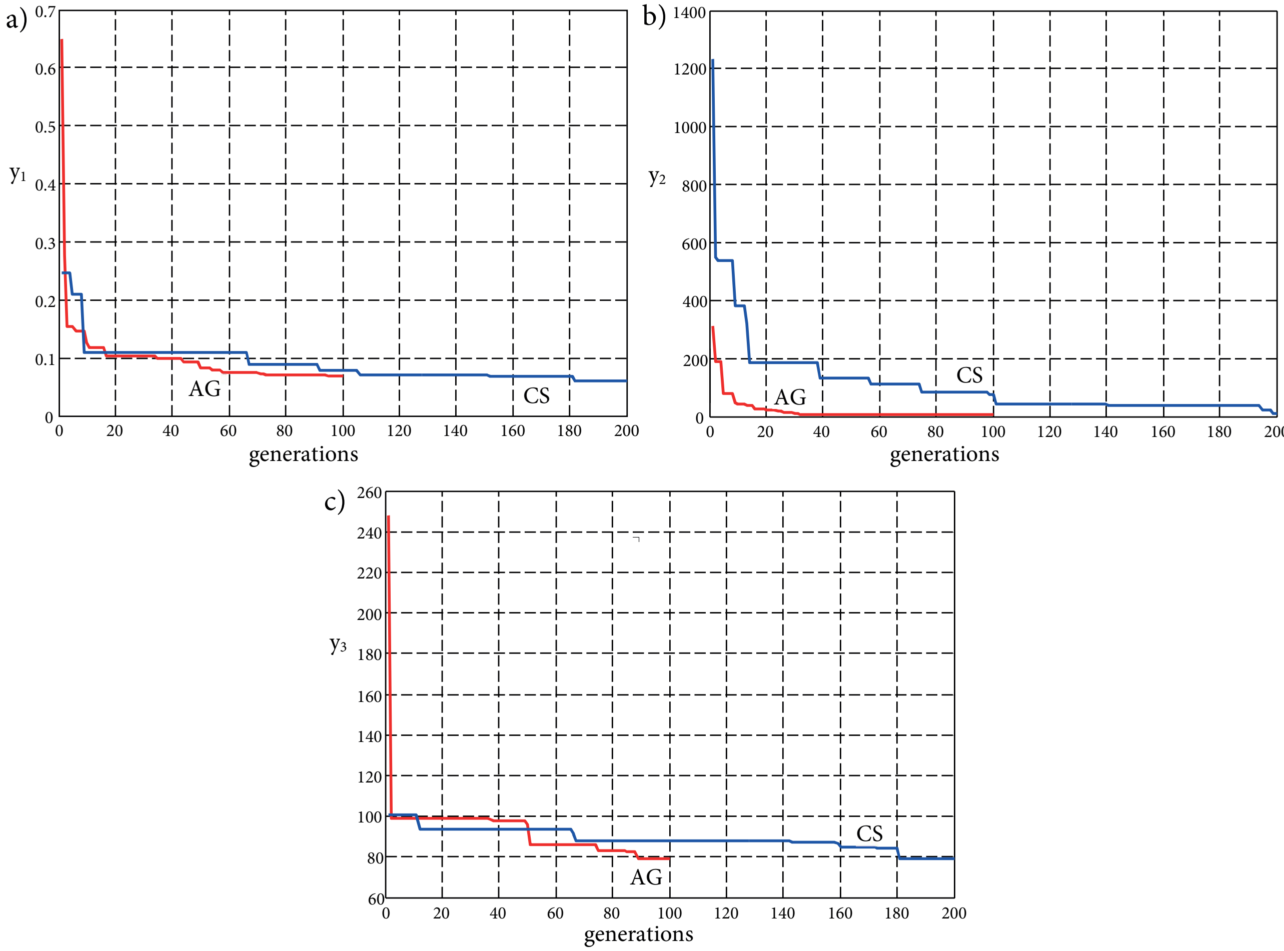

Figure 7. Curves of the minimum values during optimisation of the function of Eq. (9) with the CS algorithm, at different values of probability p_a.

CS with Lévy flight. The results are similar; however, there are certain differences that require attention. The calculation time of each generation of the CS algorithm is twice that for the GA. This results from the structure of the CS algorithm. Considering the number of generations, the CS also turned out to be slower than the GA.

Different optimisation times for different criteria can also be observed. For the y_1 function (minimising the square value of the motor speed error), both algorithms find similar solutions within a similar time (generation). For the y_2 criterion (H_∞ norm) the CS clearly needs more time. The rule for criterion y_3 is similar; however, it proves better than for the y_2 function.

Optimisations using the CS were also performed for different probabilities of detection of the laid eggs. The CS algorithm with high probability (i.e. 75% and 100%) turned out to be much poorer at searching for the solution.

It should be emphasised that both optimisation methods (GA and CS) found acceptable solutions; however, CS was the slower algorithm in the presented example.

References

[1] Doyle JC, Glover K, Khargonekar PP, Francis BA. State-space solutions to standard H_2 and H_∞ control problems. IEEE T Autom Control 1989; 34: 831-847.

[2] Yi T, Xin Z, Liang Z. Fuzzy-genetic control strategy of hybrid electric vehicle. In: Second International Conference on Intelligent Computation Technology and Automation; 10–11 October 2009; Changsha, China. pp. 720-723.

[3] Juang JG, Huang MT, Liu WK. PID control using presearched genetic algorithms for a MIMO system. IEEE T Syst Man Cy C 2008; 38: 716-727.

[4] Sieklucki G, Sykulski R, Orzechowski T. Application of incremental encoder in direct field oriented control of permanent magnet synchronous motor. Prz Elektrotechniczn 2010; 86: 216-220.

[5] Sieklucki G. Analysis of the transfer-function models of electric drives with controlled voltage source. Prz Elektrotechniczn 2012; 88: 250-255.

[6] Goldberg D. Genetic Algorithms in Search, Optimisation and Machine Learning. Warsaw, Poland: Wydawnictwa Naukowo-Techniczne, 1995 (in Polish).

[7] Klempka R. Selection of a drive controllers' parameters using genetic algorithm and different quality criteria. Prz Elektrotechniczn 2013; 89: 125-130.

[8] Kwiecień J, Filipowicz B. Comparison of firefly and cockroach algorithms in selected discrete and combinatorial problems. B Pol Acad Sci-Te 2014; 62: 797-804.

[9] Civicioglu P, Besdok E. A conceptual comparison of the cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms. Artif Intell Rev 2013; 39: 315-346.

[10] Yang XS, Deb S. Cuckoo search via Lévy flights. In: World Congress on Nature & Biologically Inspired Computing; 9–11 December 2009; Coimbatore, India. pp. 210-214.

[11] Yang XS. Nature-Inspired Metaheuristic Algorithms. 2nd ed., Bristol, UK: Luniver Press, 2010.

[12] Civicioglu P. Comparative analysis of the cuckoo search algorithm. In: Yang XS, editor. Cuckoo Search and Firefly Algorithm: Theory and Applications. Zurich, Switzerland: Springer International Publishing, 2014. pp. 85-113.

[13] Kishnani M, Pareek S, Gupta R. Optimal tuning of PID controller by cuckoo search via Lévy flights. In: International Conference on Advances in Engineering and Technology Research; 1–2 August 2014; Unnao, India. pp. 1-5.

[14] Sahithullah M, Kumar AS, Kavin KS. Shunt active filter using cuckoo search algorithm for PQ conditioning. In: International Conference on Circuit, Power and Computing Technologies; 19–20 March 2015; Nagercoil, India. pp. 1-7.

[15] Zexi D, Feidan H. Cuckoo search algorithm for solving numerical integration. In: International Conference on Cyber Technology in Automation, Control and Intelligent Systems; 8–12 June 2015; Shenyang, China. pp. 1508-1512.

[16] Rangasamy S, Manickam P. Stability analysis of multimachine thermal power systems using the nature-inspired modified cuckoo search algorithm. Turk J Electr Eng Co 2014; 22: 1099-1115.

[17] Yildiz AR. Cuckoo search algorithm for the selection of optimal machining parameters in milling operations. Int J Adv Manuf Tech 2013; 64: 55-61.

[18] Durgun İ, Yildiz AR. Structural design optimization of vehicle components using Cuckoo search algorithm. Materials Testing 2012; 54: 185-188.

[19] Yildiz AR. A comparative study of population-based optimization algorithms for turning operations. Inform Sciences 2012; 210: 81-88.