

Łukasz KNYPIŃSKI\*, Sebastian KUROCZYCKI\*, Milena KURZAWA\*

## **OPROGRAMOWANIE DO WYZNACZANIA KSZTAŁTU IMPULSU NAPIĘCIOWEGO ZASILAJĄCEGO SILNIK BLDC WYKORZYSTUJĄCE METODĘ POSZUKIWANA KUKUŁCZEGO**

W pracy przedstawiono algorytm oraz oprogramowanie do wyznaczania kształtu impulsu napięciowego zasilającego pasma stojana bezszczotkowego silnika prądu stałego (BLDC). Aplikacja została napisana w języku Python 3.8. Składa się z dwóch niezależnych modułów: (a) procedury optymalizacyjnej oraz (b) modelu matematycznego silnika BLDC. Model matematyczny silnika wykorzystuje uproszczony model o parametrach skupionych. Efektywność metody poszukiwania kukułczego badano wyznaczając minimum globalne funkcji analitycznej. Porównano wyniki obliczeń symulacyjnych dla dwóch metod optymalizacji: (a) metody poszukiwania kukułczego oraz (b) metody wzorowanej na echolokacyjnym zachowaniu nietoperzy. Wyznaczono parametry impulsu napięciowego zapewniającego minimalizację współczynnika tętnień.

**SŁOWA KLUCZOWE:** optymalizacja, metoda poszukiwania kukułczego, bezszczotkowe silniki prądu stałego, model o parametrach skupionych.

### **1. WPROWADZENIE**

Od kilkunastu lat do rozwiązywania zadań optymalizacji obiektów technicznych wykorzystywane są niedeterministyczne metody optymalizacji [1-3]. Są one bardzo chętnie stosowane do rozwiązania zadań optymalnego projektowania w przypadku zastosowania złożonych matematycznie modeli w których jest trudno wyznaczyć pochodne cząstkowe [4-6]. W obszarze zagadnień dotyczących projektowania urządzeń elektromagnetycznych, najczęściej stosowane są modele bazujące na metodzie elementów skończonych (MES). Dokładność obliczeń modeli tego typu zależy od gęstości siatki dyskretyzującej oraz od długości kroku czasowego. Z drugiej strony zastosowanie bardzo dokładnej „dyskretyzacji” powoduje zwiększenie czasu rozwiązania pojedynczego wariantu projektowanego urządzenia [7]. Proces optymalizacji w którym wyznaczane są kolejne iteracje składające się grupy osobników są bardzo złożone obliczeniowo.

W metodach niedeterministycznych (heurystycznych) rozwiązania optymal-

---

\* Politechnika Poznańska.

nego poszukuje się w obrębie grupy osobników. Obecnie najbardziej dynamicznie rozwijającą się grupę metod stanowią metody oparte na zachowaniu zwierząt w ich środowiskach naturalnych. W przypadku tych metod proces optymalizacji jest realizowany przy wykorzystaniu reguł społecznych obowiązujących w grupie osobników, nazywanych w zależności od metody: pokoleniem (algorytmy genetyczne – AG), rojem (metody roju cząstek – PSO), kolonią (metoda „nietoperzy” – BA), a nawet watahą (metoda szarych wilków – GWO). Rozpatrując kolejne iteracje algorytmu znajdujemy rozwiązanie optymalne [8, 9].

Systematycznie wykonując obliczenia dla kolejnych iteracji i kolejnych osobników musimy dla każdego wariantu urządzenia wykonać obliczenie funkcji celu, czyli wielokrotnie wyznaczyć rozkład pola elektromagnetycznego dla optymalizowanego obiektu. Zastosowanie bardzo złożonego modelu matematycznego wydłuża czas uzyskania rozwiązania optymalnego. Dla metod heurystycznych, w których wyznaczenia każdego kolejnego położenia zależy od składnika losowego. Proces optymalizacji należy kilkakrotnie powtórzyć w celu uzyskania wiarygodnych wyników obliczeń, co wydłuża czas znalezienia rozwiązania „optymalnego” dla rozpatrywanego problemu.

W celu minimalizacji liczby wywołań funkcji celu, tożsamej z czasem obliczeń naukowcy poszukują coraz nowszych i bardziej efektywnych metod optymalizacji, które pozwoliłyby na uzyskanie rozwiązania optymalnego szybciej, przy wykonaniu mniejszej liczby iteracji algorytmu optymalizacji [11-13].

W artykule przedstawiono zastosowanie metody poszukiwania kukułczego do wyznaczenia kształtu impulsu zasilającego pasma stojana zapewniającą zadaną wartość tętnień momentu elektromagnetycznego na wale maszyny.

## 2. KUKUŁKA JAKO PASOŻYT LĘGOWY

Kukułka pospolita zamieszkuje tereny Europy i Azji. Dorosłe osobniki osiągną długość 28-32 cm oraz wagę około 120 gram. Dorosłe kukułki nie tworzą par. Obserwujemy, że w środowisku naturalnym, występuję więcej samców niż samic. Kukułki nie budują własnych gniazd i nie wysiadują własnych jaj, są drapieżnikami rozrodczymi. W celu zachowania ciągłości gatunku, kukułki podrzucają swoje jaja innym gatunkom ptaków. Samce obserwują ptaki innych gatunków podczas budowania gniazd i znoszenia jaj. Kiedy ptaki budujące gniazdo opuszczają je, samica usuwa jedno jajo i podrzuca własne. Jaja gospodarzy są wyrzucane i rozbijane [14]. Samice wybierają gniazda gatunków ptaków, których jaja są podobne do jaj kukułki, są to najczęściej: świergotki, plizki, trzciniaki oraz jar zębatki.

Należy zauważyć, że czasem podrzucane jaja są rozpoznane przez gospodarzy gniazda. Wówczas jajo kukułcze jest usuwane z gniazda. Zdarza się również, że ptaki gospodarze opuszczają swoje gniazdo, w którym znajduje się obce jajo. Dlatego w numerycznym odwzorowaniu algorytmu kukułki stosuje się z zada-

nym prawdopodobieństwem „budowanie” nowych gniazd w obszarze dopuszczalnych rozwiązań analizowanego problemu optymalizacji.

Młode kukułki wylęgają się po około 11 dniach, często szybciej niż pisklęta gospodarza. Młoda kukułka po wykluciu się plecami wypycha z gniazda jaja gospodarzy. W rzeczywistości pisklęta kukułek są większe od piskląt gospodarzy, potrzebują więcej miejsca w gnieździe oraz więcej pokarmu. Wspólnie z pisklętami ptaka gospodarza nie mogłyby przeżyć, ponieważ „przybrani” rodzice nie byłoby w stanie dostarczyć wystarczającej ilości pokarmu dla wszystkich piskląt. Pisklęta kukułek są bardzo żarłoczne w stosunku do piskląt gospodarza gniazda. Niejednokrotnie podrzucone pisklęta głośno nawołują „przybranych rodziców”, domagając się więcej żywności. Głośniejsze nawoływanie w stosunku do piskląt gospodarzy gniazda powodują, że często pisklęta tego gatunku padają ofiarą ptaków drapieżnych. Naukowcy szacują, że około 1/3 złożonych jaj ginie przed osiągnięciem wieku dorosłego. Dlatego samice kukułek składają więcej jaj niż samice innych gatunków ptaków.

### 3. METODA POSZUKIWANIA KUKUŁCZEGO

Metoda poszukiwania kukułczego (CS) została opracowana w 2009 przez Xin-She Yanga oraz Suash Deb'a [15]. Powstała na podstawie odwzorowania rozrodczego zachowania kukułki zwyczajnej. W środowisku naturalnym kukułka jest drapieżnikiem rozrodczym. Swoje jaja podrzuca do gniazd innych ptaków. Przy czym kukułki mogą postępować w różny sposób: (a) podrzucanie swojego jaja do gniazda innych ptaków, po wcześniejszym usunięciu z gniazda jaja właściciela lub (b) zdarza się, że kukułki mogą przejmować gniazdo właściciela dla własnych celów rozrodczych [16].

Podstawy teoretyczne klasycznego algorytmu kukułki są opracowane według następujących zasad:

- Gniazda z najlepszymi jajami (osobnikami) są przenoszone do następnej iteracji algorytmu;

- Liczba gniazd ptaków gospodarzy jest zadawana przed rozpoczęciem procesu optymalizacji. Zakłada się, że również gospodarz może odnaleźć i rozpoznać „podrzucone” jajo (rozwiązanie analizowanego problemu). Dlatego przyjmowane jest w metodzie pewne prawdopodobieństwo  $p_a \in <0, 1>$ . Przyjęcie prawdopodobieństwa  $p_a$  oznacza, że część rozwiązań jest eliminowana i w ich miejsce losowane są nowe możliwe rozwiązania. Takie zjawisko jest odzwierciedleniem odkrycia jaja przez ptaki gospodarzy i jego wyrzuceniem lub budowie gniazda w innej lokalizacji.

Jak można zauważyć w klasycznym algorytmie CS posługujemy się określeniami, kukułka, gniazdo oraz jajo. **Gniazdo** oznacza lokalizację w której może znaleźć się potencjalne rozwiązanie – czyli **jajo**. A kolejne jaja są rozmieszczane przez kukułki w gniazdach. Pojedyncza iteracja polega na rozmieszczeniu

przez określoną liczbę kukułek jaj w dostępnych gniazdach. Autorzy prowadził badania nad opracowaniem własnego algorytmu kukułki według powyżej opisanych zasad.

W metodzie poszukiwania kukułczego nowe położenie gniazda w  $i$ -tej iteracji wyznaczane jest w następujący sposób:

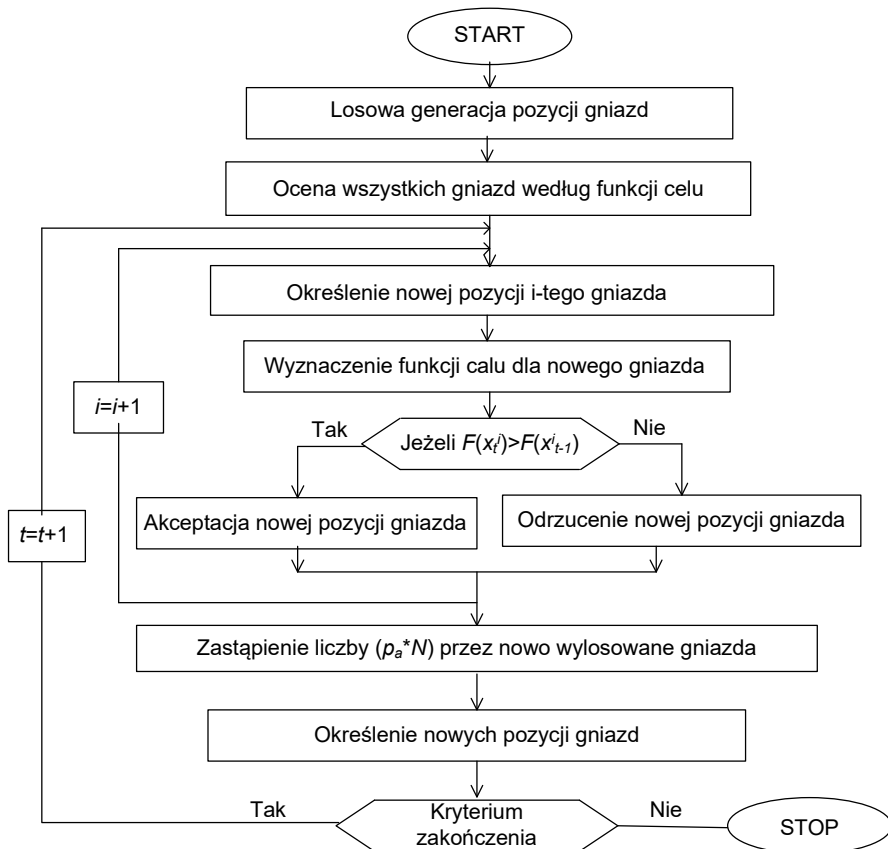
$$x_t^i = x_{t-1}^i + \alpha \kappa(\lambda) (x_b - x_{t-1}^i) \quad (1)$$

przy czym:  $t$  – numer iteracji algorytmu,  $\alpha > 0$  współczynnik dostosowujący algorytm do rozwiązywanego problemu optymalizacji,  $x_b$  – położenie najlepiej przystosowanego gniazda,  $\kappa(\lambda)$  – losowy współczynnik.

Losowy współczynnik  $\kappa(\lambda)$  wyznaczany jest na podstawie rozkładu Levy'ego [17, 18].

$$\kappa(\lambda) = t^\lambda, \quad (1 < \lambda \leq 3) \quad (2)$$

Schemat blokowy algorytmu CS przedstawiono na rysunku 1.



Rys. 1. Schemat blokowy algorytmu poszukiwania kukułczego

Kolejne kukułki rozmieszczają w gniazdach jaja z uwzględnieniem położenia najlepszej kukułki  $x_b$ . Po wyznaczeniu nowych pozycji gniazd każde jajo jest oceniane według wartości funkcji celu, jeżeli wartość funkcji celu jest lepsza od wartości funkcji celu w poprzedniej iteracji akceptowane są nowe współrzędne jaja, zaś w przeciwnym przypadku rozwiązanie jest odrzucane.

W celu uniknięcia utknięcia algorytmu w minimum lokalnym, w każdej iteracji zadania dodatkowo losowane są w sposób losowy nowe położenia gniazd. Liczba losowanych gniazd wynosi  $(N * p_a)$ , przy czym  $N$  jest zadaną liczbą gniazd. Po wylosowaniu nowych lokalizacji gniazd są one przesuwane w stronę najlepiej przystosowanego gniazda przy wykorzystaniu zależności:

$$x_t^i = x_r + \alpha \kappa(\lambda)(x_b - x_r) \quad (3)$$

gdzie:  $x_r$  – losowo określone dodatkowe położenie gniazda.

Po ocenie wszystkich losowo wybranych gniazd, z algorytmu eliminowana jest ilość  $(N * p_a)$  o najgorszej wartości funkcji celu.

#### 4. TEST POPRAWNOŚCI ALGORYTMU

Procedura optymalizacji została opracowana w języku Python 3.8. W celu weryfikacji poprawności działania procedury rozwiązano zadanie poszukiwania minimum globalnego analitycznej funkcji Rosenbrocka:

$$f(x, y) = (1 - x)^2 + 100(y - x^2)^2 \quad (4)$$

przy czym:  $-2 \leq x \leq 2$ ,  $-1,5 \leq y \leq 3$ .

Minimum globalne funkcji testowej znajduje się w punkcie (1,1) i wynosi  $f(1, 1) = 0$ .

Wykonano obliczenia optymalizacyjne przy wykorzystaniu algorytmu poszukiwania kukułki opracowanego przez autorów pracy. Przyjęto następujące parametry procedury optymalizacyjnej: (a) liczba kukułek równa 100, (b) liczba gniazd równa 125, (c) prawdopodobieństwo wykrycia jaja przez gospodarza gniazda równe 0,25 oraz (d) zadana liczba iteracji wynosząca 50. Przeprowadzono dziesięć niezależnych procesów optymalizacji dla różnych położen początkowych gniazd. W Tabeli 1 przedstawiono przebieg procesu optymalizacji dla najlepszego procesu poszukiwania minimum globalnego. W kolejnych wierszach przedstawiono współrzędne najlepszego gniazda, zaś w kolejnych kolumnach przedstawiono wartości: numer iteracji  $t$ , wartość współrzędnej  $x$  oraz  $y$  dla najlepszego gniazda, wartość funkcji Rosenbrocka oraz liczbę wywołań funkcji celu po  $N_f$  wykonanej przez procedurę optymalizacyjną.

Dla porównania jakości wyniku uzyskanego przy wykorzystaniu algorytmu poszukiwania kukułczego wykonano obliczenia dla metody wzorowanego na echolokacyjnym zachowaniu nietoperzy [19]. Podczas obliczeń symulacyjnych przyjęto następujące parametry procedury optymalizacyjnej: (a) ilość osobników

równą 100, (b) częstotliwości minimalną  $f_{\min} = 0$  oraz maksymalną  $f_{\max} = 1,2$ , początkową średnią wartość współczynnika emisyjności  $r_0 = 0$  całego stada nietoperzy, (c) początkową wartość współczynnika głośności całego stada  $A_0 = 1$ , (d) wartości współczynników  $\alpha = 0,75$  i  $\gamma = 0,5$  [20] oraz (e) maksymalną ilość kroków czasowych wynoszącą 50. Podobnie jak w przypadku pierwszej analizowanej procedury wykonano dziesięć uruchomień procedury obliczeniowej dla różnych położeń startowych nietoperzy.

Tabela 1. Przebieg procesu poszukiwania minimum dla funkcji Rosenbrocka

$t$	$x$	$y$	$f(x, y)$	$N_f$
1	-0,87082	0,56298	7,70281	100
2	-0,21423	0,12821	2,15317	225
4	0,32055	4,54405	7,90125	475
8	1,11398	1,25934	0,04678	975
10	0,82292	0,96926	0,03964	1225
15	0,97518	0,96926	0,00867	1850
20	0,98518	0,96928	0,00072	2475
30	0,98518	0,98926	0,00036	3725
50	0,98897	0,99601	0,00031	6225

Przebieg poszukiwania minimum funkcji Rosenbrocka przez algorytm „nietoperzy” przedstawiono w Tabeli 2.

Tabela 2. Przebieg procesu poszukiwania minimum dla funkcji Rosenbrocka przy wykorzystaniu algorytmu wzorowanego na echolokacyjnym zachowaniu nietoperzy

$t$	$x$	$y$	$f(x, y)$	$N_f$
1	0,83487	0,67874	0,06064	100
5	0,98768	0,98042	0,00254	600
10	0,98768	0,98042	0,00254	1000
20	0,98768	0,98042	0,00254	2000
30	0,98768	0,98042	0,00254	3000
40	0,98768	0,98042	0,00254	4000
50	0,98361	0,96833	0,00033	5000

Na podstawie uzyskanych wyników dla najlepszych procesów optymalizacji spośród dziesięciu uruchomień procedury optymalizacyjnej możemy stwierdzić, że uzyskane rezultaty dla dwóch badanych metod są zbliżone. W przypadku metody poszukiwania kukułczego uzyskaliśmy wartości współrzędnych  $x$  oraz  $y$  bardziej dokładne jak w przypadku metody wzorowanej na echolokacyjnym zachowaniu nietoperzy. Metoda kukułki wymaga większej ilości wywołań funkcji celu w stosunku do metody „nietoperzy”.

Następnie, porównano wyniki analizy statystycznej dla każdej z analizowanych metod. Wyznaczono najlepszą, najgorszą oraz średnią wartość funkcji celu. Dodatkowo wyznaczono wartość odchylenia standardowego. Wyniki dla dziesięciu niezależnych uruchomień procedury optymalizacyjnej zestawiono w Tabeli 3.

Tabela 3. Porównanie wyników analizy statystycznej dla metody poszukiwania kukułczego oraz metody nietoperzy

<i>Metoda</i>	<i>Najlepsza</i>	<i>Najgorsza</i>	<i>Średnia</i>	<i>Odchylenie standardowe</i>
CS	0,00031	0,0368	0,00695	0,0106
BA	0,00033	0,0803	0,02251	0,0294

Na podstawie uzyskanych wyników możemy stwierdzić, że wszystkie analizowane wielkości statystyczne otrzymaliśmy lepsze dla metody poszukiwania kukułczego. Dodatkowo otrzymywane wyniki przy wykorzystaniu metody CS są bardziej powtarzalne niż w przypadku algorytmu wzorowanego na echolokacyjnym zachowaniu nietoperzy.

## 5. OPROGRAMOWANIE DO WYZNACZANIA KSZTAŁTÓW IMPULSÓW ZASILAJĄCYCH SILNIK BLDC

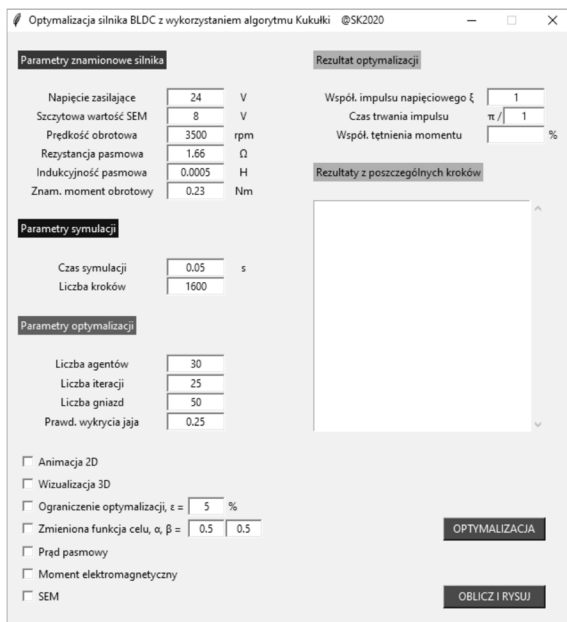
Rysunek 2 ilustruje formularz aplikacji komputerowej do wyznaczania kształtu impulsów zasilających silnik BLDC.

Oprogramowanie zostało opracowane w języku Python 3.8 w środowisku PyCharm Community Edition 2020.1.2x64. Do jego opracowania zastosowane zostały następujące biblioteki: (a) numpy, (b) tkinter, (c) matplotlib, (d) SwarmPackagePy, (e) pandas, (f) pyinstaller. Część bibliotek została zmodyfikowana przez autorów pracy w celu lepszej integralności z opracowanym oprogramowaniem. Program przeznaczony jest do zastosowania w systemie operacyjnym Windows 10 64-bit. Formularz aplikacji (rys. 2) składa się z czterech zależnych od siebie części: a) parametry znamionowe analizowanego silnika, b) parametry symulacji modelu silnika, c) parametry procedury optymalizacji (metoda CS), d) moduł wyświetlający informację o wynikach obliczeń symulacyjnych.

Przed rozpoczęciem obliczeń użytkownik musi wprowadzić **dane znamionowe silnika**, którego będzie dotyczył proces optymalizacji. Podczas obliczeń symulacyjnych wykorzystano dane znamionowe seryjnie produkowanego silnika BLDC [23].

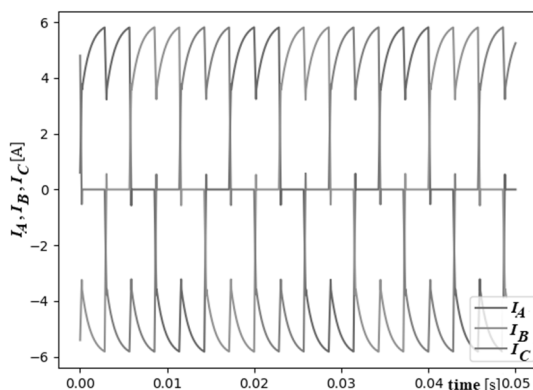
Następnie użytkownik powinien zadeklarować długość czasu symulacji oraz liczbę kroków czasowych w części **parametry symulacji**. Należy pamiętać, że dobór czasu, jak i liczby kroków symulacji ma istotny wpływ na dokładność otrzymanych wyników.

W następnym etapie, użytkownik musi zadeklarować **parametry procedury optymalizacji**: liczbę kukulek, liczbę gniazd, prawdopodobieństwo porzucenia gniazda oraz liczbę iteracji.



Rys. 2. Interfejs użytkownika opracowanego oprogramowania

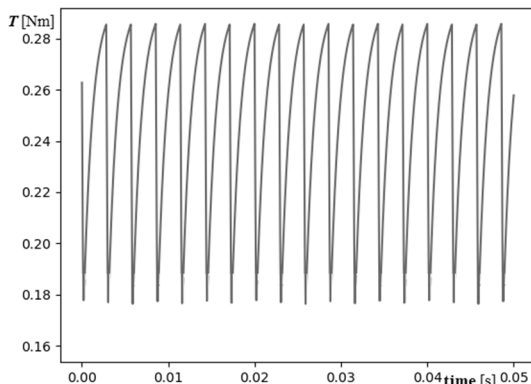
Po wprowadzeniu danych silnika możliwe jest wyznaczenie przebiegów prądu pasmowego oraz momentu elektromagnetycznego na wale silnika przy obciążeniu momentem znamionowym. Wyniki obliczeń symulacyjnych przedstawiono na rysunkach 3 oraz 4.



Rys. 3. Przebiegi prądów fazowych dla wartości momentu znamionowego



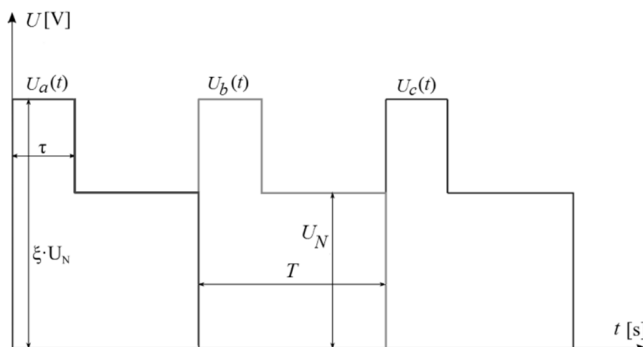
W przypadku zasilania napięciem znamionowym przy wykorzystaniu komutatora elektronicznego złożonego z 6-ciu tranzystorów współczynnik pulsacji wynosi 46,32%.



Rys. 4. Przebiegi momentu elektromagnetycznego przy znamionowym obciążeniu

## 6. WYNIKI OBLICZEŃ SYMULACYJNYCH

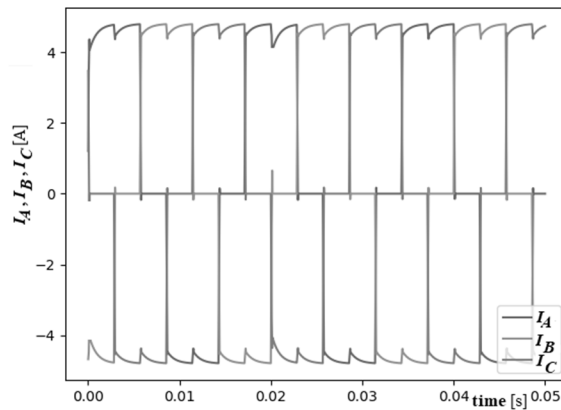
Zadaniem procedury optymalizacyjnej jest taki dobór dwóch parametrów charakteryzujących napięcia fazowe zasilające pasma stojana silnika BLDC, aby zapewnić zadaną wartość współczynnika pulsacji. Kształt przebiegu napięcia zasilającego został opisany za pomocą dwóch parametrów: (a)  $\xi$  – współczynnik określający ile razy większym napięciem należy zasilić uzwojenie stojana, aby uzyskać zadaną wartość współczynnika pulsacji, (b)  $\tau$  – czas trwania impulsu zwiększonej wartości napięcia na paśmie stojana. Zmodyfikowany kształt impulsu przedstawiono na rysunku 5.



Rys. 5. Przebiegi napięcia zasilającego pasma stojana

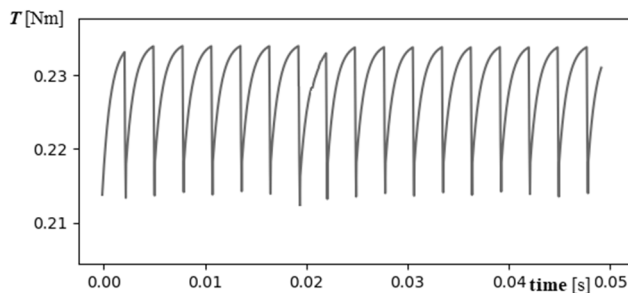
Wykonano obliczenia symulacyjne dla następujących parametrów algorytmu optymalizacji: (a) liczba kukułek 30, (b) liczba gniazd 40, (c) prawdopodobieństwo  $p_a = 0,3$  oraz (d) liczbę iteracji równą  $t_{\max} = 30$ . Obliczenia zostały wykonane dla momentu obciążenia równego momentowi znamionowemu  $T_N = 0,23$  Nm oraz zadanej wartości współczynnika pulsacji -  $\varepsilon$  [21, 22] wynoszącej 22%.

W wyniku obliczeń algorytmu optymalizacji otrzymano następujące wartości parametrów określających kształt impulsów zasilających:  $\xi = 1,3432$  oraz  $\tau = 0,1322$ . Wartości parametrów określających właściwości silnika wyniosły odpowiednio:  $\varepsilon = 22,0011\%$  oraz średnią wartość momentu  $T_{sr} = 0,2269$  Nm. Przebiegi prądów pasmowych w analizowanym silniku dla zadanej wartości współczynnika  $\varepsilon$  przedstawiono na rysunku 6.



Rys. 6. Przebiegi prądów w pasmach stojana dla zadanej dopuszczalnej wartości współczynnika pulsacji równej 22%

Rysunek 7 ilustruje przebieg momentu elektromagnetycznego na wale silnika dla zadanej wartości współczynnika pulsacji  $\varepsilon = 22\%$ .



Rys. 7. Przebieg momentu elektromagnetycznego na wale silnika

## 7. PODSUMOWANIE

W artykule przedstawiono teoretyczne podstawy algorytmu poszukiwania kukułczego. Przedstawiono procedurę optymalizacji opartą na zachowaniach rozrodczych kukułki zwyczajnej. Poprawność opracowanej procedury optymalizacyjnej była weryfikowana przez wyznaczenie minimum lokalnego testowej funkcji analitycznej. Porównano wyniki obliczeń dla opracowanej procedury optymalizacyjnej i procedury „nietoperzy”. Najlepszy wyniki dla obu porównywanych metod był podobny, zaś wartości odchylenia standardowego oraz średnia wartość funkcji celu dla rozwiązywania optymalnego były zdecydowanie lepsze dla metody poszukiwania kukułczego. Dlatego należy stwierdzić, że wyniki uzyskiwane przez metodę poszukiwania kukułczego są powtarzalne.

Wykorzystując opracowaną aplikację komputerową możliwe jest wyznaczenie kształtu impulsów zasilających pasma stojana silnika BLDC pozwalające uzyskać wymaganą wartość współczynnika pulsacji.

## LITERATURA

- [1] Wojciechowski R., Analysis and optimization of an axial flux permanent magnet coreless motor based using superposition principle and genetic algorithm, Archives of Electrical Engineering, vol. 65, no. 3, pp. 601 – 611, 2016.
- [2] Mutluer M., Sahman A., Cunkas M., “Heuristic optimization based on penalty approach for surface permanent magnet synchronous machines”, Arabian Journal for Science and Engineering, vol. 45, pp. 6751-6767, 2020.
- [3] Knypiński Ł., Optimal design of the rotor geometry of line-start permanent magnet synchronous motor using the bat algorithm, Open Physics, pp. 965 – 970, 2017.
- [3] Belahcen, A.; Floran, M.; El-Hadi-Zaim, Z. Kolondzovski, Z. Combined FE and particle swarm algorithm for optimization high speed PM synchronous machine. COMPEL 2015, 34, 2, 475 – 484.
- [4] Knypiński Ł., Pawełoszek K, Le Manech Y., Optimization of low-power line-start PM motor using gray wolf metaheuristic algorithm, Energies, vol. 13, no. 5, 2020.
- [5] Demenko A., Stachowiak D., Finite Element and Experimental Analysis of an Axisymmetric Electromechanical Converter with a Magnetostrictive Rod, Energies, vol. 13, no. 5, pp. 1 – 12, 2020.
- [6] Barański M., FE analysis of coupled electromagnetic-thermal phenomena in the squirrel cage motor working at high ambient temperature, COMPEL, vol. 38, no. 4, pp. 1120 – 1132, 2019.
- [7] Arnoux P. H., Caillard P., Gillon F., Modeling finite-element constraint to run an electrical machine design optimization using machine learning, IEEE Transactions on Magnetics, vol. 51, no. 3, 2015.
- [8] Dey B., García Márquez F. P., Basak S., Smart energy management of residential microgrid system by a novel hybrid MGWOSCACSA algorithm, Energies, vol. 13, no. 3500, pp. 1 – 23, 2020.

- [9] Devarapalli R.; Kumar Sinh N.; Bathinavenkateswara R.; Knypiński, Ł.; Jaya Naga N., Garcia Marquez F. P., Allocation of real power generation based on computing over all generation cost: an approach of Salp Swarm Algorithm. *Archives of Electrical Engineering* 2021, 78, 2, 337 – 349, 2021.
- [10] Knypiński, Ł. Constrained optimization of line-start PM motor based on the gray wolf optimizer. *Eksploracja i Niezawodność – Maintenance and Reliability*, pp. vol. 23, no. 1, pp. 1 – 10, 2021.
- [11] Devarapalli R., Bhattacharyya B., Sinha N. K., Dey B., “Amended GWO approach based multi-machine power system stability enhancement”, *ISA Transactions*, vol. 109, pp. 152–174, 2021.
- [12] Shi Y., Brain Storm Optimization Algorithm, *Advances in Swarm Intelligence*, pp. 303 – 309, pp. 2011.
- [13] Larson J., Menickelly M. and Wild S., Derivative-free optimization methods, *Acta Numerica*, vol. 28, pp. 287-404, 2019
- [14] Collar N. J., *Handbook of the Birds of the World*, vol. 4, pp. 280 -477, 1997.
- [15] Yang X. S., Deb S., Cuckoo search via Lévy flights, *World Congress on Nature and Biologically Inspired Computing*, pp. 210–214, 2009.
- [16] Morelli, F.; Møller, A. P.; Nelson, E.; Benedetti, Y.; Liang, W.; Šimová, P.; Moretti, M.; Tryjanowski, P. The common cuckoo is an effective indicator of high bird species richness in Asia and Europe. *Scientific Reports*, no. 7, 2017.
- [17]. Mesa A., Castromayor K., Garillos-Manliguez C., Calag, V., Cuckoo search via Levy flights applied to uncapacitated facility location problem. *Journal of Industrial Engineering International*, vol. 14, pp. 585–592, 2018.
- [18] Wang L., Guo H., Marignetti F., Shaver C., Bianchi N., Cuckoo Search Algorithm for Multi-Objective Optimization of Transient Starting Characteristics of a Self-Starting HVPMSM. *IEEE Transactions on Energy Conversion*, 2021.
- [19] Knypiński Ł., Zastosowanie metody wzorowanej na eholokacyjnym zachowaniu nietoperzy w optymalnym projektowaniu przetworników elektromagnetycznych, *Poznań University Academic Journals, Electrical Engineering*, No. 91, s. 365 – 374, 2017.
- [20] Ł. Knypiński, Optimal design of the rotor geometry of line-start permanent magnet synchronous motor using the bat algorithm, *Open Physics*, vol. 15, no.1, pp. 965 – 970, 2017.
- [21] Stachowiak D., Edge element analysis of brushless motors with inhomogeneously magnetized permanent magnets, *COMPEL*, vol. 23, no. 4, pp. 1119 – 1128, 2004.
- [22] Knypiński, L. Nowak, Optimization of the permanent magnet brushless DC motor employing finite element method, *COMPEL*, vol. 32, no. 4, pp 1189 – 1202, 2013.
- [23] <https://www.omc-stepperonline.com/24v-3500rpm-023nm-84w-50a-%D1%84-57x49mm-brushless-dc-motor-57blr50-24-01.html> [dostęp z dnia 19.01.2021]

---

**COMPUTER SOFTWARE FOR DETERMINING THE SHAPE  
OF THE SUPPLY VOLTAGE WAVEFORM THE BLDC MOTOR  
USING CUCKOO SEARCH ALGORITHM**

In the paper an algorithm and computer software for determining the shape of the supply voltage waveform for a brushless DC motor (BLDC) was developed. The computer application was made in the Python 3.8. The software consists of two independent modules: (a) the optimization procedure and (b) the mathematical model of the BLDC motor. The mathematical model of device was based on lumped parameters. The optimization procedure contains the cuckoo search optimization algorithm. The performance of cuckoo search algorithm was compared with bat algorithm using Rosenbrock function. Finally the optimization procedure was applied to determining the supply waveform ensuring the minimization of torque ripple in BLDC motor.

*(Received: 17.08.2021, revised: 31.10.2021)*