

Adaptacyjny regulator neuronowy typu RBF zastosowany w sterowaniu napędem elektrycznym z silnikami PMSM

Streszczenie. Niniejszy artykuł przedstawia strukturę sterowania prędkością układu napędowego z silnikiem PMSM. Pierwsza część opisywanego projektu dotyczyła konstrukcji rzeczywistego stanowiska laboratoryjnego. Istotnym założeniem, w tym etapie prac, była redukcja kosztów poprzez implementację algorytmu sterowania w tanim procesorze ARM. Kolejnym zadaniem była analiza działania adaptacyjnego regulatora prędkości, opartego o model sieci radialnej (*Radial Basis Function*), której współczynniki wagowe podlegały adaptacji w trybie *on-line*. Podczas projektowania regulatora zastosowano metaheurystyczny algorytm BAT.

Abstract. This article presents speed control structure of electrical drive with PMSM motor. First part of project is related to hardware construction of real drive. Important assumption was cost reduction of experimental platform. For this purpose the control algorithm was implemented in low-cost programmable device (ARM processor). Next stage of work was focused on design and analysis of adaptive speed controller, this part of control structure was based on Radial Basis Function neural network. Additionally, metaheuristic BAT algorithm was applied for optimization of selected elements of neural controller. (*Adaptive neural controller based on RBF model applied for electrical drive with PMSM motor*).

Słowa kluczowe: regulator neuronowy, sieć radialna, regulacja prędkości, napęd elektryczny.

Keywords: neural controller, radial basis function network, speed control, electric drive.

Wprowadzenie

Najistotniejsze wymagania (determinujące właściwości użytkowe) dotyczące nowoczesnych napędów elektrycznych stosowanych w aplikacjach przemysłowych, zakładają dynamiczne oraz precyzyjne sterowanie prędkością lub położeniem wału silnika. W ostatnich latach można zaobserwować trend badań, wynikający z przyczyn praktycznych, według którego oczekuje się również poprawnego działania układu napędowego w przypadku niepewności oraz zmian parametrów obiektu [1], [2]. Jednym ze sposobów uzyskania wymienionych powyżej cech napędu elektrycznego jest wykorzystanie właściwości silnika PMSM (*Permanent Magnet Synchronous Motor*) oraz nowoczesnych adaptacyjnych algorytmów sterowania, zastosowanych w pętli regulacji prędkości.

Rozwój teorii oraz aplikacje regulatorów neuronowych są obecnie bardzo szeroko opisywane w publikacjach naukowych. Wśród charakterystycznych zastosowań radialnych sieci neuronowych należy wymienić zastosowanie modelu tego typu w sterowaniu odwróconym wahadłem zawierającym dodatkowy stopień swobody. Mechanizm zawiera jeden silnik prądu stałego, w precyzyjnym sterowaniu położeniem istotnym zakłóceniem są nieliniowości związane z momentem tarcia. Sieci neuronowe radialne mogą stanowić skuteczny kompensator, który nie wymaga bezpośredniej identyfikacji parametrów funkcji opisujących tarcie [3]. Efektywnym rozwiązaniem jest również bezpośrednia aplikacja perceptronowej sieci neuronowej jako regulatora prędkości w napędzie z silnikiem PMSM. Wysoka dynamika napędu z testowanym modelem adaptacyjnym uzasadnia zastosowanie takich regulatorów. Warto zwrócić uwagę, że w zagadnieniach dotyczących zastosowań sieci neuronowych, istotnym elementem są rozważania związane ze złożonością obliczeniową algorytmów przeliczających wagi regulatora. Prowadzone badania mają na celu redukcję wymaganej mocy obliczeniowej, zatem testowane są uproszczone metody wyznaczania poprawek współczynników modelu. Przykładem jest zastosowanie algorytmu RPROP (*the Resilient backPROPagation*), który zakłada wykorzystanie jedynie znaku gradientu funkcji celu, zamiast pełnej wartości [4]. Poza bezpośrednim oparciem struktury sterowania na modelu neuronowym, wyróżniającym się podejściem (często opisywanym w artykułach), wynikającym z właściwości sieci z radialnymi

funkcjami aktywacji, stosowane jest łączenie metod sterowania. Przykładami są układy sterowania wykorzystujące wprowadzanie sieci RBF do: sterowania predykcyjnego [5], ślizgowego [6] oraz klasycznego (regulatory PI/PID) [7].

W niniejszym artykule analizowany jest napęd wykorzystujący adaptacyjny neuronowy regulator prędkości oparty o model sieci radialnej. Podstawowa idea przetwarzania danych takiej struktury, zakłada dzielenie przestrzeni danych w celu separacji elementów o podobnych właściwościach. W związku z tym sieci RBF, szczególnie trenowane *off-line*, są najczęściej łączone w inżynierskich aplikacjach z klasyfikacją danych [8]-[10]. Jest to związane z istotną cechą sieci radialnych – generalizacją danych. Bardzo dobre rezultaty są uzyskiwane nawet w przypadku wykorzystania w trakcie treningu ograniczonego zbioru danych, zawierającego przypadkowe próbki lub znaczące błędy np. pomiarowe. Wspomniana cecha predysponuje modele RBF do rzeczywistych aplikacji. Ponadto, w ostatnich latach obserwowany jest zdecydowany wzrost aplikacji sieci RBF dla systemów dynamicznych. Wymienione implementacje są wynikiem, możliwości efektywnego oraz relatywnie prostego treningu (szczególnie przy założeniu adaptacji w części liniowej), wprowadzanego w trybie *on-line*. W publikacjach można również znaleźć badania prezentujące większą odporność na zakłócenia wejściowe (szumy wprowadzane do danych), w porównaniu do perceptronowych sieci neuronowych [11].

W ostatnich latach wzrasta liczba zastosowań algorytmów inspirowanych obserwacjami biologicznymi, stosowanymi w syntezie układów regulacji napędami elektrycznymi [12]-[14]. Jest to spowodowane obiecującymi rezultatami otrzymywanymi za pomocą tych metod, uniwersalną metodyką aplikacji algorytmu dla odrębnych zadań, możliwością wykonywania optymalizacji uwzględniającej jednocześnie zmiany wielu zmiennych stanu oraz uproszczonymi obliczeniami (brak konieczności wyznaczania gradientu funkcji celu) [15].

W prezentowanym artykule opisano aplikację regulatora neuronowego, wykorzystującego model RBF, w sterowaniu silnikiem PMSM. Wybrane wartości parametrów w pętli sterowania prędkością optymalizowano za pomocą algorytmu BAT. Kolejne rozdziały artykułu zawierają opis struktury sterowania oraz poszczególnych metod

obliczeniowych. W celu weryfikacji eksperymentalnej skonstruowano stanowisko laboratoryjne zawierające zestaw STM32F4Discovery. Wybrane etapy wykonywania napędu oraz wyniki badań eksperymentalnych zostały zaprezentowane w ostatnich częściach publikacji.

Struktura sterowania napędem z silnikami PMSM wykorzystująca adaptacyjny regulator prędkości

W opisywanej aplikacji sterowanie silnikiem zrealizowano z wykorzystaniem falownika napięcia w strukturze DFOC. Układ sterowania bezpośrednią metodą połowo zorientowaną pozwala na uzyskanie wysokiej dynamiki sterowania zmiennymi stanami, szybkie odpowiedzi części odpowiedzialnej za kontrolę momentu elektromagnetycznego oraz redukcji tętnień prądów. Model matematyczny silnika PMSM oraz struktura sterowania zostały szczegółowo opisane w publikacjach [16], [17].

Analizowany regulator prędkości wykorzystuje dwa sygnały związane z błędem regulacji prędkości. Pierwszy z nich - e jest definiowany jako różnica pomiędzy wartością zadaną oraz aktualną prędkością silnika. Drugi - e_m jest obliczany z wykorzystaniem modelu odniesienia o transmitancji:

$$(1) \quad G_{refm}(s) = \frac{\omega_r^2}{s^2 + 2\xi_r \omega_r s + \omega_r^2},$$

w oznaczeniach przyjęto: ξ_r – współczynnik tłumienia, ω_r – zadana pulsacja rezonansowa.

Wprowadzenie transmitancji modelu odniesienia jest związane z koniecznością wygaszania adaptacji po osiągnięciu odpowiedniej dokładności sterowania prędkością.

Wewnętrzne neurony modelu wykorzystują w obliczeniach Gaussowskie funkcje aktywacji, na etapie h -tego elementu sieci dane są przetwarzane na podstawie zależności [18]:

$$(2) \quad g_h = \varphi_h(X(i)) = e^{(-\beta v_h(i))},$$

gdzie: β jest parametrem skalującym, obliczanym z wykorzystaniem stałej σ :

$$(3) \quad \beta = \frac{1}{2\sigma^2}.$$

We wzorze (2), za pomocą v_h oznaczono odległość pomiędzy elementem x_k wektora wejściowego X oraz centrum c_h neuronu h -tego:

$$(4) \quad v_h(i) = \|X(i) - c_h\|^2 = \sum_{k=1}^n (x_k(i) - c_{hk})^2.$$

Wartość wyjściowa, dla kolejnych neuronów w ostatnim etapie procesu przetwarzania sieci radialnej, jest opisana wzorem:

$$(5) \quad y_o(i) = \sum_{g=1}^G w_{og}(i) e^{(-\beta v_h(i))} = \sum_{g=1}^G w_{og}(i) e^{\left(\frac{-1}{2\sigma^2} \|X(i) - c_h\|^2\right)}, \quad o = 1, 2, \dots, O,$$

w powyższym równaniu w_{og} oznacza wartość współczynnika wagowego, pomiędzy g -tym neuronem warstwy radialnej oraz o -tym wyjściem, G – liczba neuronów warstwy radialnej. W testowanym regulatorze współczynniki wagowe warstwy wyjściowej wyznaczano *on-line* w trakcie obliczeń całej struktury. W każdej iteracji i działania programu wagi aktualizowano zgodnie z zależnością:

$$(6) \quad \mathbf{w}(i+1) = \mathbf{w}(i) + \Delta \mathbf{w}(i), \quad k = 0, 1, 2, \dots,$$

gdzie: \mathbf{w} - macierz współczynników wagowych.

Korektę wag przeprowadzano zgodnie z algorytmem LMS (*Least Mean Square*) [19]. W poszczególnych krokach obliczeniowych wykonywano adaptację według wzoru:

$$(7) \quad \mathbf{w}(i+1) = \mathbf{w}(i) - 2\gamma \left(\sum_{g=1}^G x_g(i) w_{og}(i) - d(i) \right) x_g(i),$$

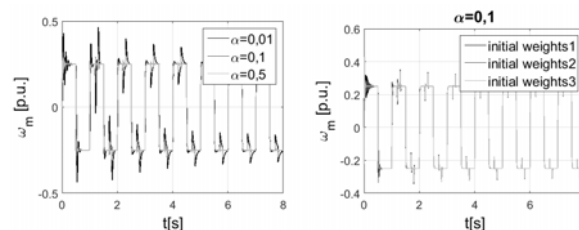
gdzie:

$$(8) \quad \alpha = 2\gamma.$$

W odniesieniu do ogólnej struktury sieci neuronowej radialnej, w opisywanej aplikacji zastosowano model zawierający dwa wejścia, do których wprowadzano następujące sygnały: aktualny uchyb prędkości oraz błąd z poprzedniej iteracji algorytmu sterowania:

$$(9) \quad X(i) = [x_1(i) \quad x_2(i)] = [e(i) \quad e(i-1)],$$

ponadto wykorzystano pięć neuronów radialnych ($G=5$), natomiast w warstwie wyjściowej zastosowano jeden neuron ($O=1$), generujący sygnał zadany momentu elektromagnetycznego w strukturze sterowania.



Rys.1. Przebiegi prędkości silnika PMSM dla różnych wartości współczynnika α oraz wartości początkowych wag

Analizowany regulator prędkości, nie wymaga w trakcie projektowania (bezpośrednio) informacji o parametrach obiektu oraz modelu matematycznym. Jednak, pojawiają się inne problemy projektowe, typowe dla struktur adaptacyjnych: problem związany z wyznaczaniem stałej algorytmu adaptacyjnego α oraz wartości początkowych wag (które najczęściej są losowane). Poprawny dobór wspomnianych parametrów wpływa na działanie regulatora neuronowego. Przykłady, przedstawiające wyniki badań symulacyjnych, zamieszczono na rysunku 1. W związku z czym, zasadne jest poszukiwanie rozwiązań w tym zakresie. W niniejszym artykule wykazano, że zastosowanie algorytmów metaheurystycznych, w opisanym zadaniu, umożliwiłoby uzyskanie bardzo dobrych rezultatów.

Algorytm optymalizacyjny BAT

Algorytm BAT należy do grupy metod optymalizacyjnych, których obliczenia inspirowane są zachowaniem naturalnych skupisk osobników. Głównym zadaniem grupy jest najczęściej zdobycie pożywienia, w przetwarzaniu danych odpowiada to minimalizacji funkcji celu w konkretnym zadaniu inżynierskim [12]-[15]. Zaletą algorytmu BAT jest szybka zbieżność algorytmu, która jest obserwowana już w początkowej fazie obliczeń. Wymieniona cecha jest związana ze specyficznymi założeniami, które prowadzą do zawężenia poszukiwań wokół ograniczonego regionu najlepszych rozwiązań. Istotną cechą analizowanej metody optymalizacji jest również występowanie przestrajalnych parametrów, opisujących oddziaływanie osobników na otoczenie (L_i oraz

r_i). W ten sposób, poza zwiększeniem efektywności, zredukowana zostaje konieczność definiowania przez projektanta kolejnych współczynników (trudnych do jednoznacznego opisanego wzorami). Odrębnymi cechami, które posiada znaczenie dla praktycznych implementacji są: brak złożonych obliczeń oraz prostota podstawowych założeń algorytmu BAT.

Schemat blokowy prezentujący ogólne założenia algorytmu BAT przedstawiony został na rysunku 2. Pierwszym etapem obliczeń jest deklaracja parametrów, definiujących właściwości populacji. Następnie algorytm rozpoczyna część iteracyjną, w której wyznaczane są nowe parametry osobników, każdy z nich jest opisywany poprzez częstotliwość emitowanego sygnału f_i [20]:

$$(10) \quad f_i = f_{\min} + (f_{\max} - f_{\min})\beta_{BAT},$$

położenie x_i :

$$(11) \quad x_i(k) = x_i(k-1) + v_i(k)$$

oraz prędkość v_i :

$$(12) \quad v_i(k) = v_i(k-1) + (x_i(k) - x_{i,best})f_i.$$

W kolejnej części, w zależności od współczynnika ilości impulsów r_i , generowane są kolejne rozwiązania w obszarze zawierającym najlepsze osobniki lub wartości są losowane. Ostatni etap danej iteracji uwzględnia: >sprawdzenie ograniczeń dla potencjalnie najlepszych osobników z aktualnego cyklu (założono poziom $<0,1>$ dla α oraz $<-10,10>$ dla wag), >modyfikację parametrów algorytmu, realizowaną zgodnie z poniższymi wzorami:

$$(13) \quad L_i(k+1) = \alpha_{BAT}L_i(k)$$

$$(14) \quad r_i(k+1) = r_i^0(1 - \exp(-\gamma_{BAT}k)),$$

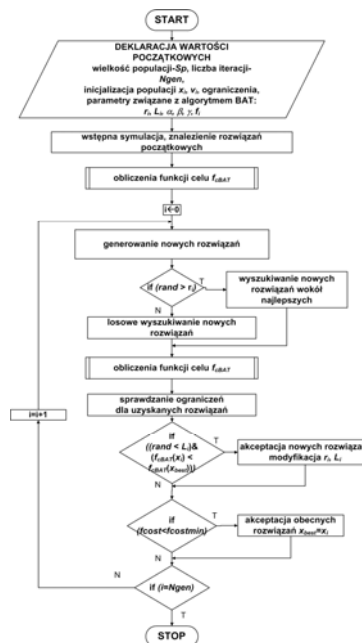
gdzie: r_i^0 – jest wartością początkową, α_{BAT} oraz γ_{BAT} stałe algorytmu, >ocenę uzyskanych rozwiązań.

W zadaniu opisywanym w niniejszym artykule, optymalizowano wartość stałego współczynnika części obliczającej korekty parametrów regulatora RBF oraz wartości początkowe wag sieci radialnej. Algorytm zaimplementowany został jako skrypt programu Matlab, który współpracował z modelem nakładki graficznej Simulink (zawierającym całą strukturę sterowania). Opracowany program wymagał zdefiniowania kilku wartości opisujących złożenia dla skryptu. Warto zwrócić uwagę, że są one często kompromisem pomiędzy dokładnością obliczeń a czasem pracy programu (generalnie, zwiększanie liczby iteracji oraz populacji może wpłynąć na dokładniejsze wyniki, jednak przetwarzanie danych trwa dłużej). Przykładowe parametry zapewniające dobre rezultaty oraz zbieżność uzyskiwanych wyników przy kolejnych powtórzeniach algorytmu są następujące: $S_p=20$, $N_{gen}=10$, $\alpha_{BAT}=0,5$, $\gamma_{BAT}=0,01$. W obliczeniach założono funkcję celu opisaną następującą zależnością:

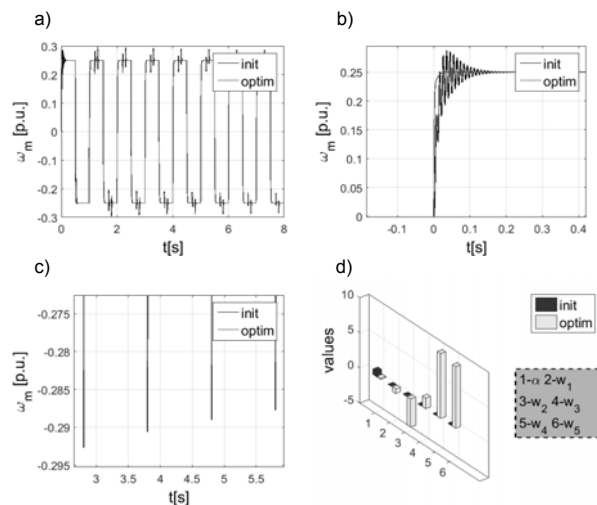
$$(15) \quad f_{cBAT} = \left(\sum_{i=1}^p \omega_{ref} - \omega_m \right) + a \left(\sum_{i=1}^k \omega_{ref} - \omega_m \right),$$

gdzie: ω_{ref} – prędkość zadana, ω_m – prędkość mierzona, t_s – krok obliczeniowy (element skalujący), a – stały współczynnik (przyjęto $a=10$), p – całkowita liczba próbek, k – liczba próbek.

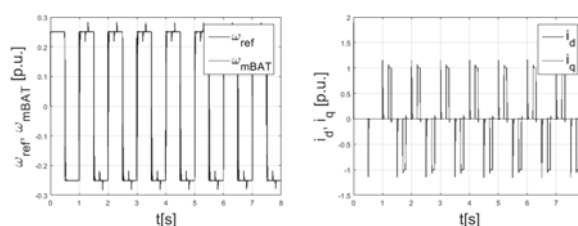
Niedokładność sterowania w algorytmie optymalizacyjnym przedstawia błąd pomiędzy wartością zadaną oraz aktualną prędkości. Charakterystycznym elementem funkcji (15) jest wprowadzony dodatkowy (drugi) człon sumowania, umożliwiający zwiększenie wpływu oddziaływania na obliczenia początkowej fazy rozruchu napędu (k pierwszych próbek zarejestrowanych po rozruchu silnika).



Rys.2. Schemat obliczeń algorytmu optymalizacyjnego BAT



Rys.3. Przebiegi prędkości silnika PMSM oraz parametrów regulatora neuronowego RBF w strukturze sterowania – rezultaty optymalizacji



Rys.4. Przebiegi prędkości oraz prądów (i_d , i_q) w strukturze sterowania z regulatorem RBF zoptymalizowanym za pomocą algorytmu BAT

Rezultat działania algorytmu BAT, przedstawia rysunek 3 (również zmiany wartości optymalizowanych parametrów - rysunek 3d). Jako stan odniesienia, początkowy (init) przyjęto sytuację, w której wagi początkowe sieci neuronowej oraz wartość współczynnika α (uwzględniając odpowiednie ograniczenia tego parametru) wylosowano. Test wykonany dla tego stanu prezentuje istotność poprawnego doboru wymienionych parametrów regulatora neuronowego. W przebiegach prędkości, w trakcie kilku początkowych sekund dostrajany jest regulator, obserwowane są oscylacje (przebiegi na rysunkach 3a oraz 3b). Zdecydowana poprawa jest widoczna w badaniach wykonanych dla wartości uzyskanych za pomocą algorytmu BAT (optim), tą część obliczeń zrealizowano w trybie *off-line*. Warto zwrócić uwagę, że algorytm adaptacyjny regulatora RBF wykonuje ciągłe obliczenia w trakcie pracy napędu (zgodnie z zależnością (7)). Efekt wprowadzanych zmian parametrów również wpływa na przebiegi prędkości (rysunek 3c). Błąd regulacji, liczony zgodnie ze wzorem (15), dla losowo wyznaczonych parametrów wynosił: 0,1153, natomiast po optymalizacji algorytmem BAT: 0,0929. Następnie zaprezentowano pracę zoptymalizowanego układu regulacji w przypadku znamionowych parametrów obiektu (rysunek 4). W początkowej fazie napęd działa bez obciążenia, silnik cyklicznie wykonuje nawroty do wartości 25% prędkości znamionowej. Od drugiej sekundy załączona została część programu sterującego, odpowiadająca za przełączanie obciążenia. W przebiegach prędkości można zaobserwować krótkotrwałe zakłócenia, jednak regulator skutecznie modyfikuje pracę układu, jest to również widoczne na wykresie prądu i_q .

Stanowisko laboratoryjne oraz wyniki badań eksperymentalnych

W ramach opisywanego projektu, wykonane zostało stanowisko laboratoryjne, umożliwiające testowanie struktur sterowania prędkością silnika PMSM. Ostateczna postać części wykonawczej została przedstawiona na rysunku 5. Budowa stanowiska badawczego opiera o wykorzystanie systemu profili aluminiowych, na których zamontowano dwie maszyny elektryczne. Silnikiem napędzającym jest silnik synchroniczny z magnesami trwałymi firmy ABB typ BSM0400CN00 (parametry przedstawiono w tabeli 1). W obiektach badawczych istnieje wiele rozwiązań dotyczących praktycznej realizacji maszyny roboczej. W prostych rozwiązaniach w roli obciążenia stosowane są hamulce mechaniczne wykorzystujące zjawisko tarcia. Jednak symulacja obciążenia, którego załączanie oraz wartość może być skutecznie modyfikowane (programowo), najczęściej odbywa się z wykorzystaniem drugiego silnika. W analizowanym przypadku jako obciążenie użyto silnika PMSM o identycznych parametrach jak maszyny napędzającej. Silniki połączone sztywnym wałem za pomocą dwóch sprzęgieł bezluzowych.

W pracy wykorzystano falownik, który posiada łatwy dostęp do wejść sterowników tranzystorów IGBT. Zastosowano trzy sterowniki IR21064, każdy dla jednej gałęzi falownika. Cechą tych modułów jest brak czasu martwego, w porównaniu z innymi modelami występującymi na rynku. Projektant decyduje o tym, ile wynosi czas pomiędzy załączeniem tranzystorów w jednej gałęzi. Falownik posiada moduł FS10R06VL4 B2 zawierający sześć tranzystorów IGBT. Następnie przy użyciu zasilacza laboratoryjnego ustawiano żądane napięcie stałe. Zastosowanie programowalnego układu zasilania pozwala na dokładną konfigurację oraz umożliwia ustawienie ograniczenia prądowego, co jest istotne przy testowaniu prototypów. Pomiędzy zasilaczem, a falownikiem

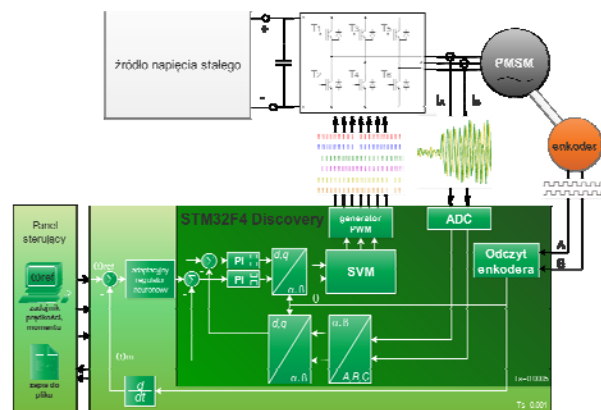
dodatkowo podłączono równolegle cztery kondensatory o pojemności 1500 μ F i napięciu maksymalnym 400V.



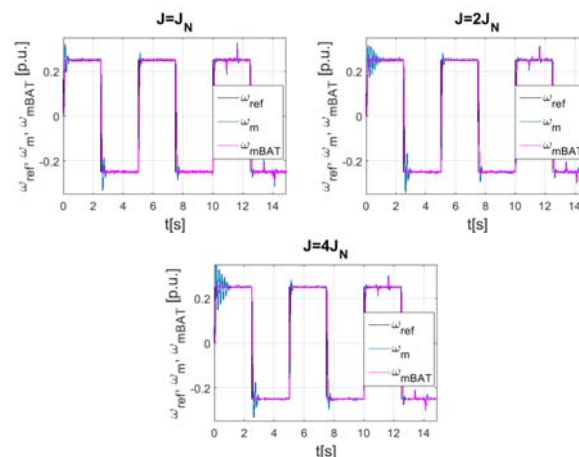
Rys.5. Stanowisko badawcze

Tabela 1. Parametry silników PMSM

BSM0400CN00	
Moc [W]	400
Znamionowy moment obrotowy [Nm]	1,27
Maksymalny moment obrotowy [Nm]	3,82
Prędkość znamionowa [obr/min]	3000
Prędkość maksymalna [obr/min]	5000
Moment [kg*cm ²]	0,24
Prąd znamionowy [A]	2,7
Prąd maksymalny [A]	8,1
Waga [kg]	1,8



Rys.6. Schemat przedstawiający przetwarzanie danych w strukturze sterowania rzeczywistym napędem



Rys.7. Przebiegi prędkości w rzeczywistym układzie napędowym – wpływ zmian momentu bezwładności

Napęd wyposażony został w dwa enkodery inkrementalne, zliczające z dokładnością: 10000imp./obr. (silnik) oraz 2000imp./obr. (maszyna robocza). Ze względu na wymagane wysokie dokładności pomiarowe oraz szybkość przetwarzania sygnału pomiar prądów zrealizowano przy użyciu przetwornika typu LEM HO-

P/SP33. Istotną zaletą przetwornika jest kompatybilność napięcia zasilania z napięciem pracy mikrokontrolera, dzięki temu nie jest konieczny dodatkowy układ dopasowujący sygnał wyjściowy przetwornika. Układ mocy połączono z mikrokontrolerem STM, poprzez optoizolację opartą o transoptory szybko 6N137.

Układem programowalnym, do którego zaimplementowano program sterujący pracą napędu (obsługa czujników pomiarowych, sterowanie częścią energoelektroniczną, algorytm obliczeniowy regulatora prędkości, itd.), jest zestaw STM32F4Discovery. Należy podkreślić, że proponowane rozwiązanie jest bardzo tanie, w porównaniu z często stosowanymi procesorami sygnałowymi. W zaimplementowanym kodzie wyróżnić można trzy główne części, wykonywane z różnymi częstotliwościami, które mogą być modyfikowane w zależności od potrzeb aplikacyjnych (rysunek 6). Podprogram nadzorujący pracę falownika, tj. załączenie układu, zadajnik prędkości oraz sygnały we/wy cyfrowych realizowany jest z częstotliwością asynchroniczną w głównej pętli programu. Dwa pozostałe człony programu wykonywane są z częstotliwością synchroniczną zgodną z dwoma *timerami* o częstotliwości $f_{speed}=1\text{kHz}$ (dla pętli regulacji prędkości) oraz $f_{current}=2\text{kHz}$ (dla pętli kształtowania momentu elektromagnetycznego).

Badania eksperymentalne zostały wykonane dla znamionowych oraz zakłóconych parametrów obiektu. W tym celu, na wale sprzęgającym montowano kolejne tarcze obciążające (rysunek 5), modyfikowano wartość momentu bezwładności J . Wyniki badań załączono na rysunku 7. W pierwszej części przebiegów, silnik nie jest obciążany, następnie wprowadzane są zmiany obciążenia. Trajektoria zadana prędkości jest odtwarzana bardzo precyzyjnie, nawet dla kolejnych testów, w których zmieniano wartość J . W testach przeprowadzonych na stanowisku laboratoryjnym, uzyskano również poprawę działania (zdecydowane skrócenie czasu dostrajania regulatora prędkości) poprzez wstępne dobranie parametrów regulatora za pomocą algorytmu BAT.

Podsumowanie

Artykuł przedstawia przykład zastosowania metaheurystycznego algorytmu BAT w optymalizacji procesu projektowania adaptacyjnego neuronowego regulatora, zastosowanego w sterowaniu prędkością napędu elektrycznego. Celem przeprowadzonych badań była próba wprowadzenia sformalizowanej metody wyznaczania, trudnych do jednoznacznego opisu matematycznego, parametrów regulatora prędkości opartego o model sieci radialnej. Zastosowanie algorytmu BAT w projektowaniu adaptacyjnego regulatora prędkości napędu elektrycznego z silnikiem PMSM, umożliwia redukcję oscylacji prędkości w początkowej fazie działania napędu (w trakcie dostrajania modelu neuronowego). Takie postępowanie, po wprowadzeniu odpowiednich ograniczeń, może również prowadzić do zapewnienia stabilności układu regulacji. Poza tym, wyeliminowanie zakłóceń, może zredukować prawdopodobieństwo występowania potencjalnych uszkodzeń części mechanicznej napędu. Aplikacja neuronowych regulatorów prędkości silnika PMSM umożliwia precyzyjne sterowanie napędem. Układ podąża dokładnie za trajekcją zadaną również w przypadku zmian momentu bezwładności silnika. Właściwości proponowanej struktury sterowania potwierdzone zostały w badaniach eksperymentalnych.

Autorzy: dr hab. inż. Marcin Kamiński, mgr inż. Karol Najdek, Politechnika Wroclawska, Katedra Maszyn, Napędów i Pomiarów Elektrycznych, ul. Smoluchowskiego 19, 50-372 Wrocław, E-mail: marcin.kaminski@pwr.edu.pl, karol.najdek@gmail.com.

LITERATURA

- [1] Kazmierkowski M.P., Franquelo L.G., Rodriguez J., Perez M.A., Leon J.I., High-Performance Motor Drives, *IEEE Ind. Electronics Magn.*, 5 (2011), n.2, 6-26
- [2] De Doncker R.W., Modern Electrical Drives: Design and Future Trends, *IEEE 5th International Power Electronics and Motion Control Conference*, 1 (2006), 1-8
- [3] Xia D., Wang L., Chai T., Neural-Network-Friction Compensation-Based Energy Swing-Up Control of Pendubot, *IEEE Trans. Ind. Electronics*, 61 (2014), n.3, 1411-1423
- [4] Pajchrowski T., Zawirski K., Nowopolski K., Neural Speed Controller Trained Online by Means of Modified RPROP Algorithm, *IEEE Trans. Ind. Informatics*, 11 (2015), n.2, 560-568
- [5] Csekő L.H., Kvasnica M., Lantos B., Explicit MPC-Based RBF Neural Network Controller Design With Discrete-Time Actual Kalman Filter for Semiactive Suspension, *IEEE Trans. Control. Syst. Tech.*, 23 (2015), n.5, 1736-1753
- [6] Hongbin W., Zelin Y., Zhen Z., RBF-based terminal sliding mode control for a class of underactuated mechanical system, *Chinese Control and Decision Conference (CCDC)*, (2016), 664-667
- [7] Yang Z.-K., Liu C.-Y., Song X.-L., Song Z.-Y., Wang Z.-S., Application of RBF neural network PID in wet flue gas desulfurization of thermal power plant, *International Conference on Machine Learning and Cybernetics (ICMLC)*, 1 (2016), 301-306
- [8] Sun Y., An Y., Research of on-line monitoring and fault diagnosis system for cold-rolling based on RBF neural network, *International Conference on Test and Measurement, ICTM '09*, 1 (2009), 165-168
- [9] Oyang Y.-J., Hwang S.-C., Ou Y.-Y., Chen C.-Y., Chen Z.-W., Data classification with radial basis function networks based on a novel kernel density estimation algorithm, *IEEE Trans. Neural Netw.*, 16 (2005), n.1, 225-236
- [10] Maglogiannis I., Sarimveis H., Kiranoudis C.T., Chatziioannou A.A., Oikonomou N., Aidinis V., Radial Basis Function Neural Networks Classification for the Recognition of Idiopathic Pulmonary Fibrosis in Microscopic Images, *IEEE Trans. Inf. Tech. in Biomedicine*, 12 (2008), n.1, 42-54
- [11] Yu H., Xie T., Paszczynski S., Wilamowski B.M., Advantages of Radial Basis Function Networks for Dynamic System Design, *IEEE Trans. Ind. Electronics*, 58 (2011), n.12, 5438-5450
- [12] Wang X., Ufnalski B., Grzesiak L.M., Adaptive speed control in the PMSM drive for a non-stationary repetitive process using particle swarms, *10th International Conference on Compatibility, Power Electronics and Power Engineering (CPE-POWERENG)*, (2016), 464-471
- [13] Premkumar K., Manikandan B.V., Speed control of Brushless DC motor using bat algorithm optimized Adaptive Neuro-Fuzzy Inference System, *Applied Soft Computing*, 32 (2015), 403-419
- [14] Tarczewski T., Grzesiak L.M., Application of artificial bee colony algorithm to auto-tuning of linear-quadratic regulator for PMSM position control, *Przegląd Elektrotechniczny*, 92 (2016), n.6, 57-62
- [15] Mirjalili S., Mirjalilib S.M., Lewis A., Grey Wolf Optimizer, *Advances in Engineering Software*, 69 (2014), 46-61
- [16] Zawirski K., Deskur J., Kaczmarek T., *Automatyka napędu elektrycznego*, Wydawnictwo Politechniki Poznańskiej, 2012.
- [17] Pajchrowski T., Adaptacyjne sterowanie silnikiem synchronicznym o magnesach trwałych, *Poznan University of Technology Academic Journals. Electrical Engineering*, (2013), nr.75, 127-133
- [18] Osowski S., *Sieci neuronowe do przetwarzania informacji*, Oficyna Wydawnicza Politechniki Warszawskiej, 2006
- [19] Haykin S.O., *Adaptive Filter Theory*, Pearson (5th edition, June 2013), 2013
- [20] Yang X.-S., A New Metaheuristic Bat-Inspired Algorithm, in: *Nature Inspired Cooperative Strategies for Optimization (NISCO 2010) (Eds. J. R. Gonzalez et al.)*, *Studies in Computational Intelligence*, Springer Berlin, 284, Springer, (2010), 65-74