

Algorytm GWO zastosowany w optymalizacji adaptacyjnego regulatora neuronowo-rozmytego układu dwumasowego

Streszczenie. Publikacja przedstawia aplikację adaptacyjnego regulatora neuronowo-rozmytego w sterowaniu układem dwumasowym. W trakcie wyznaczania poprawek dla przestrajanego modelu wykorzystano dodatkowy predyktor neuronowy. Struktura sterowania zawierająca opisywany regulator charakteryzuje się wysoką dynamiką oraz szybką reakcją na zmiany parametrów obiektu. Jednak w etapie projektowania, występuje konieczność wyznaczania stałych algorytmu adaptacyjnego, trudnych do jednoznacznego opisanie zależnościami matematycznymi. W przedstawionym zadaniu zastosowano algorytm GWO (Grey Wolf Optimizer). Artykuł zawiera opis kolejnych etapów przetwarzania w trakcie optymalizacji oraz uzyskane wyniki badań symulacyjnych oraz eksperymentalnych.

Abstract. This article presents application of adaptive neuro-fuzzy controller for two-mass system. Updates of weights are calculated based on signal from additional neural predictor. High dynamic of control structure with proposed model and fast reaction against changes of plant parameters are observed. However, design process needs selection of learning coefficients used in calculations of adaptive law. Mathematical representation of those parameters is difficult. For this purpose GWO (Grey Wolf Optimizer) is implemented. Paper contains description of several stages of optimization and obtained results. (The GWO algorithm applied for optimization of adaptive neuro-fuzzy controller used for two-mass system).

Słowa kluczowe: optymalizacja, algorytm GWO, adaptacyjny regulator neuronowo-rozmyty, sterowanie prędkością, napęd elektryczny.

Keywords: optimization, GWO algorithm, adaptive neuro-fuzzy controller, speed control, electric drive.

Wprowadzenie

W niniejszym artykule opisano założenia teoretyczne oraz wstępne testy zmodyfikowanego adaptacyjnego regulatora neuronowo-rozmytego, zastosowanego w pętli regulacji prędkości napędu elektrycznego zawierającego sprzężyste sprzęgło [1]-[3]. Obliczenia korekt (*on-line*) współczynników wagowych regulatora realizowane są na podstawie sygnału z predyktora neuronowego. Efektem takiej budowy modelu jest lepsza reakcja na zmiany sygnału zadanego oraz zakłócenia wprowadzane w obiekcie. Zatem, ogólna struktura zawiera dwa modele przestrajane w trakcie pracy napędu. Szczegóły regulatora zostały przedstawione w dalszej części artykułu. Wykazano również, że istotnym elementem etapu projektowania jest poprawne wyznaczenie współczynników wprowadzanych do algorytmu adaptacji. Jednak zależności matematyczne opisujące te wartości są trudne do uzyskania. W związku z tym, w celu optymalizacji tych parametrów zastosowano metaheurystyczny algorytm *Grey Wolf Optimizer* [4].

Analizując publikację przedstawiającą opisy teoretyczne oraz aplikacje, w różnych dziedzinach, nowoczesnych algorytmów optymalizacyjnych, możliwe jest zaobserwowanie wzrostu zainteresowania tymi metodami. Pierwsze zagadnienia związane z inteligencją roju (*Swarm Intelligence*) zostały zdefiniowane w artykule [5]. Jednak bardzo wiele z nich, nowych algorytmów, zaproponowano w ostatnich latach. Charakterystyczną cechą tych technik jest wykorzystanie obserwacji zachowań grup organizmów. Ograniczając rozważania do zastosowań w zagadnieniach dotyczących napędów elektrycznych, wśród najpopularniejszych z nich należy wymienić efektywne przykłady aplikacji: *Particle Swarm Optimization* [6], [7], *Artificial Bee Colony* [8], [9] oraz algorytm *BAT* [10], [11]. Wymienione metody stają się coraz bardziej popularne ze względu na charakterystyczne, korzystne cechy (dotyczące założeń teoretycznych oraz związanych z implementacją praktyczną). Poprawki, wyznaczone przez wiele algorytmów, wyliczane są najczęściej z wykorzystaniem wartości pochodnych funkcji celu względem współczynników. Wymienione, podstawowe założenia optymalizacji mogą być problematyczne w praktycznej implementacji dla konkretnego zadania, szczególnie przy założeniu obliczeń realizowanych *on-line* (równoległe do

pracy całego procesu). Odpowiednim przykładem takiego zadania jest aplikacja adaptacyjnych regulatorów bazujących na modelach neuronowych [12]-[14]. Jedną z najbardziej istotnych zalet wynikających z zastosowania algorytmów optymalizacyjnych wykorzystujących metody inspirowane naturalnymi populacjami, jest brak konieczności wyznaczania gradientu funkcji celu. Możliwe jest również zaobserwowanie pominięcia złożonych operacji elementarnych, wykonywanych na danych przetwarzanych przez *SI*. Najistotniejszymi czynnikami determinującymi zawartość obliczeń, w rezultacie również czas operacji, są: liczba osobników populacji oraz długość wektora danych opisujących proces.

Publikacja składa się z sześciu głównych rozdziałów. Po wprowadzeniu, przedstawiającym najistotniejsze zagadnienia dotyczące analizowanej tematyki, omówiona została ogólna postać struktury sterowania. W dalszej części opisano szczegóły zmodyfikowanego regulatora neuronowo-rozmytego, a następnie zaprezentowano zasadę obliczeń algorytmu optymalizacyjnego GWO. Artykuł został zakończony przedstawieniem weryfikacji eksperymentalnej układu regulacji oraz analizą uzyskanych wyników.

Opis analizowanej struktury sterowania

Analizowana struktura sterowania oparta została o model kaskadowy, tzn. zawiera dwie pętle regulacji, wewnętrzną związaną z kształtowaniem momentu elektromagnetycznego oraz nadrzędny układ regulacji prędkości. Układ dwumasowy opisuje poniższa zależność [2], [11], [15]:

$$(1) \quad \frac{d}{dt} \begin{bmatrix} \omega_1(t) \\ \omega_2(t) \\ m_s(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & -\frac{1}{T_1} \\ 0 & 0 & \frac{1}{T_2} \\ \frac{1}{T_c} & -\frac{1}{T_c} & 0 \end{bmatrix} \begin{bmatrix} \omega_1(t) \\ \omega_2(t) \\ m_s(t) \end{bmatrix} + \begin{bmatrix} \frac{1}{T_1} & 0 \\ 0 & -\frac{1}{T_2} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} m_e(t) \\ m_L(t) \end{bmatrix},$$

gdzie: ω_1 – prędkość silnika, ω_2 – prędkość maszyny roboczej, m_s – moment skrętny, m_e – moment elektromagnetyczny, m_L – moment obciążenia, T_1 – mechaniczna stała czasowa silnika napędzającego, T_2 –

mechaniczna stała czasowa maszyny roboczej, T_c – stała czasowa elementu sprzęgającego. Najistotniejszym założeniem dla pętli regulacji momentu m_e jest możliwie najszybsze generowanie tej zmiennej stanu na podstawie sygnału zadanego z regulatora prędkości m_{ez} . Zgodnie z teorią sterowania wprowadzono opis tej części napędu za pomocą następującej transmitancji:

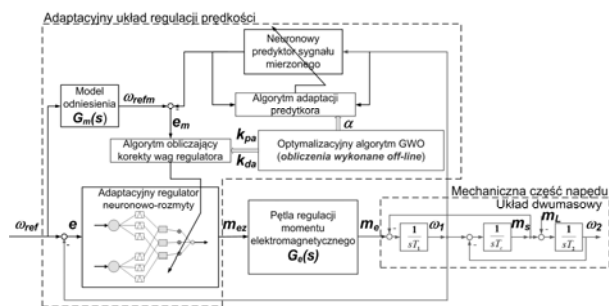
$$(2) \quad G_e(s) = \frac{1}{T_{me}s + 1},$$

gdzie: T_{me} – stała czasowa obwodu elektromagnetycznego.

W układzie regulacji wykorzystano dodatkowy model odniesienia $G_m(s)$, korygujący sygnał zadany dla algorytmu adaptacyjnego. Na podstawie tej wartości wyznaczany jest błąd e_m . Wprowadzona transmitancja opisana jest wzorem:

$$(3) \quad G_m(s) = \frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2},$$

w którym: ζ jest współczynnikiem tłumienia, natomiast ω opisuje pulsację rezonansową.



Rys. 1. Schemat przedstawiający przetwarzanie danych w strukturze sterowania

Schemat przedstawiający poszczególne elementy układu regulacji zamieszczony został na rysunku 1. Głównym elementem w pętli sterowania prędkością jest model neuronowo-rozmyty, którego parametry przestrajano *on-line*, podobnie jak bloku predykcji. Szczegóły dotyczące budowy regulatora adaptacyjnego oraz predyktora neuronowego opisano w kolejnym punkcie artykułu. Przyjęto następujące, nominalne wartości sterowanego obiektu: $T_1=T_2=203\text{ms}$, $T_c=1,2\text{ms}$ oraz $T_{me}=0\text{s}$ (w badaniach symulacyjnych). Założono pełną dostępność pomiarową wszystkich wymaganych zmiennych napędu elektrycznego.

Adaptacyjny regulator neuronowo-rozmyty współpracujący z neuronowym predyktorem prędkości

Regulatory rozmyte są modelami umożliwiającymi uzyskanie wysokiej precyzji sterowania obiektem lub procesem przemysłowym. Wśród charakterystycznych cech tego typu układów należy wymienić specyficzny proces projektowania, którego podstawę stanowi baza reguł opisująca obserwacje modelowanego systemu. Struktura regulatora neuronowo-rozmytego została szczegółowo przedstawiona w publikacjach [16], [17]. Kolejne etapy przetwarzania realizowane są sekwencyjnie w warstwach modelu. Wstępne działania związane są z ewentualnym skalowaniem sygnałów wejściowych, sprawdzaniem ograniczeń oraz formowaniem wektora danych dla funkcji przynależności. W opisywanym układzie regulacji zastosowano dwa wejścia:

$$(4) \quad X_{nf} = \{x_1 = e(k), x_2 = \Delta e(k)\}.$$

Zamieszczone w warstwie drugiej trójkątne funkcje przynależności f_{mf} realizują operacje rozmywania danych. Każdy węzeł warstwy trzeciej reprezentuje przesłankę reguły rozmytej. W testowanym modelu zastosowano 9 reguł. Zastosowany operator Π odpowiada wprowadzonemu działaniu *T-norm*, zgodnie z którym przyjęto funkcję mnożącą sygnały wejściowe każdej reguły. W kolejnej warstwie wykonywane jest tzw. wyostrzenie wartości, obliczana jest wartość wyjściowa modelu rozmytego y_{fnn} . Sygnały wejściowe x_o^3 , na tym etapie, mnożone są z wartościami tzw. singletonów w_o :

$$(5) \quad x_4 = \sum_{o=1}^R w_o x_o^3,$$

gdzie: R – liczba reguł ($R=9$),

w takim przypadku wyjście warstwy czwartej (przy liniowej funkcji f_4):

$$(6) \quad y_4 = y_{fnn} = f_4(x_4).$$

Model neuronowo-rozmyty został przedstawiony na rysunku 2. W analizowanym regulatorze, adaptacji podlegają wartości współczynników W_o , które przeliczane są w każdej iteracji k (uwzględniając zapis macierzowy):

$$(7) \quad W_o(k+1) = W_o(k) + \Delta W_o(k).$$

W celu wyznaczenia korekt $\Delta W_o(k)$, zdefiniowano następującą funkcję celu:

$$(8) \quad E_c = \frac{1}{2}(\omega_{refm} - \omega_1)^2 = \frac{1}{2}(e_m)^2.$$

Precyzując, dla pojedynczego połączenia warstwy ostatniej modelu:

$$(9) \quad \Delta W_{oh} = -\eta \frac{\partial E_c}{\partial W_o},$$

$$(10) \quad \frac{\partial E_c}{\partial W_o} = \delta_o x_o^3,$$

stosując tzw. regułę łańcuchową:

$$(11) \quad \delta_o = \frac{\partial E_c}{\partial e_m} \frac{\partial e_m}{\partial \omega_1} \frac{\partial \omega_1}{\partial y_{fnn}} \frac{\partial y_{fnn}}{\partial x_4} = \frac{\partial E_c}{\partial y_{fnn}} \frac{\partial y_{fnn}}{\partial x_4}.$$

W praktycznych zastosowaniach obliczenia (11) mogą być utrudnione, dlatego wykorzystano aproksymację pochodnej funkcji celu E_c względem wyjścia y_{fnn} , poprzez zależność [18]:

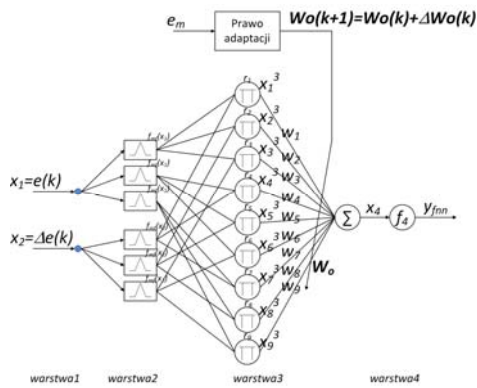
$$(12) \quad \frac{\partial E_c}{\partial y_{fnn}} = k_{padapt} e_m + k_{daadapt} \frac{de_m}{dt},$$

w ostatecznej implementacji należy również pamiętać, że zgodnie z założeniami (6):

$$(13) \quad y_{fnn} = x_4.$$

W celu redukcji definiowanych parametrów projektowych wprowadzono następujące równania:

$$(14) \quad \begin{aligned} k_{pa} &= \eta k_{padapt} \\ k_{da} &= \eta k_{daadapt} \end{aligned}$$



Rys.2. Model neuronowo-rozmyty zastosowany w regulatorze układu dwumasowego

Sieci neuronowe są modelami, których obliczenia inspirowane są obserwacjami naturalnych komórek nerwowych. Bazujące na tej podstawie algorytmy przetwarzania danych są obecnie bardzo popularne, ze względu na uzyskiwane właściwości sieci: zdolności adaptacyjne (odtworzenie danych), równoległe obliczenia, aproksymacja, generalizacja wartości wykorzystanych w treningu oraz predykcja [19]. Najistotniejsze rozwiązania dotyczące ostatniej cechy, realizowane są dwoma sposobami. Pierwsze podejście zakłada trening na podstawie uprzednio zebranego, odpowiednio opracowanego zbioru danych. Uzyskiwanie sygnałów z przyszłych próbek wymaga jednak złożonego projektowania sieci neuronowych [20]. Bardzo interesujący wydaje się koncept polegający na zastosowaniu, w predykcji sygnałów, modeli neuronowych trenowanych *on-line*. W ten sposób możliwe jest uzyskanie predyktora, który nie wymaga równań opisujących odtwarzany sygnał, a także zakłada uproszczone poszukiwanie rozwiązań. Generowanie danych wyprzedzonych w stosunku do oryginalnej zmiennej stanu, może być obliczone z wykorzystaniem szybkiego dostarczania sieci neuronowej. Należy podkreślić, że tak trenowany model dąży do odtwarzania aktualnego sygnału zadanego [21], [22]. Dodatkowo, zakładając kaskadową budowę q jednostek neuronowych, gdzie pierwsza jest dostarczana do sygnału wejściowego $in(k)$, natomiast każda kolejna dąży do odtwarzania aktualnie zadanego sygnału z poprzedniego modelu, na wyjściu całej struktury możliwe jest uzyskanie $in(k+q)$ [23]. W celu uzyskania szybkich obliczeń w poszczególnych etapach predyktora, zastosowano prosty model neuronu liniowego (**ADaptive Linear NEuron**) [20]. Wektor wejściowy X poszczególnych modułów predyktora zawiera sygnał wejściowy in (w opisywanej aplikacji, w pierwszej fazie obliczeń ω_1) oraz opóźnione wartości tej zmiennej. Adaptacja pojedynczego modelu *ADALINE* wymaga zdefiniowania funkcji celu:

$$(15) E_{Adaline}(\mathbf{w}_a) = (y_{nprcdv}(k) - ir(k))^2 = 0,5 \left(\sum_{a=1}^A x_a(k) w_a(k) - ir(k) \right)^2,$$

gdzie: y_{nprcdv} – wartość wyjściowa kolejnego modelu *ADALINE*, ir – wartość wejściowa modelu *ADALINE*, x_a – wartość elementu wektora wejściowego, w_a – współczynnik wagowy.

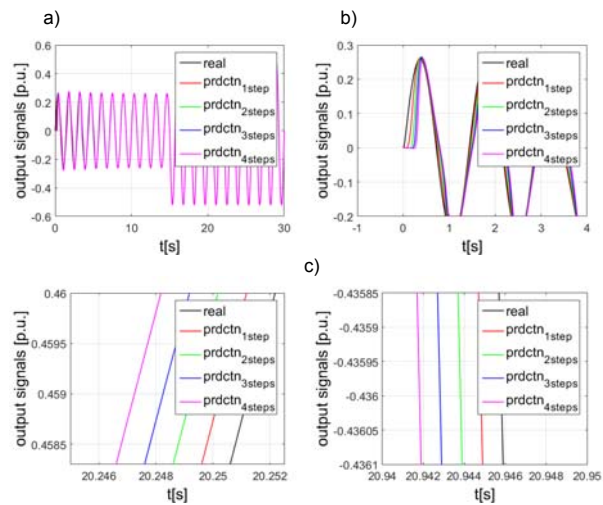
Na podstawie (15) wyznaczony został gradient funkcji:

$$(16) \mathbf{g}(k) = \frac{\partial E_{Adaline}}{\partial \mathbf{w}_a} = \left(\sum_{a=1}^A x_a(k) w_a(k) - ir(k) \right) x_a(k),$$

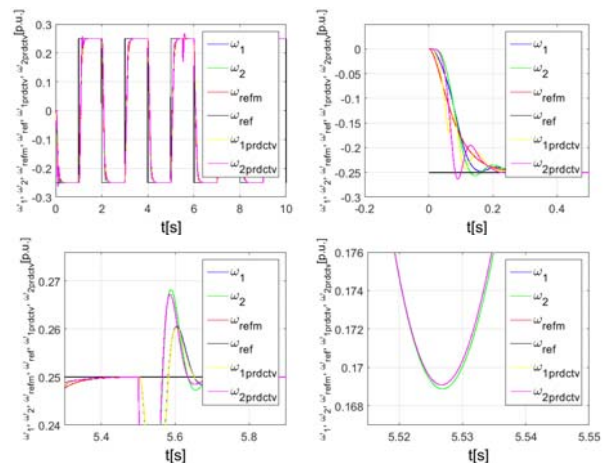
który wykorzystano w celu opisanego prawa adaptacji:

$$(17) \mathbf{w}_a(k+1) = \mathbf{w}_a(k) - \alpha \left(\sum_{a=1}^A x_a(k) w_a(k) - ir(k) \right) x_a(k).$$

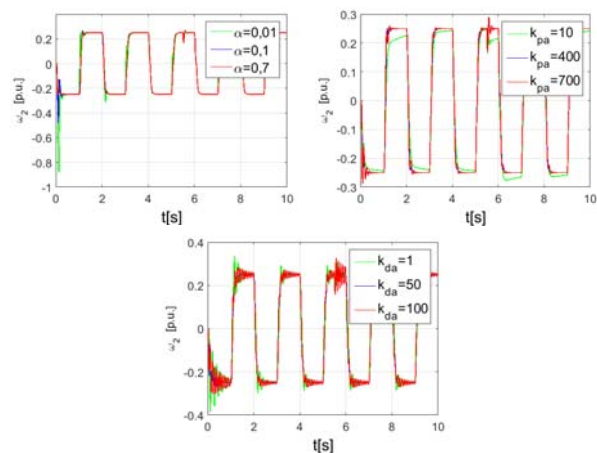
Opisane powyżej obliczenia realizowane są we wszystkich etapach przetwarzania predyktora.



Rys.3. Przykładowe sygnały wyjściowe bloku neuronowego



Rys.4. Przebiegi prędkości w strukturze sterowania układem dwumasowym



Rys.5. Wpływ stałych współczynników adaptacyjnych regulatora prędkości na przebiegi wyjściowe

Przykład przebiegów, prezentujących działanie bloku przetwarzającego wstępnie dane dla algorytmu adaptacji

regulatora neuronowo-rozmytego, przedstawiono na rysunku 3. W analizowanym regulatorze wykorzystano cztery szeregowo połączone jednostki predyktora (modele ADALINE). Wykonany model podłączony został do zmiennej wyjściowej układu regulacji. Trajektorię zadaną stanowił sygnał sinusoidalny, którego amplitudę zmieniano w trakcie symulacji (rysunek 3a). Obserwowane są sygnały poszczególnych etapów przetwarzania modelu neuronowego. Na powiększonych wykresach widoczny jest proces dostrajania bloku neuronowego, startującego z losowymi wartościami początkowymi wag. W tej fazie działania struktury, sygnał rzeczywisty wyprzedza wartości wyjściowe poszczególnych jednostek predyktora (rysunek 3b). Po kilku okresach sygnału zadanego, można zauważyć efekt treningu sieci neuronowej, a w rezultacie zmianę kolejności przebiegów (rysunek 3c).

Na rysunku 4 zaprezentowano sygnały zadane oraz wyjściowe prędkości silników układu napędowego. Celem przeprowadzonych testów była obserwacja wpływu wprowadzenia modelu neuronowego do regulatora neuronowo-rozmytego. Przeprowadzone badania prezentują najistotniejsze różnice w stanach przejściowych pracy napędu. Analizując wyjście całej struktury, czyli prędkość obciążenia ω_2 (uzyskane bez bloku neuronowego) oraz $\omega_{2prcdtv}$, obserwowalne są właściwości analogiczne do wniosków uzyskanych wcześniej, dla predyktora (rysunek 3). Należy podkreślić, że wyniki przeprowadzono dla identycznych warunków pracy regulatorów, wprowadzono jedynie model neuronowy. Na wykresie (do około $t=0,1$ s) widoczny jest wpływ początkowego dostrajania sieci, jednak nawet w tym obszarze można zaobserwować, że $\omega_{2prcdtv}$ (kolor różowy) osiąga oraz stabilizuje się szybciej na poziomie zadanym niż ω_2 (kolor zielony). W czasie $t=5,5$ s. przełączano obciążenie, widoczne jest poprawienie reakcji regulatora po wprowadzeniu bloku predyktora. Poprzez wcześniejsze uzyskanie informacji o zmianach prędkości ω_1 , przyspieszone zostało działanie części obliczającej poprawki wag regulatora, a ostatecznie uzyskano poprawniejsze oddziaływanie na wartość wyjściową napędu.

W następnych badaniach zmieniana była wartość parametrów w układzie adaptacji testowanego regulatora prędkości (rysunek 5). Obserwowalna jest duża podatność regulatora na zakłócenia, przy błędnie wyznaczonych parametrach: α , k_{pa} , k_{da} . Szczególnie wyróżnia się, w każdym przypadku początek wykresów (do około $t=1$ s). Nieodpowiednio dobrane współczynniki układu adaptacji mogą prowadzić również do oscylacji w dalszych fazach działania napędu. Uzyskane wyniki uzasadniają dalsze prace - aplikację algorytmu GWO w optymalizacji wspomnianych parametrów.

Algorytm GWO

Grey Wolf Optimizer jest metaheurystycznym algorytmem, którego obliczenia realizowane są iteracyjnie w celu wprowadzania modyfikacji analizowanej populacji (przestrzeni rozwiązań) oraz wyznaczania najlepszych osobników (optymalnych wartości przetwarzanych zmiennych). Program wstępnie wymaga zdefiniowania parametrów dotyczących populacji oraz założeń związanych z obliczeniami. Wprowadzane są następujące informacje: deklarujące liczbę iteracji (k_{max}) lub warunki zatrzymania algorytmu, a także liczba oraz wartości ograniczeń optymalizowanych parametrów, jak również liczebność populacji (n). W kolejnym kroku wyznaczane są początkowe parametry opisujące sposób zmian przetwarzanego zbioru danych [4]:

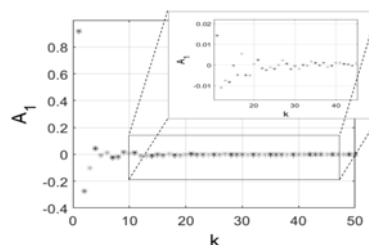
$$(18) \quad \mathbf{A} = 2a\mathbf{r}_1 - a,$$

$$(19) \quad \mathbf{C} = 2\mathbf{r}_2,$$

gdzie: \mathbf{r}_1 oraz \mathbf{r}_2 są losowymi wektorami z wartościami zawartymi w przedziale $<0,1>$, natomiast a jest wartością stałą dla danej iteracji k :

$$(20) \quad a = 2 - \frac{2k}{k_{max}} \quad \text{dla } k = 0 \dots k_{max}.$$

Należy zwrócić uwagę, że wartość parametru a , a w efekcie elementów wektora \mathbf{A} , maleje w trakcie obliczeń (w zakresie od 2 do 0), celem tego współczynnika jest zmniejszanie zakresu poszukiwań najlepszych rozwiązań. Zakładane jest działanie algorytmu GWO (w początkowej fazie), które ma na celu wyznaczanie oraz analizowanie wyników o wartościach znacznie różniących się od aktualnych (z poprzedniej iteracji). Następnie optymalizacja koncentruje się na wyliczaniu kolejnych wartości jedynie wokół elementów odpowiadających najmniejszej wartości funkcji celu. W ten sposób, pierwsze iteracje pozwalają zapobiegać uzyskiwaniu rozwiązań lokalnych, natomiast w dalszych krokach zwiększana jest precyzja rozwiązań poprzez poszukiwania w ograniczonym obszarze (rysunek 6).



Rys.6. Przykładowy przebieg parametrów wprowadzających ograniczanie przestrzeni poszukiwań algorytmu GWO

Następnie wyznaczana jest wartość funkcji celu dla każdego rozwiązania (w trakcie inicjalizacji wyznaczone losowo). W tym zadaniu przyjęto następującą zależność:

$$(21) \quad f_{cGWO} = \left(\sum_{i=1}^p |\omega_{refm} - \omega_2|_{t_s} \right) + q \left(\sum_{i=1}^k |\omega_{refm} - \omega_2|_{t_s} \right),$$

gdzie: ω_{ref} – prędkość zadana, t_s – krok obliczeniowy (element skalujący), q – stały współczynnik, p – całkowita liczba próbek, k – liczba próbek.

Na podstawie uzyskanych wartości, wyliczane są najlepsze aktualne rozwiązania – X_α , X_β oraz X_σ . W następnym kroku rozpoczyna się pierwsza iteracja algorytmu, która związana jest wyznaczeniem zaktualizowanej pozycji każdego z osobników populacji zgodnie z zależnością:

$$(22) \quad X_{k+1} = \frac{X_1 + X_2 + X_3}{3},$$

opisującą średnią z wartości:

$$(23) \quad X_1 = X_\alpha - A_1 D_\alpha,$$

$$(24) \quad X_2 = X_\beta - A_2 D_\beta,$$

$$(25) \quad X_3 = X_\sigma - A_3 D_\sigma,$$

powyższe wzory uwzględniają odległości pomiędzy aktualizowanym elementem zbioru rozwiązań X oraz najlepszymi wynikami z poprzedniej iteracji X_α , X_β , X_σ :

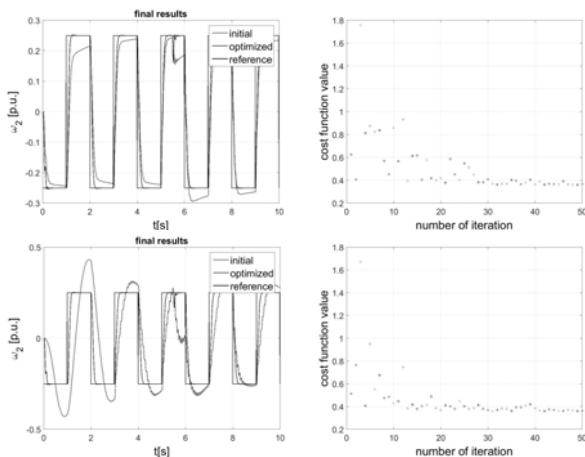
$$(26) \quad D_\alpha = |C_1 X_\alpha - X|,$$

$$(27) \quad D_{\beta} = |C_2 X_{\beta} - X|,$$

$$(28) \quad D_{\sigma} = |C_3 X_{\sigma} - X|.$$

Kolejny punkt obliczeń algorytmu zakłada ponowne przeliczenia wartości A , a oraz C . Po kolejnej ocenie nowego zbioru rozwiązań (za pomocą funkcji celu (21)), wyznaczone są najlepsze wartości uzyskane z uwzględnieniem obecnej iteracji. W zależności od spełnienia odpowiednich warunków (brak osiągnięcia k_{max} lub założonej wartości funkcji celu dla najlepszego rozwiązania) wykonywane jest kolejne powtórzenie działania algorytmu (rozpoczynane od obliczenia nowej pozycji osobników (22)-(28)).

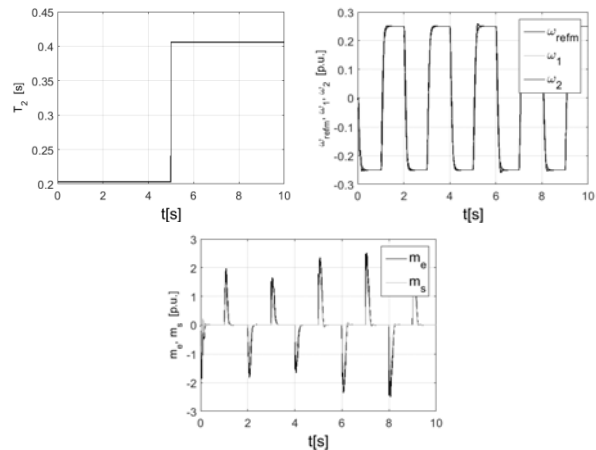
Rysunek 7 przedstawia efekt optymalizacji wybranych parametrów analizowanej struktury sterowania za pomocą algorytmu GWO. Poszczególne testy przeprowadzono dla tych samych parametrów regulatora, natomiast losowane były wartości początkowe współczynników uczenia. Zaprezentowano dwa przykłady działania optymalizatora. Dla poszczególnych badań wykreślono także wartości funkcji celu uzyskiwane w poszczególnych iteracjach pracy napisanego programu. Ogólny trend prezentuje zmniejszanie błędów, jednak widoczne są szczególne symulacje, w których algorytm sprawdzał rozwiązania odbiegające od ostatnich najlepszych wyników. Wspomniana właściwość jest obserwowalna w początkowych iteracjach. Efekt wprowadzanych zmian, ostateczna wartość wyznaczonych parametrów, po wprowadzeniu do układu regulacji, pozwala na zdecydowaną poprawę wyników oraz uzyskanie bardzo zbliżonych przebiegów prędkości ω_2 po kolejnych optymalizacjach (istotna właściwość GWO). W konfiguracji programu przyjęto: $n=30$, $k_{max}=50$. Dla wektora optymalizowanych wartości: $\{\alpha, k_{pa}, k_{da}\}$, wektor dolnych ograniczeń zawierał: $\{0;0;0\}$, natomiast górne wartości graniczne: $\{1;500;100\}$. Przykładowe wartości początkowe wylosowano następująco: $\{0,9541;5,4281;5,4011\}$, po optymalizacji otrzymano: $\{1,0000;200,0089;12,0373\}$.



Rys.7. Przebiegi prędkości maszyny roboczej w strukturze sterowania z regulatorem neuronowo-rozmytym współpracującym z predyktorem neuronowym – przykładowe efekty optymalizacji parametrów za pomocą algorytmu GWO

Ostatnim etapem badań symulacyjnych była analiza działania zoptymalizowanego układu regulacji w przypadku zmiany parametrów układu dwumasowego. Zarejestrowane przebiegi obu prędkości oraz momentów: skrętnego oraz elektromagnetycznego zamieszczone zostały na rysunku 8. W trakcie symulacji zakłócana była wartość stałej T_2 . Adaptacja regulatora prędkości wymusza zwiększenie

momentu elektromagnetycznego po wprowadzonych fluktuacjach. Większa wartość T_2 powoduje, że w prędkościach pojawiają się nieznaczne przeregulowania, które są redukowane w kolejnych nawrotach napędu.



Rys.8. Przebiegi zmiennych stanu układzie regulacji prędkości napędu, uwzględniającym parametry wyznaczone za pomocą algorytmu GWO – wpływ zmian stałej T_2

Badania eksperymentalne

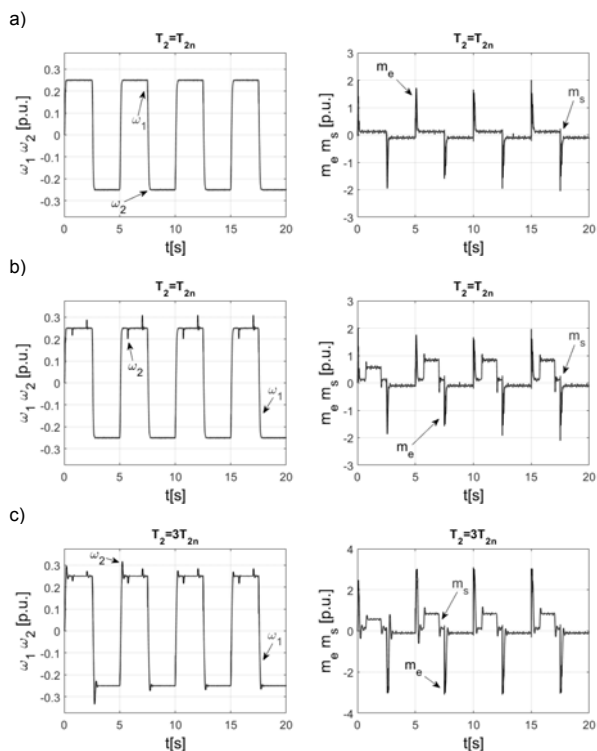
Opisany we wcześniejszych częściach artykułu adaptacyjny regulator neuronowo-rozmyty został poddany testom na stanowisku laboratoryjnym. Dokładny opis rzeczywistego napędu przedstawiony został w artykułach [2], [11], [17].

Zarejestrowane wyniki badań zamieszczono na rysunku 9. Trajektoria zadana napędu wymusza, w każdym przypadku, cykliczne nawroty z prędkością ustaloną równą 25% wartości znamionowej. Przebiegi prędkości silników zmieniają się zgodnie, również precyzyjnie, z sygnałem referencyjnym (rysunek 9a). Regulator skutecznie dostraja się do obiektu. Należy również pamiętać, że w badaniach na obiekcie rzeczywistym pojawiają się dodatkowe utrudnienia związane z nieliniowościami napędu, odpowiadające oddziaływaniu np. tarcia silników lub luzu elementów łączących. Tego typu zakłócenia nie były brane pod uwagę w modelu układu dwumasowego wykorzystanym w projektowaniu regulatora (rysunek 1). W kolejnych badaniach powtórzono pierwszy test, jednak po ustaleniu prędkości na poziomie zadanym przełączano obciążenie (rysunek 9b). Obserwowana jest bardzo szybka reakcja regulatora (poprzez forsowanie odpowiednich zmian momentu elektromagnetycznego) na pojawiające się krótkotrwałe zakłócenia w przebiegach obu prędkości. Zmiana stałej czasowej T_2 , w badaniach laboratoryjnych, jest uzyskiwana poprzez zamontowanie kolejnych tarcz obciążających wał maszyny roboczej. Rezultaty otrzymane dla takiego stanu pracy zaprezentowano na rysunku 9c. Warto zaobserwować, że w stanach przejściowych, zgodnie z oczekiwaniami, moment elektromagnetyczny osiąga większe wartości, w porównaniu do wcześniejszych startów napędu ze znamionowymi parametrami. Przy pierwszych nawrotach obserwowane jest znaczne przekroczenie zadanej wartości w sygnałach prędkości, jednak algorytm adaptacji dostraja parametry W_0 regulatora, co prowadzi do redukcji przeregulowań w kolejnych cyklach pracy napędu.

Podsumowanie

Przedstawiony artykuł opisuje efektywne, oryginalne zastosowanie adaptacyjnego regulatora zastosowanego algorytmu GWO w celu optymalizacji parametrów dla

napędu z elastycznym sprzęgłem. Głównym elementem pętli prędkościowej był model neuronowo-rozmyty współpracujący z predykcyjnym blokiem neuronowym.



Rys.9. Przebiegi zmiennych stanu w układzie napędowym – badania eksperymentalne

Poprzez taką konstrukcję, uzyskano przyspieszoną reakcję oddziaływania części adaptacyjnej na pracę całej struktury. Analizowana metoda optymalizacji współczynników bloków adaptacyjnych została wprowadzona w etapie projektowania układu regulacji. Uzyskano bardzo dobre rezultaty działania struktury sterowania z parametrami (stałe współczynniki algorytmów adaptacyjnych) wyznaczonymi przez GWO. Należy również podkreślić powtarzalność uzyskiwanych wyników oraz krótki czas obliczeń opracowanego programu. Sterowanie prędkością układu dwumasowego za pomocą opisanego modelu, zrealizowane z założeniem wykorzystania zredukowanego wektora zmiennych stanu, umożliwiło uzyskanie precyzyjnej kontroli napędu. Dodatkowo wprowadzanie poprawek w nastawach regulatora (*on-line*) prowadzi do dostrajania modelu po załączeniu dodatkowych tarcz obciążających (zmieniających parametry układu dwumasowego).

Autor: dr hab. inż. Marcin Kamiński, Politechnika Wroclawska, Katedra Maszyn, Napędów i Pomiarów Elektrycznych, ul. Smoluchowskiego 19, 50-372 Wrocław, E-mail: marcin.kaminski@pwr.edu.pl

LITERATURA

[1] Erenturk K., Fractional-Order PI^λD^μ and Active Disturbance Rejection Control of Nonlinear Two-Mass Drive System, *IEEE Trans. Ind. Electronics*, 60 (2013), n. 9, 3806-3813
 [2] Ortowska-Kowalska T., Kamiński M., Szabat K., Implementation of a Sliding-Mode Controller With an Integral Function and Fuzzy Gain Value for the Electrical Drive With an

Elastic Joint, *IEEE Trans. Ind. Electronics*, 57 (2010), n. 4, 1309-1317
 [3] Saarakkala S.E., Hinkkanen M., State-Space Speed Control of Two-Mass Mechanical Systems: Analytical Tuning and Experimental Evaluation, *IEEE Trans. Ind. App.*, 50 (2014), n. 5, 3428-3437
 [4] Mirjalili S., Mirjalilib S.M., Lewis A., Grey Wolf Optimizer, *Advances in Engineering Software*, 69 (2014), 46-61
 [5] Beni G., Wang J., Swarm Intelligence in Cellular Robotic Systems, *Robots and Biological Systems: Towards a New Bionics?*, (1993), 703-712
 [6] 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
 [7] Calvini M., Carpita M., Formentini A., Marchesoni M., PSO-Based Self-Commissioning of Electrical Motor Drives, *IEEE Trans. Ind. Electronics*, 62 (2015), n. 2, 768-776
 [8] Tarczewski T., Grzesiak L.M., Artificial Bee Colony Based Auto-Tuning of PMSM State Feedback Speed Controller, *IEEE International Power Electronics and Motion Control Conference (PEMC)*, (2016), 1155-1160
 [9] Rajasekhar A., Das S., Abraham A., Fractional Order PID controller design for speed control of chopper fed DC Motor Drive using Artificial Bee Colony algorithm, *World Congress on Nature and Biologically Inspired Computing*, (2013), 259-266
 [10] 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
 [11] Kamiński M., Recurrent Neural Controller Applied for Two-Mass System, 21st International Conference on Methods and Models in Automation and Robotics (MMAR), (2016), 128-133
 [12] 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
 [13] He W., Ouyang Y., Hong J., Vibration Control of a Flexible Robotic Manipulator in the Presence of Input Deadzone, *IEEE Trans. Ind. Informatics*, 13 (2017), n. 1, 48-59
 [14] Brock S., Łuczak D., Nowopolski K., Pajchrowski T., Zawirski K., Two Approaches to Speed Control for Multi-Mass System With Variable Mechanical Parameters, *IEEE Trans. Ind. Electronics*, 64 (2017), n. 4, 3338-3347
 [15] Rossi M., Mauri M., Carmeli M.S., Castelli-Dezza F., Latency Effect in a Variable Speed Control on Torsional Response of Elastic Drive Systems, XXII International Conference on Electrical Machines (ICEM), (2016), 2280-2286
 [16] Lin F.-J., Wai R.-J., Adaptive Fuzzy-Neural-Network Control for Induction Spindle Motor Drive, *IEEE Trans. Energy Conversion*, 17 (2002), n. 4, 507-513
 [17] Szabat K., Direct and Indirect Adaptive Control of a Two-Mass Drive System — a Comparison, *Proceedings of the IEEE Int. Symp. Ind. Electronics*, (2008), 564-569
 [18] Lin F.-J., Wai R.-J., Hybrid Controller Using a Neural Network For a PM Synchronous Servo-Motor Drive, *IEE Proc.-Electr. Power Appl.*, 145 (1998), n. 3, 223 – 230
 [19] Osowski S., Sieci Neuronowe do Przetwarzania Informacji, Oficyna Wydawnicza Politechniki Warszawskiej, 2006
 [20] Ren H., Zhang Y., Nonlinear Control Based on an Improved Neural Predictive Control Scheme, 3rd International Conference on Computer Science and Network Technology (ICCSNT), (2013), 1222-1225
 [21] Kamran F., Harley R.G., Burton B., Habetler T.G., Brooke M.A., A Fast On-Line Neural-Network Training Algorithm for a Rectifier Regulator, *IEEE Trans. Power Electronics*, 13 (1998), n. 2, 366-371
 [22] Cocchi G., Uncini A., Subband Neural Networks Prediction for On-Line Audio Signal Recovery, *IEEE Trans. Neural Networks*, 13 (2002), n. 4, 867-876
 [23] Jin Y., Su C., Adaptive Model Predictive Control Using Diagonal Recurrent Neural Network, Fourth International Conference on Natural Computation (ICNC), (2008), 276-280