

doi:10.15199/48.2016.05.07

Adaptacyjne sterowanie neuronowym regulatorem prędkości dla napędu bezpośredniego z silnikiem PMSM

Streszczenie. W pracy przedstawiono wybrane właściwości adaptacyjnego regulatora neuronowego sterowanego online dla napędu bezpośredniego w warunkach mechanicznych zmian parametrów obiektu. Opisano algorytm RPROP, który porównano z wybranymi algorytmami uczenia sieci neuronowych sterowanych online, takich jak: algorytm wstecznej propagacji, momentowa metoda wstecznej propagacji i Quickprop. Algorytmy zostały zaimplementowane na stanowisku laboratoryjnym.

Abstract. The paper presents selected properties of adaptive neural network controller trained online with direct drive for mechanical conditions change object parameters. In the work was described algorithm RPROP and it was compared with the selected training algorithms of neural networks trained online, such as: BackPropagation, Momentum BackPropagation and Quickprop. The algorithms were implemented on a laboratory stand. (*Adaptive control neural speed control for direct drive with PMSM motor*).

Słowa kluczowe: sterowanie adaptacyjne, sztuczne sieci neuronowe, napęd bezpośredni,

Keywords: adaptive control, artificial neural networks, direct drive,

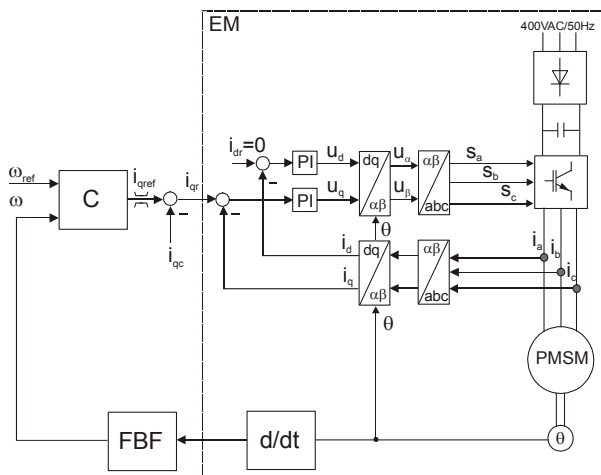
Wstęp

Napęd bezpośredni to takie rozwiązanie konstrukcyjne, w którym silnik elektryczny bezpośrednio jest połączony z maszyną roboczą bez przekładni mechanicznej. Od takiego silnika wymagana jest specjalna konstrukcja, która umożliwia pracę układu napędowego z małymi prędkościami obrotowymi, z reguły nieprzekraczającymi 100 obr./min, a często znacznie niższymi. Wzrost wymagań przemysłowych, obserwowany od wielu lat, obliuguje stosowanie, coraz to bardziej precyzyjnych rozwiązań, zwłaszcza w przemysłowych napędach robotów [1,2], maszynach technologicznych na przykład papierniczych [3], czy zwijarek i innych serwonapędach [4]. Brak przekładni mechanicznej wprowadza wiele zalet, takich jak: wyeliminowanie luzów wprowadzanych przez przekładnię, co poprawia dokładność statyczną pracy napędu, poprawia właściwości dynamiczne napędu, zwiększa sprawność układu napędowego (wyeliminowanie strat mechanicznych w przekładni) i niezawodność (mniejsza liczba elementów mechanicznych). Wymienione zalety napędów bezpośrednich wymagają jednak stosowania odpowiednich algorytmów regulacji w układzie sterowania. W napędzie bezpośrednim moment bezwładności maszyny roboczej jest wielokrotnie wyższy niż moment własny silnika, dlatego duże zmiany momentu bezwładności tej maszyny mają wpływ na warunki pracy układu napędowego (w napędzie klasycznym z przekładnią, moment bezwładności przeliczony na stronę silnika ma wartość podobną). W wielu układach napędowych moment bezwładności zależy od kąta położenia wału. Dlatego zachodzi potrzeba zaprojektowania układu regulacji niewrażliwego lub mało wrażliwego na zmiany mechaniczne parametrów. Zaproponowanie odpowiednich metod oraz takiego regulatora, który zapewni uzyskanie napędu elektrycznego o dużej precyzji jest zadaniem trudnym, którym zajmuje się wiele ośrodków na świecie. W celu zapewnienia poprawnej pracy układu napędowego stosuje się regulatory odporne [5] lub adaptacyjne [6], bardzo często z wykorzystaniem metod inteligencji obliczeniowej, zwłaszcza sztucznych sieci neuronowych [7,8,9,10]. Zastosowanie sieci neuronowych do regulacji prędkości obrotowej jest szeroko rozwijane w literaturze [10-17]. Stosowane są różne podejścia, które różnią się zarówno rodzajem SSN, algorytmem ich uczenia czy strukturą sieci neuronowej. W pracach [11,12,13] przedstawiono dwie różne koncepcje adaptacyjnego regulatora neuronowego. Regulator

adaptacyjny z modelem referencyjnym przedstawiono w pracy [13], w którym regulator neuronowy uczony jest na podstawie błędu, jaki powstaje pomiędzy wzorcowym sygnałem wyjściowym z modelu referencyjnego, a rzeczywistym sygnałem regulowanego obiektu. W pracy [14] przedstawiono koncepcję z regulatorem neuronowym ze sprzężeniem zwrotnym od prądu zadanego. Inną ciekawą koncepcją, w której regulator neuronowy uczony jest na podstawie własnego błędu regulacji przedstawiono w pracach [11, 22]. W publikacjach [15,16] zastosowano model odwrotny obiektu. Wymaga to jednak opracowania modelu neuronowego na podstawie danych charakteryzujących obiekt. Wadą takiego rozwiązania jest to, że pojawiają się problemy związane z wyborem: danych do treningu, parametrów uczenia oraz struktury regulatora neuronowego. Często proponowaną metodą jest regulacja adaptacyjna z modelem odniesienia MRAC [13,17]. W [8] zastosowano SSN do uaktualniania nastaw regulatora klasycznego, w zależności od zmian parametrów obiektu. W pracy [5] wykorzystano SSN do kompensacji momentu obciążenia. W pracy [11], wykorzystano liniową sieć typu ADALINE jako regulator prędkości w napędzie z połączeniem sprężystym. W wielu tych pracach wykorzystuje się znany algorytm wstecznej propagacji błędu do uczenia SSN. Uczenie sieci neuronowej *offline* jest jednak dość skomplikowane i czasochłonne, ponieważ wymaga to znacznie bardziej złożonej strategii projektowania, głównie ze względu na: odpowiednią liczbę neuronów, warstw ukrytych oraz danych uczących [18]. W związku z tym zastosowanie ich jako regulatorów prędkości w układzie napędowym jest dość skomplikowane, zwłaszcza gdy mamy do czynienia ze zmianą parametrów obiektu. Układy z sieciami neuronowymi trenowanymi *online* pozbawione są tych wad. Aktualizacja wag sieci neuronowej w tych regulatorach następuje już po pierwszym uruchomieniu układu sterowania, w trakcie pierwszych okresów próbkowania regulatora. Spotyka się różne sposoby aktualizacji współczynników wag. Popularnym algorytmem jest wsteczna propagacja błędu. Podstawową wadą tego algorytmu jest wyznaczanie gradientu błędu dla każdej wagi, co komplikuje algorytm, zwiększa czas jego wykonania w układzie mikroprocesorowym.

Celem pracy jest przedstawienie i opisanie wybranych właściwości neuronowego regulatora prędkości, którego parametry zmieniane są *online* według szybkiego algorytmu RPROP [19] (*ang. Resilient backPROPagation*). W pracy przedstawiono również porównanie tego

algorytmu z innymi wybranymi metodami uczenia sieci neuronowych, takimi jak: algorytm wstecznej propagacji błędów *BP* (ang. *BackPropagation*), momentowa metoda wstecznej propagacji błędów *MBP* (ang. *Momentum BackPropagation*) i *Quickprop*. Do badań napędu bezpośrednio zastosowany został silnik synchroniczny z magnesami trwałymi (ang. *Permanent Magnet Synchronous Motor*) ze względu na bardzo dobre właściwości dynamiczne [21,22]. Schemat badanego napędu z *PMSM* pokazano na rys. 1. Zastosowany silnik sterowany jest wektorowo z zachowaniem stałego kąta mocy $\delta=\pi/2$, co odpowiada zerowej wartości prądu w osi *d*. Analizowany regulator prędkości wylicza wartość zadaną prądu w osi *q*, która odpowiada zadanemu momentowi elektromagnetycznemu silnika [22].



Rys.1. Struktura sterowania silnikiem PMSM (*FBF*- filtr antyrezonansowy, *EM* – część elektromechaniczna układu, *C*-regulator, i_{qref} - sygnał wyjściowy regulatora, i_{qc} - dodatkowe wejście kompensujące tarcie i moment tętający, i_{qr} - prąd zadany)

Model obiektu do ciągłej zmiany momentu bezwładności i obciążenia

Przy założeniu, że wszystkie części ruchome układu napędowego wirują z tą samą prędkością obrotową momenty bezwładności są niezmiennicze, wszystkie połączenia są sztywne, wówczas cały układ można potraktować jak bryłę sztywną o stałym momencie bezwładności *J*. Dla takich założeń równanie ruchu obrotowego można zapisać [21,22]:

$$(1) \quad m - m_L = J \frac{d\omega}{dt}$$

Jeżeli w układzie napędowym założymy brak połączeń sprężystych i zmian energii potencjalnej można zapisać wyrażenie [21,22]:

$$(2) \quad m_d \omega = \frac{d}{dt} \left(\frac{J \omega^2}{2} \right)$$

a następnie po przekształceniach uzyskać zależność:

$$(3) \quad m_d = J \frac{d\omega}{dt} + \frac{\omega}{2} \frac{dJ}{dt}$$

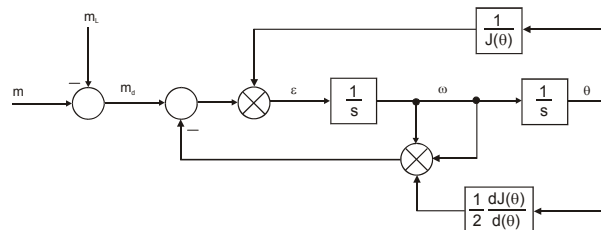
W układach w których moment bezwładności zależy od kąta obrotu (układy korbowe zamieniające ruch obrotowy na posuwisto zwrotny np. pompy i sprężarki tłokowe, prasy) można zapisać następujące wyrażenie [22]:

$$(4) \quad J(\theta) = J_{const} + J_{var}(\theta)$$

Wówczas po przekształceniach otrzymuje się zależność [21,22]:

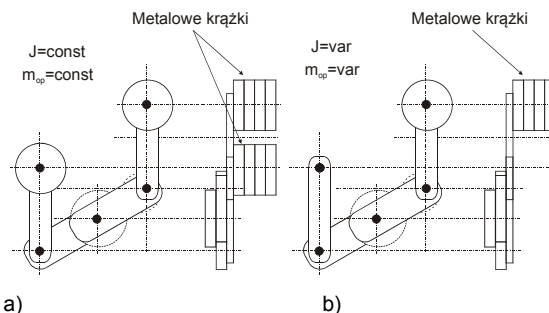
$$(5) \quad m_d = J \frac{d\omega}{dt} + \frac{\omega^2}{2} \frac{dJ}{d\theta}$$

Rozwiązanie równania 5 przedstawiono na rysunku 2 [22].

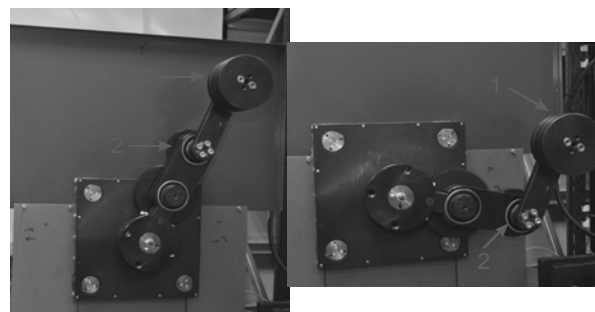


Rys.2. Schemat blokowy przedstawiający układ ze zmiennym momentem bezwładności zależnym od kąta obrotu wału

W ramach realizowanych prac, został zbudowany model mechaniczny obciążenia, którego zadaniem była ciągła zmiana momentu bezwładności i momentu obciążenia w funkcji zmian położenia wału silnika. Na rysunku 3a zaprezentowano schemat poglądowy modelu obciążenia ze stałym momentem bezwładności dla przypadku, w którym założone są metalowe krążki na obu ramionach. W celu uzyskania zmiennego momentu bezwładności w funkcji kąta należy zdemontować metalowe krążki z jednego ramienia. To powoduje, że zmienia się również moment oporowy, który zależy od kąta położenia wału. Taką sytuację ukazuje rysunek 3b.



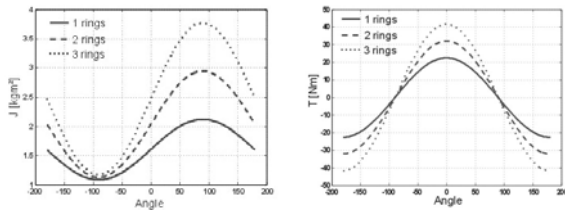
Rys.3. Schemat poglądowy przedstawiający model obciążenia: a) ze stałym momentem bezwładności i obciążenia; b) ze zmiennym momentem bezwładności i obciążenia (fotografia rys. 5)



Rys.4. Fotografia przedstawiająca front stanowiska do ciągłej zmiany momentu bezwładności i momentu obciążenia. 1- metalowe krążki do zmiany masy; 2- ramię o zmiennej długości

Na rysunku 5 przedstawiono zmiany momentu bezwładności oraz obciążenia w funkcji kąta obrotu wału. Minimalna wartość to 1,2 kgm² dla -90° (krążki na dole zainstalowane na jednym ramieniu), maksymalna wartość

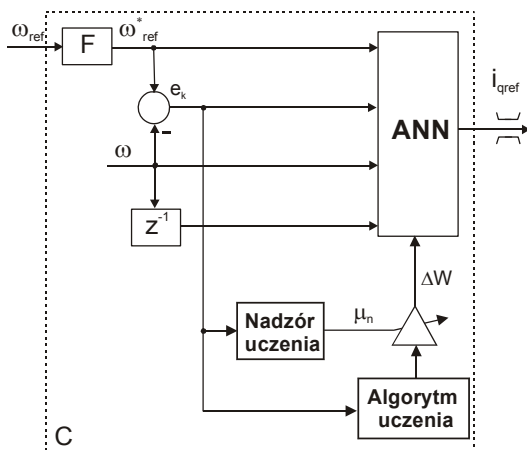
to 3,8 kgm² dla 90⁰ (krążki na górze). Z rys. 5 można odczytać również, że największy moment oporowy osiąga wartość 42Nm dla 0⁰ i -42Nm dla +/-180⁰.



Rys. 5. Przebieg momentu bezwładności i obciążenia dla różnych wartości zainstalowanych krążków

Koncepcja regulatora neuronowego

Uczenie sieci neuronowej regulatora adaptacyjnego odbywa się w trybie *online*, a więc w czasie rzeczywistym. Z tego powodu struktura sieci neuronowej regulatora powinna być możliwie prosta, a algorytm uczenia nie powinien wymagać złożonych działań matematycznych. Na podstawie licznych testów symulacyjnych i laboratoryjnych przyjęto prostą strukturę SSN o jednym neuronie wyjściowym i trzech neuronach w warstwie ukrytej pokazanej na rys. 6. Jako sygnały wejściowe przyjęto uchyb regulacji prędkości e , sygnał prędkości zadanej ω_{ref} , sygnał prędkości mierzonej ω i jej pochodną w formie różnicy wstecznej. Dodatkowo regulator został uzupełniony w układ nadzoru uczenia, aby nie doprowadzić do przeuczenia SSN.



Rys. 6. Struktura regulatora neuronowego

Algorytm RPROP

Najbardziej popularną metodą uczenia SSN jest metoda wstecznej propagacji błędów. Metoda ta zaliczana do metod uczenia z nauczycielem, wymaga jednak znajomości zestawu zbioru uczących, a sam algorytm wykorzystuje gradient funkcji celu, obliczany w kolejnych krokach. Obliczanie w trybie *online* wymagało przyjęcia uczenia sieci bez nauczyciela. Kierując się prostotą metody uczenia przyjęto jako funkcję celu błąd uchybu regulacji prędkości [7,8]:

$$(5) \quad E(k) = \frac{1}{2} (\omega_{ref}(k) - \omega(k))^2 = \frac{1}{2} e^2(k)$$

W celu uproszczenia obliczeń prowadzonych w czasie rzeczywistym został wybrany do nauki sieci prosty algorytm RPROP (*ang. Resilient backPROPagation*) [19]. Podstawową zaletą tego algorytmu jest wyznaczenie jedynie znaku gradientu funkcji celu zamiast pełnej jego wartości. Zmiana wartości wag sieci neuronowej algorytmu RPROP [19] opisana jest następującym wyrażeniem:

$$(6) \quad \Delta w_{ij}(k) = -\eta_{ij}(k) \cdot \text{sign} \left(\frac{\partial E}{\partial w_{ij}}(k) \right) = -\eta_{ij}(k) \cdot \text{sign}(S_{ij}(k))$$

gdzie przez $S_{ij}(k)$ oznaczono gradient funkcji błędów.

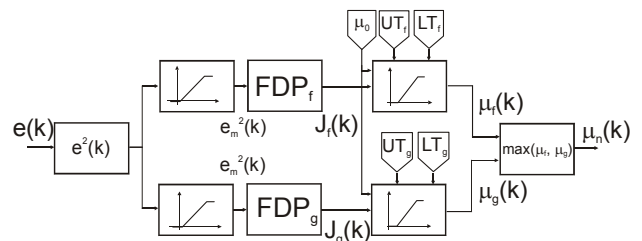
Charakterystyczne dla algorytmu RPROP jest indywidualne wyznaczanie stałej uczenia $\eta_{ij}(k)$ dla każdej wagi i modyfikowanie jej wartości w każdym kroku zgodnie ze wzorem:

$$(7) \quad \eta_{ij}(k) = \begin{cases} \min(a \cdot \eta_{ij}(k-1), \eta_{\max}) & \text{if } S_{ij}(k) \cdot S_{ij}(k-1) > 0 \\ \max(b \cdot \eta_{ij}(k-1), \eta_{\min}) & \text{if } S_{ij}(k) \cdot S_{ij}(k-1) < 0 \\ \eta_{ij}(k-1) & \text{w innych przypadkach} \end{cases}$$

gdzie: $0 < b < 1 < a$, a -współczynnik wzrostu, b -współczynnik zmniejszania aktualnej wartości stałej uczenia.

Przeprowadzone i opisanie wcześniej badania na stanowisku laboratoryjnym [20, 30] opisanego regulatora neuronowego, wykazały potrzebę wprowadzenia modyfikacji algorytmu RPROP, która gwarantowała w warunkach rzeczywistych sygnałów mierzonych sprzężenia zwrotnego, w obecności zakłóceń, stabilny proces uczenia *online*, który nie doprowadzał do przetrenowania SSN. W niniejszej pracy, zaproponowano alternatywny i równie skuteczny sposób rozwiązania problemu przeuczenia sieci neuronowej w porównaniu z [20, 30], przy zachowaniu zalet algorytmu RPROP. Główny moduł uczący został uzupełniony o zewnętrzny układ nadzorujący proces nauki w czasie rzeczywistym (rys. 6). Wielkością wejściową jest aktualna wartość uchybu $e(k)$, a wielkością wyjściową układu nadzorującego jest aktualna wartość współczynnika uczenia μ_n . Wartość współczynnika $\mu_n = 0$ powoduje zatrzymanie adaptacji. Jeżeli wykorzystywany jest algorytm RPROP, sygnał μ_n jest interpretowany jako wartość logiczna, który można zapisać:

$$(8) \quad \Delta w_{ij}(k) = \begin{cases} \text{adaptacja wstrzymana} & \text{dla } \mu_n = 0 \\ -\mu_{ij}(k) \cdot \text{Sign} \left(\frac{\partial E}{\partial w} \right) & \text{dla } \mu_n \neq 0 \end{cases}$$

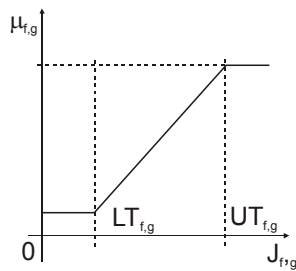


Rys. 7. Struktura układu nadzorującego uczenie

Schemat układu nadzorującego został przedstawiony na rysunku 7. Sygnał wejściowy aktualnej wartości uchybu $e(k)$ zostaje podniesiony do kwadratu $e^2(k)$, następnie wartość ta zostaje ograniczona do wartości dodatniej $e_{max}(k)$. Liczne badania symulacyjne i eksperymentalne wykazały potrzebę wprowadzenia filtra dolnoprzepustowego, po to, aby wytłumić gwałtowne zmiany współczynnika uczenia podczas odpowiedzi na skok wartości zadanej i momentu oporowego. Ponadto, zostały wprowadzone dwie linie o dwóch różnych stałych czasowych T_f i T_g . Jedna linia o krótszej stałej czasowej poprawia pracę napędu podczas procesów dynamicznych, druga, dłuższa stała czasowa minimalizuje uchyb regulacji w stanach statycznych. Wartość wskaźnika $J_{f,g}(k)$ obliczana jest na podstawie równania:

$$(9) \quad J_{f,g}(k) = e_m^2(k) \cdot \frac{T_s}{T_s + T} - \left(\frac{T_s}{T_s + T} - 1 \right) \cdot J_{f,g}(k-1)$$

gdzie: $e_m^2(k)$ – aktualna wartość uchybu za blokiem ograniczenia, T_s -okres próbkowania układu, $T_{f,g}$ -dobrana stała czasowa.



Rys.8. Struktura wewnętrzna bloku do wyznaczania współczynnika uczenia $\mu_{f,g}$, ($LT_{f,g}$ – dolna granica obliczania współczynnika uczenia, $UT_{f,g}$ – górna granica obliczania współczynnika uczenia)

Signal wartości wskaźnika $J_{f,g}$ jest wykorzystywany do wyznaczenia wyjściowego współczynnika uczenia $\mu_{f,g}$ w kolejnym bloku, przedstawionym na rys. 8. Ostateczna wartość współczynnika μ_n wyznaczana jest na podstawie równania:

$$(10) \quad \mu_n = \max(\mu_f, \mu_g).$$

Parametry układu nadzorującego proces uczenia, dobierane są metodą prób i błędów.

Algorytm wstecznej propagacji BP

Algorytm wstecznej propagacji, należący do gradientowych metod optymalizacji, pozostaje nadal jedną z najpopularniejszych metod uczenia wielowarstwowych sieci neuronowych. Modyfikacja wag przebiega zgodnie z formułą [26]:

$$(11) \quad \Delta w_{ij}(k) = -\eta^*_{ij}(k) \cdot \left(\frac{\partial E}{\partial w_{ij}}(k) \right) = -\eta^*_{ij}(k) \cdot (S_{ij}(k))$$

gdzie η^* - współczynnik uczenia.

Momentowy algorytm wstecznej propagacji MBP

Podstawowy algorytm wstecznej propagacji błędów okazuje się zbyt wolny, w przypadku wyboru małego współczynnika uczenia, natomiast zbyt duża wartość prowadzi do problemów ze stabilnością. W literaturze znane są różne metody poprawiające szybkość uczenia sieci neuronowych. Jedną z nich jest wprowadzenie w procesie uaktualniania wag pewnej bezwładności tzw. momentu, proporcjonalnego do zmiany tej wagi z poprzedniej iteracji. Modyfikacja wag przebiega zgodnie z formułą [27]:

$$(12) \quad \Delta w_{ij}(k) = -\eta^*_{ij}(k) \cdot \left(\frac{\partial E(w(k))}{\partial w_{ij}} \right) + \alpha(w(k) - w(k-1))$$

gdzie α - współczynnik momentu.

Algorytm Quickprop

W celu przyspieszenia metody uczenia wprowadzano modyfikacje algorytmu wstecznej propagacji. Jedną z nich jest zastosowanie zmiennego tempa uczenia. Do tych metod oprócz algorytmu *RPROP* zalicza się *Quickprop* [28]. Algorytm modyfikacji wag przebiega zgodnie z formułą:

$$(13) \quad \Delta w_{ij}(k) = -\mu \cdot \left(\frac{\partial E}{\partial w_{ij}}(k) + \gamma w_{ij}(k) \right) + \alpha_{ij}(k) \cdot \Delta w_{ij}(k-1)$$

gdzie: μ - współczynnik uczenia, α - współczynnik momentu γ - dodatkowy współczynnik zmniejszający wagi.

Współczynnik momentu uaktualnia się podczas procesu uczenia zgodnie z wzorem [28]:

$$(14) \quad \alpha_{ij}(k) = \begin{cases} \alpha_{\max} & \text{dla } \beta_{ij}(k) > \alpha_{\max} \\ \text{lub } S_{ij}(n) \cdot \Delta w_{ij}(k) \cdot \beta_{ij}(k) < 0 \\ w \text{ innych przypadkach } \beta_{ij}(k) \end{cases}$$

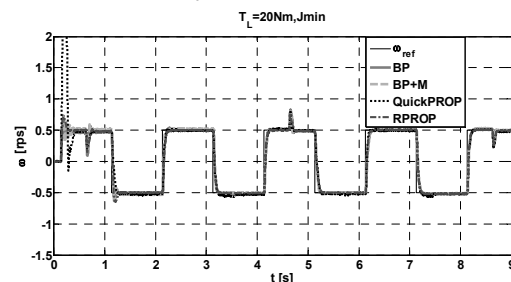
gdzie: α_{\max} – największa wartość współczynnika momentu.

Współczynnik $\beta_{ij}(k)$ wyznaczany jest według zależności:

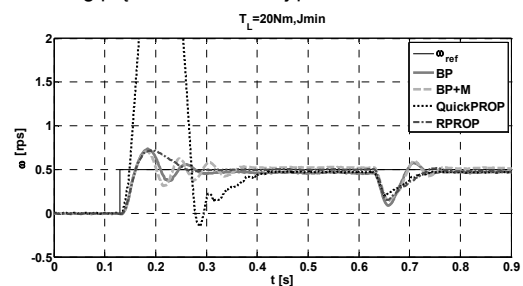
$$(15) \quad \beta_{ij}(k) = \frac{S_{ij}(k)}{S_{ij}(k-1) - S_{ij}(k)}$$

Badania eksperymentalne

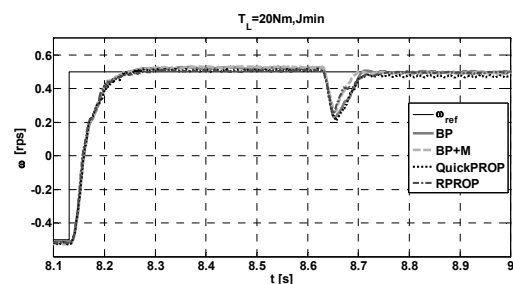
Wszystkie wymienione algorytmy zostały zaimplementowane na procesorze sygnałowym *ADSP-21060* w środowisku *VisualDSP*. Czas próbkowania układu mikroprocesorowego wynosił 100 μ s, a częstotliwość pracy falownika *PWM* wynosiła odpowiednio 10 kHz. Stała czasowa filtra dolnoprzepustowego *F* wynosiła 30 ms. Parametry napędu zostały przedstawione w załączniku. W pracy zaprezentowano porównanie wybranych algorytmów uczenia regulatora neuronowego dla dwóch skrajnych momentów bezwładności $J_{\min} = 1.0 \text{ kgm}^2$ i $J_{\max} = 3.8 \text{ kgm}^2$, których przebiegi pokazano na rys. 9÷13. Badania ze zmiennym momentem bezwładności w funkcji kąta położenia wału zostały przedstawione w pracach [25,29].



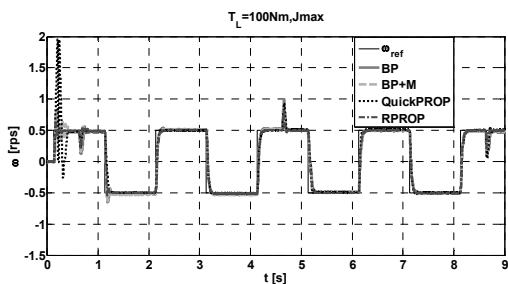
Rys. 9. Przebieg prędkości obrotowej podczas uczenia dla J_{\min}



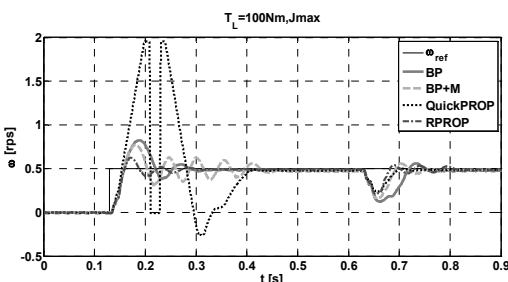
Rys.10. Przebieg prędkości obrotowej na początku rozruchu podczas uczenia dla J_{\min}



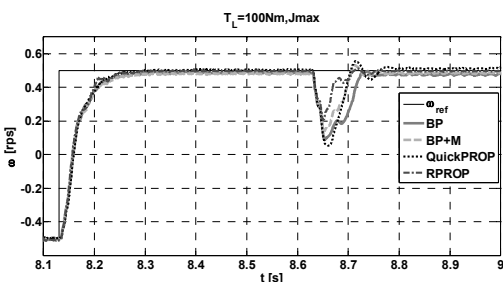
Rys.11. Przebieg prędkości obrotowej po nauczaniu dla J_{\min}



Rys. 12. Przebieg prędkości obrotowej podczas uczenia dla J_{max}



Rys. 13. Przebieg prędkości obrotowej na początku rozruchu podczas uczenia dla J_{max}



Rys. 14. Przebieg prędkości obrotowej po nauczeniu dla J_{max} .

W Tab.1 i Tab. 2 przedstawiono odpowiednio wybrane wskaźniki jakości regulacji dla badanych algorytmów i skrajnych momentów bezwładności. Na rys. 9 i 12 przedstawiono odpowiedź prędkości obrotowej na skok prędkości zdanej ± 0.5 obr/s podczas procesu uczenia układu napędowego, dla skrajnych parametrów mechanicznych napędu.

Tabela 1. Wskaźniki jakości regulacji dla J_{min} po skoku momentu oporowego

J_{min}	t_{odp} [ms]	Uchyb dynamiczny $\Delta\omega$ (rps)	IAE (rev)	ITAE (rev·s)
BP	54	0,277	0,0209	0,00208
BPM	54	0,251	0,0125	0,00162
QuickPROP	56	0,288	0,0214	0,00249
RPROP	55	0,253	0,0151	0,00148

Tabela 2. Wskaźniki jakości regulacji dla J_{max} po skoku momentu oporowego

J_{min}	t_{odp} [ms]	Uchyb dynamiczny $\Delta\omega$ (rps)	IAE (rev)	ITAE (rev·s)
BP	71	0,410	0,0230	0,00267
BPM	62	0,342	0,0188	0,00228
QuickPROP	61	0,434	0,0198	0,00247
RPROP	63	0,283	0,0126	0,00157

W czasie 0.63s i 8.63s został załączony, a w czasie 4.63s wyłączony skokowo moment oporowy, odpowiednio o wartości 20Nm dla minimalnego momentu bezwładności J_{min} i 100Nm dla maksymalnego momentu bezwładności J_{max} . Prąd w osi q podczas procesu przejściowego, oraz

podczas skokowego załączenia obciążenia nie był w ograniczeniu. Problem ograniczenia sygnału wyjściowego neuronowego regulatora prędkości (prąd zadany) został przedstawiony w pracy [20]. Rys. 10 i 13 przedstawia fragment prędkości obrotowej podczas rozruchu. Wagi początkowe miały taką samą wartość dla każdego testowanego algorytmu podczas rozruchu. Można zauważyć znaczne przeregulowanie dla metody *QuickPROP*, oraz oscylacje w sygnale prędkości dla metody *BPM* w porównaniu z pozostałymi testowanymi algorytmami w tej fazie procesu. Zachowanie pozostałych metod należy uznać za poprawne. Na rys. 11 i 14 przedstawiono porównanie odpowiedzi prędkości obrotowej na skok prędkości zdanej ± 0.5 obr/s dla testowanych algorytmów, gdzie w czasie 8.63s zostało załączone skokowo obciążenie. W odpowiedzi na skok momentu oporowego pojawiły się oscylacje, będące wynikiem wielomasowej struktury obiektu sterowanego [23, 30]. W Tab. 1 i 2 przedstawiono wskaźniki jakości po skoku momentu oporowego, takie jak: czas odpowiedzi liczony do 90% prędkości referencyjnej, uchyb dynamiczny zdefiniowany jako maksymalny błąd prędkości po skoku obciążenia i kryteria całkowite *IAE* oraz *ITAE*. Jeśli chodzi o czasy odpowiedzi, to są one podobne, uchyb dynamiczny dla J_{min} jest na podobnym poziomie (przy skokowym momencie oporowym $T_L = 20Nm$), natomiast dla J_{max} ($T_L = 100Nm$) można zauważyć znaczące różnice na korzyść algorytmu *RPROP*. Porównując kryteria całkowite *IAE* i *ITAE*, należy wyróżnić dwie metody *BPM* i *RPROP*, ponieważ charakteryzują się najmniejszymi wskaźnikami w porównaniu z *BP* i *QuickPROP*.

Podsumowanie

W pracy przedstawiono wybrane właściwości adaptacyjnego układu sterowania prędkością obrotową na zmiany momentu bezwładności z neuronowym regulatorem prędkości, którego parametry zmieniane są online według szybkiego algorytmu *RPROP* z zaproponowanym modulem nadzoru uczenia *SSN*. Efektem końcowym było porównanie opisanego algorytmu z innymi wybranymi algorytmami trenowania sieci neuronowych, takich jak: algorytm wstecznej propagacji *BP* (ang. *BackPropagation*), momentowa metoda wstecznej propagacji błędów *MBP* (ang. *Momentum BackPropagation*), *Quickprop*. Należy podkreślić poprawne działanie każdego z przetestowanych algorytmów. Wyraźne różnice można zaobserwować podczas procesów przejściowych, zwłaszcza na początku rozruchu oraz podczas załączenia momentu oporowego. Należy zwrócić uwagę, na dobre zachowanie się algorytmu *RPROP*.

Załącznik. Dane układu napędowego

Parametry	Jednostka	Wartość
Moment bezwładności silnika	kg·m ²	0,046
Minimalny moment bezwładności	kg·m ²	1,2
Maksymalny moment bezwładności	kg·m ²	3,8
Stała momentu	Nm/A	17,5
Stała napięciowa	V/1000rpm	1560
Moment znamionowy	Nm	50
Prędkość znamionowa	rpm	145
Indukcyjność	mH	45
Rezystancja	Ω	18,5
Napięcie znamionowe	V	310
Prąd znamionowy	A	1,94

Praca finansowana ze środków MNiSzW, w ramach dotacji statutowej Politechniki Poznańskiej nr 04/45/DSPB/0121.

Autorzy: dr inż. Tomasz Pajchrowski, Politechnika Poznańska, Instytut Automatyki i Inżynierii Informatycznej, ul. Piotrowo 3a, 60-965 Poznań, E-mail: tomasz.pajchrowski@put.poznan.pl; inż. Wojciech Kogut, Instytut Automatyki i Inżynierii Informatycznej, ul. Piotrowo 3a, 60-965 Poznań, E-mail: wojciech.kogut@student.put.poznan.pl.

LITERATURA

- [1] Asada H, Youcef-Toumi K., Direct-Drive Robots: Theory and Practice, *The MIT Press*, 1977
- [2] Brian R. Carlisle, Carl R. Witham, Donald R. Allan, John W. Meadows, Direct drive robotic systems, patent USA nr 5314293, 1994.
- [3] Ikaheimo J., Permanent magnet motors eliminate gearboxes, *ABB Review* 4 (2002), 22-25
- [4] Younkin G.W., Industrial servo control systems: fundamentals and applications, *CRC* 2003.
- [5] Chen T.C., Sheu T.T., Model reference robust speed control for induction-motor drive with time delay based on neural network, *IEEE Trans. Systems, Man and Cybernetics*, 31 (2001), n.6, 746-753
- [6] Li. W., Cheng X., Adaptive high-precision control of positioning tables theory and experiments, 1994; 2(3):265-70
- [7] Bose B.K., Neural Network Applications in Power Electronics and Motor Drives—An Introduction and Perspective, *IEEE Transactions on Industrial Electronics*, 54 (2007), n.1, 14-33
- [8] Bose B.K., Chio K.M., Kim H.J., Self Tunning Neural Network Controller for Induction Motor Drives, *IEEE Annual Conf. of the IEEE Industrial Electronics Society*, 2002, Vol.1, 152+156.
- [9] Kazmierkowski M. P., Orłowska-Kowalska T., NN state estimation and control in converter-fed induction motor drives, Chapter 2 in a book: *Soft Computing in Industrial Electronics*, Physica-Verlag, 2002, Heilderberg, Germany.
- [10] Kaminski M., Orłowska-Kowalska T., FPGA Implementation of ADALINE-based Speed Controller for Two-mass System, *IEEE Trans. on Industrial Informatics*, 9 (2013), n.3, 1301-1311
- [11] Grzesiak L., Meganck V., Sobolewski J., Ufnalski B., DTC_SVM Drive with ANN-based Speed Controller, *PELINCEC Conference*, Warsaw University of Technology, CD, (2005)
- [12] Grzesiak L.M., Meganck V., Sobolewski J., Ufnalski B., On-line Trained Neural Speed Controller with Variable Weight Update Period for Direct-Torque-Controller AC Drive, *EPE-PEMC Conference*, Portoroz, CD, (2006)
- [13] Orłowska-Kowalska T., Szabat K., Control of the Drive System with Stiff and Elastic Coupling using Adaptive Neuro-Fuzzy Approach, *IEEE Trans. On Industrial Electronics*, 51 (2007), n.4, 228-240
- [14] Rahman M.A., Hoque M.A., On-line adaptive artificial neural network based vector control of permanent magnet synchronous motors, *IEEE Transactions on Energy Conversion*, 13 (1988), n.4, 311-318
- [15] Chen D., York M., Adaptive Neural Inverse Control Applied to Power Systems, *IEEE Power Systems Conf. and Exposition*, 2006, 2109-2115.
- [16] COLINA-MORLES E., MORT N., Inverse Model Neural Network - Based Control of Dynamic Systems, *IEE Control Conf. Publ.*, 1994, Vol. 2, 955- 960.
- [17] Chen T.C, Sheu T.T., Model Reference Neural Network Controller for Induction Motor Speed Control, *IEEE Trans. Energy Conversion*, 17 (2002), n.2, 157-162
- [18] Kamiński M., Orłowska-Kowalska T., Regulator neuronowe trenowane off-line zastosowane w strukturze sterowania prędkością napędu dwumasowego, *Przegląd Elektrotechniczny*, 91 (2015), nr 1, 9-12
- [19] Riedmiller M., Braun H., A direct adaptive method for faster backpropagation learning: The RPROP algorithm, *IEEE International Conference on Neural Networks*, 28 March – 1 April, (1993), vol. 1, 586-591
- [20] Pajchrowski T, Zawirski K, Nowopolski K., A Neural Speed Controller Trained On-Line by Means of Modified RPROP Algorithm, *IEEE Trans. Ind. Inform.*, 11 (2015), n.2, 560-568
- [21] Leonard W., Control of electrical drives, *Berlin, Heiderberg, Springer* (2001)
- [22] Zawirski K., Deskur J., Kaczmarek T., *Automatyka Napędu Elektrycznego*, Wydawnictwo Politechniki Poznańskiej, Poznań 2012
- [23] Deskur J., Pajchrowski T., Zawirski K.: Speed Controller for a Drive With Complex Mechanical Structure And Variable Parameters, *Proceedings of 16th International Power Electronics and Motion Control Conference and Exposition, PEMC'2014*, 21-24 September 2014, Antalya/Turkey.
- [24] Gong S., Ding X., He X., Ren H., Research of PMSM controller based on 2DOF-PID algorithm, *International Conference on Electrical Machines and Systems, ICEMS*, (2007), 1116-1119,.
- [25] Pajchrowski T., Application of an Internal Model Speed Control for PMSM with variable mechanical parameters, *Proceedings of 2015 IEE 2nd International Conference on Cybernetics CYBCONF*, Gdynia, Poland, 24-26 June 2015, paper on pen drive, ISBN 978-1-4799-8321-6
- [26] Rumelhart D. E., Hinton G. E., Williams J.: Learning representations by backpropagating errors, *Nature*, 323 (1986), n.9, 533-536,
- [27] Rutkowski L., *Computational Intelligence Methods and Techniques*, Springer-Verlag, 2008
- [28] Scott E. Fahlman, An empirical study of study of learning speed in backpropagation networks, *Technical report CMU-CS-88-162*, School of Computer Science, Carnegie Mellon University, Pittsburg, PA 15213, September 1988
- [29] Pajchrowski T., The direct drive with variable moment of inertia in the structure of the reference model, *16th International Power Electronics and Motion Control Conference and Exposition, PEMC, 21-24 Sept. 2014 Antalya, Turkey*, 831-836
- [30] Pajchrowski T., Zawirski K., Nowopolski K., Application of Adaptive Neural Controller for Drive with Elastic Shaft and Variable Moment of Inertia, *17th Conference on Power Electronics and Applications, EPE'15-ECCE Europe*, Geneva, Switzerland, 8 -10, September, 2015, ISBN: 9789075815238 and CFP15850-USB.