

Sterowanie adaptacyjne silnikiem liniowym z neuronowym kompensatorem tarcia strojonym on-line

Streszczenie. W artykule przedstawiono sterowanie silnikiem liniowym z magnesami trwałymi w warunkach zmieniającego się tarcia. Do sterowania wykorzystano algorytm wstępnego całkowania z neuronowym kompensatorem tarcia. Do uczenia on-line modelu neuronowego wykorzystano wyznaczaną przez różniczkowe prawa adaptacji poprawkę dla modelowanego tarcia. Symulacje i eksperymenty przeprowadzone na stanowisku rzeczywistym potwierdziły skuteczność pokazanego rozwiązania.

Abstract. This paper presents a control of linear motor with permanent magnets in various friction conditions. The control algorithm uses backstepping algorithm with neural friction compensator. Correction of modelled friction, calculated by differential adaptive law was used for on-line learning of neural model. Simulations and experiments carried out on a real drive, confirmed the efficiency of the solution. (**Adaptive control of linear motor with online learning of neural friction compensator**).

Słowa kluczowe: silnik liniowy, kompensacja tarcia, sieci neuronowe, sterowanie adaptacyjne

Keywords: linear motor, friction compensation, neural networks, adaptive control

oi:10.12915/pe.2014.06.12

Wstęp

Tarcie może być zjawiskiem pożądanym, jak np. w hamulcach, ale z reguły jest ono przeszkodą tak jak w serwonapędach. Tarcie może prowadzić do zwiększenia błędów śledzenia wartości zadanej, cykli granicznych, a nawet niestabilności układu regulacji. Szczególnie widoczne jest to w układach charakteryzujących się niewielkimi przemieszczeniami, małymi prędkościami i pracą nawrotną. W takich przypadkach modelowanie i kompensacja tarcia są zagadnieniami ważnymi. Uzasadnieniem tego jest prowadzenie wielu prac badawczych z zakresu identyfikacji i kompensacji tarcia [1]. Niektóre z opisywanych metod bazują na znajomości dokładnych modeli nieliniowych tarcia, podczas gdy inne traktują tarcie jako część zakłóceń działających na układ [2].

Siła tarcia jest najczęściej modelowana jako nieliniowa funkcja prędkości [3]. Ze względu na złożoność zjawisk statycznych i dynamicznych przygotowanie dokładnego modelu tarcia jest pracochłonne i trudne, szczególnie gdy dokładność tych modeli zależy od czynników zewnętrznych takich jak temperatura, stopień smarowania, zużycie materiału czy stopień zabrudzenia. Dokładność układu sterowania serwomechanizmem może się więc pogorszyć wraz ze zmianą czynników wpływających na tarcie. Kolejnym problemem podczas modelowania tarcia jest jego niepewność. W związku z powyższym zasadnym jest zastosowanie podejścia opartego na wykorzystaniu nieliniowego modelu tarcia wraz z mechanizmem jego adaptacji do zmieniających się warunków [4,5,6].

Sztuczne sieci neuronowe są w stanie "nauczyć się" aproksymować dowolną funkcję wielu zmiennych. Ich niewątpliwym atutem jest możliwość adaptacji do nowych warunków. Z tego względu w pracy zdecydowano się zadanie identyfikacji i kompensacji tarcia powierzyć sieci neuronowej. Chcąc umieścić w sieci sformalizowaną w postaci wzorów wiedzę na temat tarcia, zdecydowano się wykorzystać sieć neuronową z rozszerzeniem funkcyjnym Pao [7].

Obiekt sterowania

Będziemy sterować położeniem silnika liniowego opisanego równaniami

$$(1) \quad \dot{x} = v$$

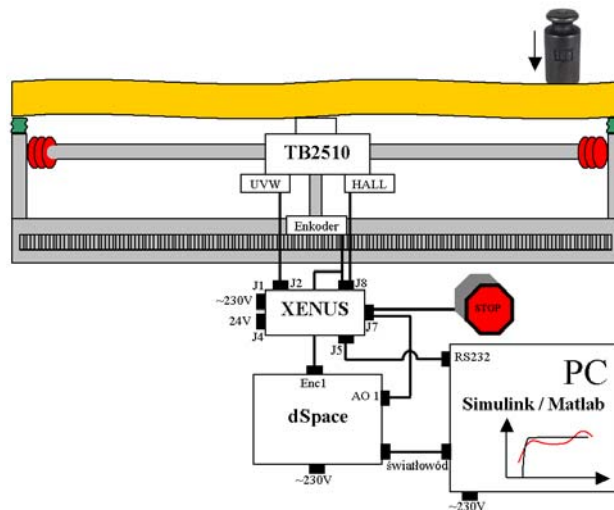
$$(2) \quad \dot{v} = \frac{F}{m} - \frac{T}{m}$$

gdzie: x – położenie silnika, v – jego prędkość, m – masa silnika (razem z czujnikami, przewodami i wózkiem), F – siła generowana przez twornik (sterowanie), T – suma

wszystkich sił zewnętrznych, na którą składa się głównie tarcie, ale także sprężystość przewodów, grawitacja itp.



Rys.1. Zdjęcie stanowiska laboratoryjnego



Rys.2. Schemat stanowiska laboratoryjnego

Stanowisko laboratoryjne

Skuteczność zaproponowanej metody sprawdzono w rzeczywistym układzie regulacji położenia (rys.1). Schemat stanowiska został przedstawiony na rysunku 2. Układ składał się z :

- silnika liniowego TrustTube TB2510 (rys.1) o wadze 7,04kg i maksymalnej sile 104N, z enkoderem liniowym o rozdzielczości 1µm,

- falownika Xenus XTL-230-18 z wbudowanym regulatorem prądu,
- aparatury kontrolno-pomiarowej opartej na karcie procesorowej dSpace DS1006,
- metalowej szyny umożliwiającej zmianę tarcia na drodze przejazdu silnika.

Wstępna identyfikacja oporów ruchu

Wstępną identyfikację całkowitego oporu ruchu silnika (nazywanego dalej po prostu tarcie) przeprowadzono dla przypadku silnika nieobciążonego. W prostym eksperymencie, korzystając z równania dynamiki silnika

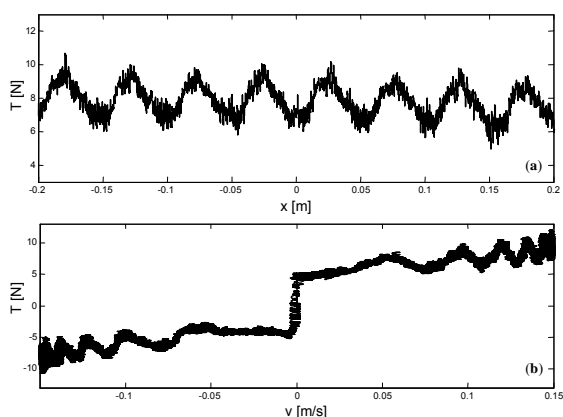
$$(3) \quad T = F - m\ddot{x}$$

została wyznaczona wartość siły tarcia podczas przejazdu ze stałą prędkością i podczas pracy nawrotnej. Zależności wyznaczonego tarcia od położenia i prędkości silnika zostały pokazane na rysunku 3.

Analiza otrzymanych przebiegów pokazuje

- okresową zmienność siły tarcia wraz ze zmianą położenia silnika, przy czym okres tych zmian wynosi $d=51,2\text{mm}$ i jest równy podziałce biegunowej silnika,
- nieciągłość tarcia dla prędkości $v=0$, która wynika z istnienia tarcia statycznego.

Obserwacje te zostaną wykorzystane podczas doboru struktury sieci neuronowej pełniącej rolę kompensatora tarcia.



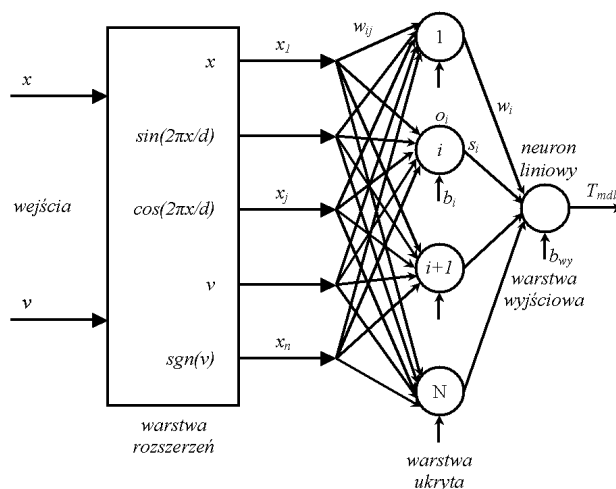
Rys.3. Zależność siły tarcia od: (a) położenia silnika przy stałej prędkości przejazdu, (b) prędkości silnika

Sieć neuronowa z rozszerzeniem funkcyjnym Pao

W trakcie procesu tworzenia modeli należy wziąć pod uwagę dwa kryteria tj. ich dokładność i złożoność. W przypadku modeli rozmytych zwiększenie ich dokładności odbywa się zazwyczaj poprzez zwiększenie liczby reguł. Niestety bardziej skomplikowane modele wymagają większego nakładu obliczeniowego, co może mieć wpływ na czas obliczenia sterowania. W przypadku gdy dysponujemy wiedzą sformalizowaną w postaci wzorów, możliwe jest zawarcie jej w funkcjach następników modelu rozmytego typu TSK [8]. Zastosowanie takiego rozwiązania pozwala na znaczne zwiększenie dokładności modelu przy niewielkim zwiększeniu stopnia skomplikowania modelu, a nawet przy jego redukcji, wynikającej z możliwości zmniejszenia liczby reguł. W przypadku sztucznych sieci neuronowych zwiększenie dokładności modelu odbywa się poprzez zwiększenie liczby warstw i liczby neuronów. Podobnie, jak w przypadku systemów rozmytych, tak samo w przypadku sieci neuronowej możliwe jest wykorzystanie wiedzy danej w postaci wzorów. Korzysta się przy tym z rozszerzenia funkcyjnego Pao.

Uwzględniając obserwacje poczynione podczas próby wstępnego określenia siły tarcia, wprowadzone zostaną

rozszerzenia funkcyjne: $\sin(2\pi x/d)$, $\cos(2\pi x/d)$, $\text{sgn}(v)$, przy czym pierwsze dwa mają za zadanie ułatwić modelowanie składowej zmiennej okresowo siły tarcia (jej fazę i amplitudę), natomiast trzecie ma ułatwić modelowanie tarcia statycznego. Neuronowy kompensator będzie się składał z warstwy rozszerzeń funkcyjnych, warstwy ukrytej i neuronu wyjściowego. Jego struktura została przedstawiona na rysunku 4.



Rys.4. Struktura wykorzystanego kompensatora neuronowego

Siła tarcia przybliżana przez sieć neuronową mającą n wejść i N neuronów w warstwie ukrytej będzie określona wzorem

$$(4) \quad T_{mdl} = \sum_{i=1}^N s_i w_i + b_{wy}$$

gdzie s_i – sygnał wyjściowy neuronu warstwy ukrytej, połączonego wagą w_i z neuronem wyjściowym o polaryzacji b_{wy} dany jest wzorem

$$(5) \quad s_i = \tanh(o_i)$$

w którym o_i jest pobudzeniem i -tego neuronu warstwy ukrytej o polaryzacji b_i , połączonego z j -tym wejściem poprzez wagę w_{ij}

$$(6) \quad o_i = \sum_{j=1}^n x_j w_{ij} + b_i$$

Gdybyśmy dysponowali dokładnym modelem tarcia, jego odpowiedź byłaby równa tarcu rzeczywistemu $T=T_{mdl}$. W rzeczywistości aby otrzymać wartość dokładną tarcia, należałoby ją skorygować za pomocą współczynnika korekcji k , $T=kT_{mdl}$. Estymata tego współczynnika \hat{k} będzie wyznaczana przez adaptacyjny algorytm wstecznego całkowania.

Algorytm sterowania

Oznaczmy błąd śledzenia położenia zadanego x_r jako

$$(7) \quad e = x_r - x$$

W pierwszym kroku zdefiniujemy funkcję Lapunowa jako

$$(8) \quad V = \frac{1}{2} e^2$$

Jej pochodna systemowa będzie równa

$$(9) \quad \dot{V} = e\dot{e} = e(\dot{x}_r - \dot{x}) = e(\dot{x}_r - v)$$

Wprowadźmy wirtualne sterowanie v_r (prędkość zadana silnika) i błąd śledzenia prędkości y

$$(10) \quad y = v_r - v,$$

wtedy

$$(11) \quad \dot{V} = ey + e(\dot{x}_r - v_r).$$

Jeżeli wybierzemy sterowanie określimy jako

$$(12) \quad v_r = k_1 e + \dot{x}_r,$$

gdzie k_1 będzie dodatnim wzmocnieniem to otrzymamy

$$(13) \quad \dot{V} = -k_1 e^2 + ey.$$

W drugim kroku zdefiniujemy funkcję Lapunowa jako

$$(14) \quad V = \frac{1}{2} e^2 + \frac{1}{2} y^2 + \frac{1}{2\gamma} \tilde{k}^2,$$

gdzie

$$(15) \quad \tilde{k} = k - \hat{k}$$

jest błędem estymaty współczynnika korygującego k . Pochodna funkcji Lapunowa po czasie będzie równa

$$(16) \quad \dot{V} = -k_1 e^2 + ey + y\dot{y} + \frac{1}{\gamma} \tilde{k} \dot{\tilde{k}}.$$

Podstawiając kolejno zróżniczkowane równanie (2), następnie (9) i (12) do (13) otrzymamy

$$(17) \quad \begin{aligned} \dot{V} &= -k_1 e^2 + y \left(e + \dot{v}_r - \frac{F}{m} + \frac{kT_{mdl}}{m} \right) + \frac{1}{\gamma} \tilde{k} \dot{\tilde{k}} = \\ &= -k_1 e^2 + y \left(e + \dot{v}_r - \frac{F}{m} + \hat{k} \frac{T_{mdl}}{m} \right) + y \tilde{k} \frac{T_{mdl}}{m} + \frac{1}{\gamma} \tilde{k} \dot{\tilde{k}} \end{aligned}$$

Jeżeli wybierzemy sterowanie postaci

$$(18) \quad F = m(e + \dot{v}_r + k_2 y) + \hat{k} T_{mdl},$$

gdzie k_2 jest dodatnim wzmocnieniem, to pochodna systemowa przyjmie postać

$$(19) \quad \dot{V} = -k_1 e^2 - k_2 y^2 + y \tilde{k} \frac{T_{mdl}}{m} + \frac{1}{\gamma} \tilde{k} \dot{\tilde{k}}.$$

Zakładając stałość parametru k (bądź jego wolnozmiennność), możemy przyjąć że $\dot{\tilde{k}} = -\dot{\hat{k}}$, co sugeruje nam na napisanie praw adaptacji estymaty \hat{k} jako

$$(20) \quad \dot{\hat{k}} = \frac{\gamma}{m} y T_{mdl}.$$

Ostatecznie otrzymamy

$$(21) \quad \dot{V} = -k_1 e^2 - k_2 y^2,$$

co dowodzi stabilności układu sterowania.

Podsumowując, algorytm sterowania będzie określony wzorami (12), (18) i (20). Parametry k_1 i k_2 są wzmocnieniami, γ jest parametrem określającym szybkość adaptacji. Są to parametry wymagające strojenia.

Uczenie modelu on-line

Uczenie modelu neuronowego poprzez zmianę wag (w_{ij} , w_i) i polaryzacji (b_i , b_{wy}) ma na celu minimalizację kwadratu błędu między skorygowaną przez współczynnik \hat{k} estymatą tarcia \hat{T} , a odpowiedzią modelu T_{mdl}

$$(22) \quad E = \frac{1}{2} \sum e_{mdl}^2 = \frac{1}{2} \sum (\hat{T} - T_{mdl})^2 = \frac{1}{2} \sum (\hat{k} T_{mdl} - T_{mdl})^2.$$

Do strojenia parametrów modelu zastosowano metodę największego spadku. Każdy parametr p zmieniał się według formuły

$$(23) \quad p(k+1) = p(k) - \eta \frac{\partial E}{\partial p},$$

gdzie η określa szybkość uczenia. Po uwzględnieniu wzorów (4-6) i (22), pochodne cząstkowe błędu uczenia względem strojonych parametrów zostały wyznaczone jako:

$$(24) \quad \frac{\partial E}{\partial w_i} = \frac{\partial E}{\partial e} \frac{\partial e}{\partial T_{mdl}} \frac{\partial T_{mdl}}{\partial w_i} = -es_i$$

$$(25) \quad \frac{\partial E}{\partial b_{wy}} = \frac{\partial E}{\partial e} \frac{\partial e}{\partial T_{mdl}} \frac{\partial T_{mdl}}{\partial b_{wy}} = -e$$

$$(26) \quad \frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial e} \frac{\partial e}{\partial T_{mdl}} \frac{\partial T_{mdl}}{\partial s_i} \frac{\partial s_i}{\partial o_i} \frac{\partial o_i}{\partial w_{ij}} = -ew_i (1 - \tanh^2(o_i)) x_j$$

$$(27) \quad \frac{\partial E}{\partial b_i} = \frac{\partial E}{\partial e} \frac{\partial e}{\partial T_{mdl}} \frac{\partial T_{mdl}}{\partial s_i} \frac{\partial s_i}{\partial o_i} \frac{\partial o_i}{\partial b_i} = -ew_i (1 - \tanh^2(o_i)).$$

Eksperymenty numeryczne

W oparciu o symulacje komputerowe przeprowadzono dwa eksperymety. Pierwszy z nich miał na celu pokazanie wpływu wyboru struktury sieci na dokładność modelowania tarcia i błąd śledzenia trajektorii zadanej w stanie quasi-ustalonym. Pokazano także jak błędy modelowania i śledzenia zmieniają się w czasie podczas uczenia on-line kompensatora. W eksperymencie drugim zbadana została zdolność adaptacji modelu do nowych warunków wynikających ze zmiany tarcia.

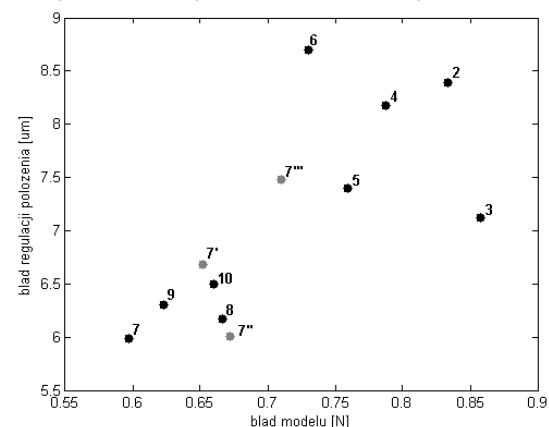
Podczas symulacji korzystano z rozmytego modelu tarcia opisanego w [9], nastrojonego off-line w oparciu o dane zebrane przy pomocy parametrycznego obserwatora OPI [10]. We wszystkich eksperymencie numerycznych wartość zadana położenia zmieniała się według zależności

$$(28) \quad x_z(t) = 0,3 \sin(0,5t).$$

Parametry układu regulacji były równe: $k_1=40$, $k_2=120$, $\gamma=5000$, $\eta=10^{-5}$.

Eksperyment 1

Sprawdzony został wpływ liczby neuronów N w warstwie ukrytej na błąd modelowania tarcia oraz błąd śledzenia trajektorii zadanej. Wartości błędów (dokładnie pierwiastek błędu średniokwadratowego – RMSE z jednego cyklicznego przejazdu) zostały przedstawione na rysunku 5.

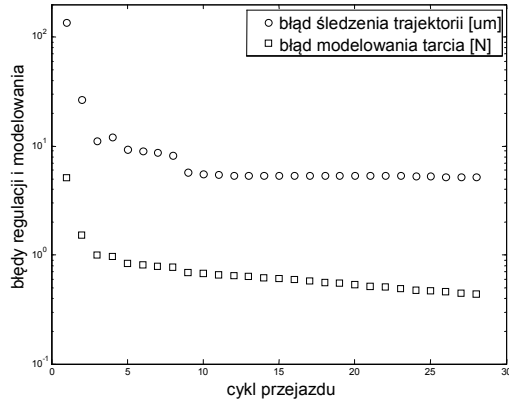


Rys.5. Wartość błędów RMS modelowania i regulacji dla układów sterowania z neuronowym modelem tarcia zawierającym od $N=2$ do 10 neuronów w warstwie ukrytej

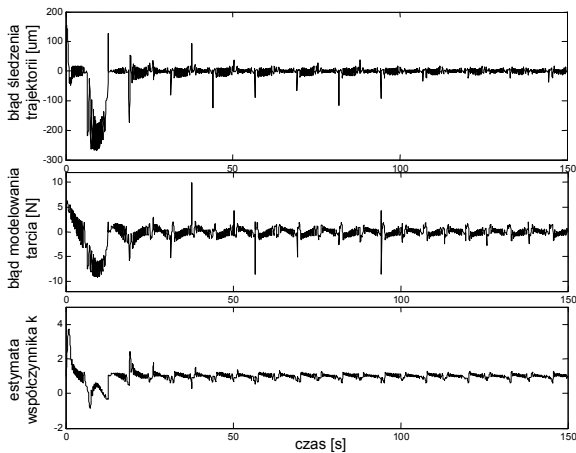
Najmniejszy błąd modelowania tarcia osiągnięto dla struktury zawierającej $N=7$ neuronów w warstwie ukrytej. Dla tak określonej struktury obliczenia zostały powtórzone trzykrotnie, zaczynając za każdym razem od losowo wybranych wartości wag. Analizując wyniki można stwierdzić, że:

- istnieje wpływ wyboru początkowych wartości wag na końcowy wynik strojenia, o czym świadczy rozrzut punktów 7,7',7'',7''', ale nie jest on duży. Błąd modelu mieścił się w zakresie 0,59-0,71N.

- występuje dodatnia korelacja między liczbą neuronów a dokładnością modelu i sterowania.

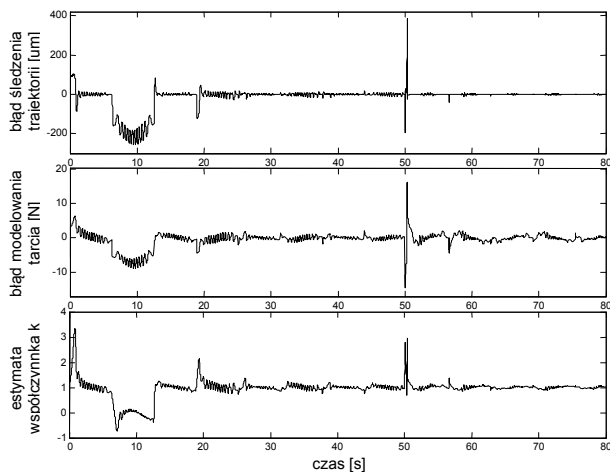


Rys.6. Zmiana wartości błędów RMS modelowania i regulacji dla układów sterowania z neuronowym modelem tarcia zawierającym $N=5$ neuronów w warstwie ukrytej

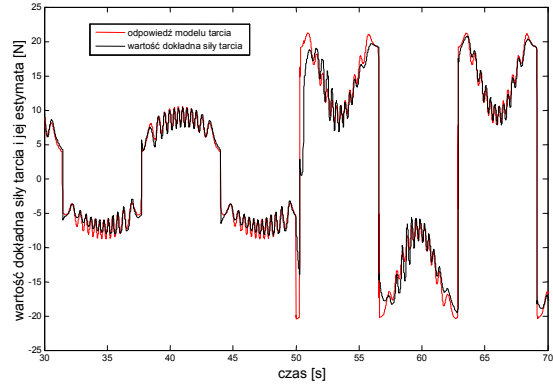


Rys.7. Przebiegi czasowe błędu śledzenia trajektorii, błędu modelu i estymaty współczynnika korygującego

Średnie wartości błędów (RMSE) modelu i regulacji dla kolejnych cykli przejazdów pokazano na rysunku 6. Na rysunku 7 zostały pokazane ich przebiegi czasowe, a także zmiana estymaty współczynnika korygującego model \hat{k} . Zmniejszanie się tych błędów świadczy o poprawnym działaniu sieci neuronowej. Potwierdza to także przebieg współczynnika korygującego w okolicy wartości 1.



Rys.8. Przebiegi błędów regulacji i modelowania oraz współczynnika korekcji podczas zmiany tarcia



Rys.9. Dokładna wartość siły tarcia i jej estymata przed i po zmianie tarcia w chwili $t=50s$

Eksperyment 2

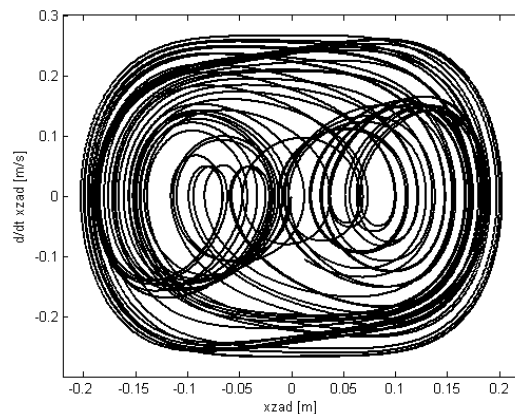
W kolejnej próbie została zbadana zdolność adaptacji modelu do nowych warunków wynikających ze zmiany tarcia. W chwili czasowej $t=50s$, tarcie zostało zwiększone o składnik $40\cos(3x)\text{sgn}(v)$. Zarejestrowane przebiegi błędów i współczynnika korekcji zostały pokazane na rysunku 8. Rysunek 9 przedstawia przebieg dokładnej wartości tarcia i jej nieskorygowanej estymaty wyznaczonej przez model. Podobnie jak w poprzednim eksperymencie zdolność adaptacji sieci została potwierdzona niewielkim błędem regulacji i modelowania oraz dążeniem korekty do jedności.

Eksperymenty na stanowisku rzeczywistym

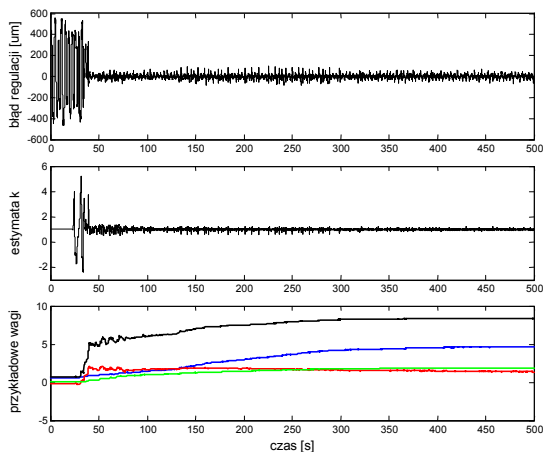
Działanie algorytmu sprawdzono w układzie rzeczywistym. Podobnie jak poprzednio przeprowadzono dwa eksperymety. W pierwszym z nich, dla układu ze stałym tarcie, przeszedzono proces uczenia się sieci neuronowej od zera (wartości losowe wag). W drugim eksperymencie zbadano zdolność adaptacji sieci neuronowej do nowych warunków tarcia.

Chcąc wykazać, że sieć neuronowa potrafi uczyć się, a następnie modelować tarcie nie tylko na trajektorii powtarzalnej (28), ale dla dowolnej trajektorii zadanej wykorzystano przebiegi generowane przez oscylator Duffinga, którego parametry zostały dobrane tak, aby otrzymać układ chaotyczny, co pozwoliło uniknąć powtarzalności przejazdów. Tak określona wartość zadana została wykorzystana m.in. w sterowaniu prędkością mieszadła miksera przemysłowego [11]. Równania określające wartość zadaną otrzymano po przeskalowaniu rozwiązania układu (29). Trajektorię zadaną dla silnika pokazuje rysunek 10.

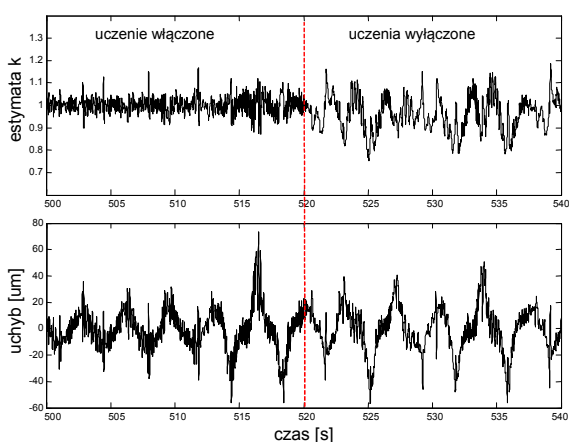
$$(29) \quad \begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -px_2 - p_1x_2 - p_2x_1^2 - q\cos(\lambda t) \end{aligned}$$



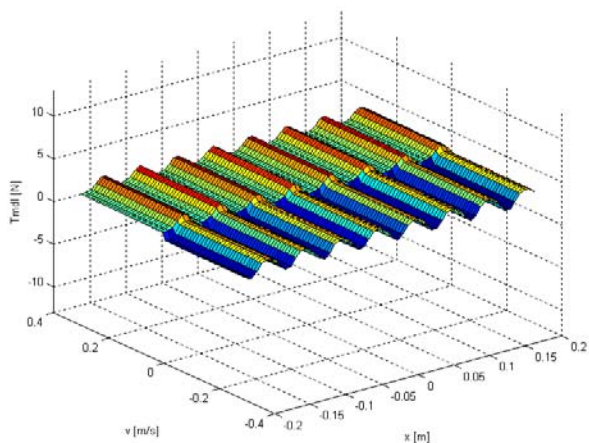
Rys.10. Wykres fazowy oscylatora Duffinga dla parametrów $p=0,45$; $p_1=-1,1$; $p_2=1$; $q=2,1$; $\lambda=1,8$



Rys.11. Przebiegi błędu regulacji, estymaty współczynnika korygującego i przykładowych wag w trakcie uczenia sieci od podstaw



Rys.12. Estymata korekty k i błąd śledzenia trajektorii zadanej po wyłączeniu uczenia



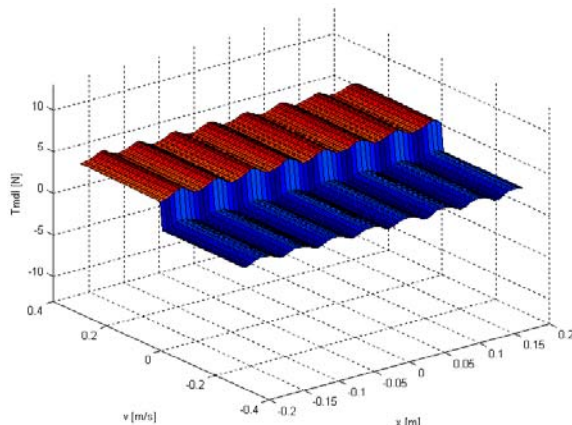
Rys.13. Powierzchnia wygenerowana przez neuronowy model tarcia z losowymi wartościami wag. Czas uczenia $t=0s$

Eksperyment 3

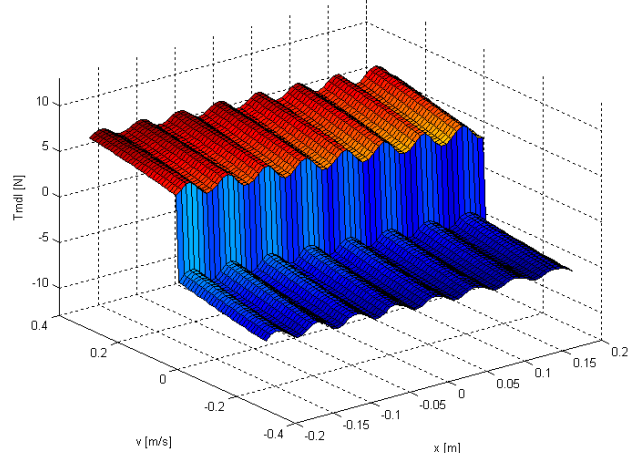
W porównaniu z eksperymentami numerycznymi została zredukowana szybkość adaptacji z wartości $\gamma=5000$ do 2000. Pozostałe parametry, czyli wzmocnienia i szybkość uczenia sieci pozostały bez zmian. Przebiegi uchybu, błędu modelowania i przykładowe wagi dla uczenia sieci od zera zostały pokazane na rysunku 11. W pierwszej fazie eksperymentu zarówno adaptacja jak i uczenie sieci było wyłączone ($\eta=0$, $\gamma=0$, $k=1$). Sytuacji tej towarzyszył błąd regulacji na poziomie $400\mu m$. Po włączeniu adaptacji

(skokowa zmiana z $\gamma=0$ na 2000) estymata współczynnika k zmieniała się od -2 do 5 , co jednak nie wystarczyło aby poprawić jakość regulacji. Dopiero pozwolenie sieci na uczenie się (zmiana z $\eta=0$ na $\eta=10^{-5}$) przyniosło widoczne zmniejszenie błędu śledzenia trajektorii. Uczenie sieci widoczne jest w zmianie wag i skutkuje coraz mniejszą potrzebą korekty odpowiedzi modelu neuronowego (zbliżanie się estymaty k do 1).

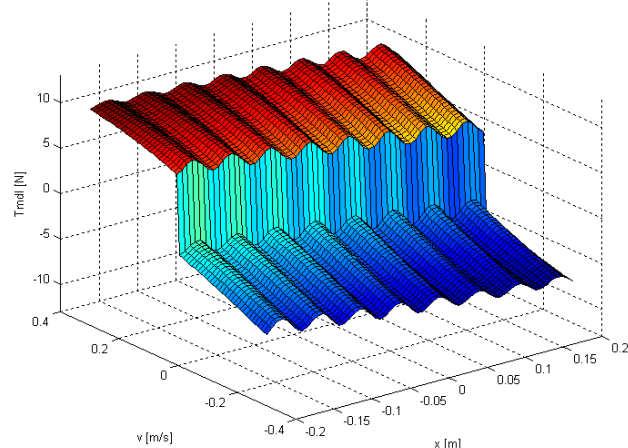
Sprawdzono także reakcję układu nauczonego (ustalone wagi) na wyłączenie uczenia $\eta=0$. Nieznaczne zwiększenie różnicy między współczynnikiem korygującym, a wartością 1 pokazane na rysunku 12 pokazuje, że zależność $T_{mdl}=f(x,v)$ została przez sieć zapamiętana w procesie powolnego ustalania wag, a nie jest tylko imitowana przez szybkie zmiany wag neuronu wyjściowego.



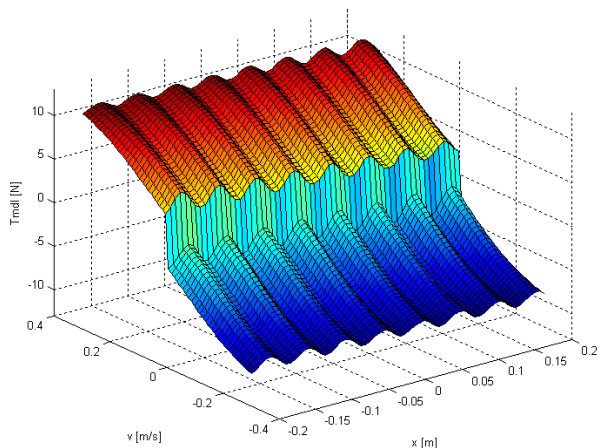
Rys.14. Powierzchnia wygenerowana przez model po 10s uczenia



Rys.15. Powierzchnia wygenerowana przez model po 20s uczenia

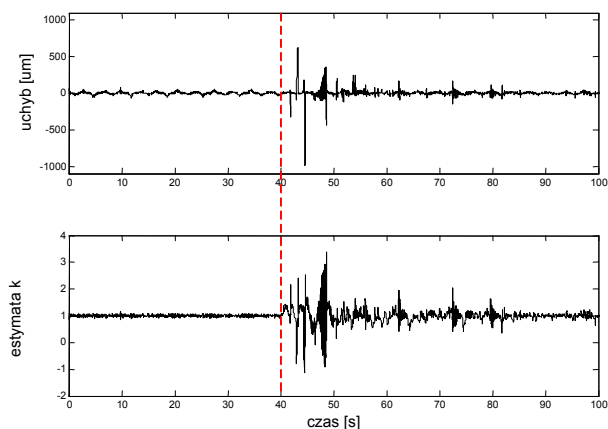


Rys.16 Powierzchnia wygenerowana przez model po 100s uczenia

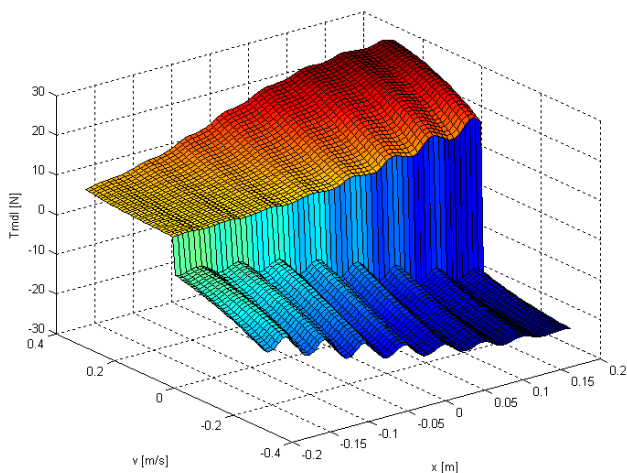


Rys.17. Powierzchnia wygenerowana przez model po zakończeniu uczenia

W oparciu o zarejestrowaną historię parametrów sieci, możliwe było pokazanie w jaki sposób ewoluował strojony kompensator neuronowy (rys.13-17)



Rys.18. Błąd śledzenia trajektorii zadanej i estymata k w trakcie zmian tarcia



Rys.19. Powierzchnia wygenerowana przez model w układzie ze zmienionym tarcie

Eksperyment 4

Sprawdzona została zdolność adaptacji nauczonej sieci (o powierzchni z rys.17) do nowych warunków tarcia. Tak jak to zostało zaznaczone na rysunku 2, podczas pracy w układzie z nauczoną siecią zmienione zostało tarcie poprzez dociśnięcie metalowego profilu do ruchomej części silnika. Zmiana błędu i estymaty \hat{k} została pokazana na rysunku 18. Powierzchnię wygenerowaną przez model podlegający dalszemu douczaniu przez kolejne 100s pokazuje rysunek 19.

Wnioski

W artykule przedstawiono adaptacyjne sterowanie położeniem silnika liniowego w warunkach zmieniającego się tarcia. Wykorzystano przy tym model neuronowy tarcia z rozszerzeniem funkcyjnym Pao uczony on-line. Wyniki otrzymane w drodze symulacji komputerowych zostały potwierdzone na stanowisku rzeczywistym.

Praca dotyczy sterowania elektrycznym silnikiem liniowym, jednak opisana w niej metoda sterowania i uczenia modelu tarcia może być wykorzystana wszędzie tam, gdzie występuje potrzeba kompensacji tarcia, a więc także w sterowaniu silnikiem obrotowym czy układem z siłownikiem pneumatycznym.

LITERATURA

- [1] Armstrong-Hélouvy B., Dupont P., Canudas de Wit C., A survey of models, analysis tools and compensation methods for the control of machines with friction, *Automatica, Journal of IFAC*, Volume 30, n.7, pp. 1083-1138, July 1994
- [2] Gao Z., Active disturbance rejection: A paradigm shift in feedback control system design, *Proc. IEEE Amer. Control Conf.*, pp.2399 -2405, 2006
- [3] Armstrong-Hélouvy B., Control of machines with friction, *Springer*, 1991
- [4] Hutamarn S., Pratumsuwan P., Pongaeen W., Adaptive neuro-fuzzy friction compensator in servo hydraulic system, *International Conference on Mechatronics and Automation (ICMA)*, pp. 2113-2117, Beijing, 2011
- [5] Ohri J., Dewan L., Soni M. K., Fuzzy adaptive dynamic friction compensator for robot, *International Journal of Systems Applications, Engineering & Development*, n.4, Volume 2, pp. 157-161, 2008
- [6] Han Me Kim, Seong Ik Han, Jong Shik Kim, Precision position control of servo systems using adaptive back-stepping and recurrent fuzzy neural networks, *Journal of Mechanical Science and Technology*, Volume 23, n.11, pp 3059-3070, 2009
- [7] Pao V. H., Adaptive Pattern Recognition and Neural Networks, *Addison-Wesley*, Reading MA 1996
- [8] Jastrzębski M., On-line parameter tuning of discontinuous fuzzy friction compensator in linear drive, *ICAT-2013 XXIV International Conference on Information, Communication and Automation Technologies*, Sarajewo, 2013
- [9] Jastrzębski M., Wykorzystanie rozmytego modelu siły oporu w adaptacyjnym sterowaniu silnikiem liniowym, *Przegląd Elektrotechniczny*, 88 (2012), nr 4B, 93-97
- [10] Kabziński J., Fuzzy modeling of disturbance torques/forces in rotational/linear interior permanent magnet synchronous motors, *European Conference on Power Electronics and Applications*, Dresden, 2005
- [11] Kabziński J., Chaotic mixing with adaptively controlled permanent magnet motor, *ISIE 2011, 20th IEEE International Symposium on Industrial Electronics*, 2011

Autor: dr inż. Marcin Jastrzębski, Politechnika Łódzka, Instytut Automatyki, ul. Stefanowskiego 18/22, 90-924 Łódź, E-mail: marcin.jastrzebski@p.lodz.pl