

Andrzej Żak
Akademia Marynarki Wojennej

NEURONOWY MODEL RUCHU POJAZDU PODWODNEGO

STRESZCZENIE

W pracy przedstawiono metodę modelowania ruchu bezzalogowego pojazdu podwodnego z wykorzystaniem techniki sztucznych sieci neuronowych. Opisano zastosowane sieci neuronowe, w tym dynamiczny model neuronu, metodę adaptacyjnego doboru jego parametrów oraz metodę nauczania neuronowej sieci wielowarstwowej. Zaprezentowano algorytm modelowania ruchu pojazdu podwodnego oraz przedstawiono wyniki badań eksperymentalnych.

WSTĘP

Modelowanie jest pojęciem bardzo ogólnym, mającym swój zwyczajowy sens w mowie potocznej. W technice jest ono ściśle związane z tworzeniem modeli badanych obiektów. Przez model obiektu rozumiane jest przedstawienie interesujących, istotnych właściwości rzeczywistego obiektu w dogodnej postaci. Model powinien zewnętrznie zachowywać się podobnie jak obiekt, aczkolwiek może mieć inną strukturę wewnętrzną. Tak więc, pod pojęciem modelowanie najczęściej rozumiana jest procedura, w wyniku której na podstawie sygnałów wejściowych i wyjściowych obiektu powstaje jego model uznany za najlepszy zgodnie z przyjętym kryterium jakości [2].

W wielu wypadkach modelowanie rozpoczyna się od określenia podstawowych praw fizycznych obowiązujących w badanych procesach. Z praw tych wynika pewna liczba zależności, a jeśli znane są wartości liczbowe wszystkich zewnętrznych i wewnętrznych warunków modelowanego układu i wiedza o fizyce zachodzących zjawisk jest pełna, to można wyznaczyć wartości liczbowe wszystkich współczynników. Jednak takie przypadki są rzadkie, ponieważ wymagana do prawidłowego

zaprojektowania układu wiedza o nim oraz nieokreśloności wnoszone przez środowisko rzadko są dostępne *a priori*, co wynika z niekompletnej wiedzy o zjawiskach fizycznych zachodzących wewnątrz procesu. Z drugiej strony, często interesujące są tylko zależności między sygnałami wejściowymi i wyjściowymi obiektu. W tym przypadku taki obiekt można potraktować jako „czarną skrzynkę”, tj. układ o jednym lub wielu wejściach i wyjściach, lecz o nieznannej strukturze.

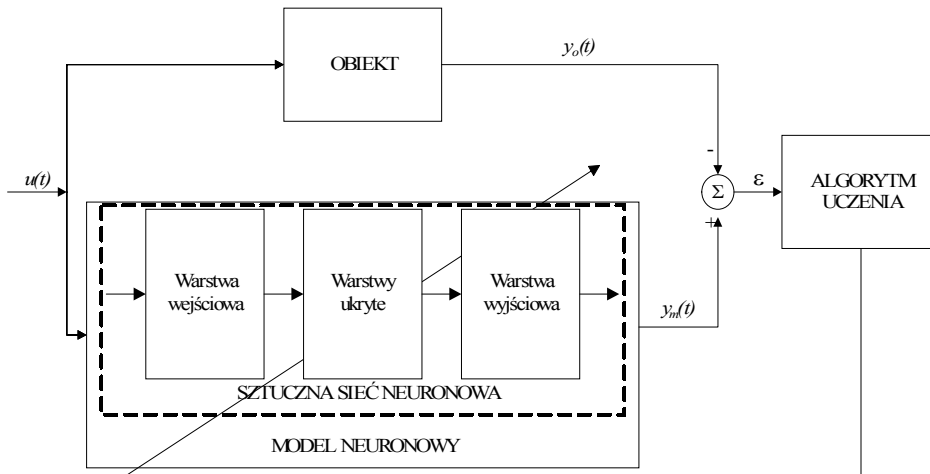
W ostatnich latach do modelowania obiektów wielowymiarowych coraz częściej wykorzystuje się metody oparte na sztucznych sieciach neuronowych [7, 8]. Ogromna popularność tej techniki i rosnąca fala jej zastosowań związana jest ze zdolnością sieci neuronowych do adaptacji i samoorganizacji, dzięki czemu może być ona wykorzystywana w wielu zastosowaniach. Sztuczne sieci neuronowe są systemami współbieżnymi, co daje możliwość znacznego przyspieszenia obliczeń w większości zadań, do których są stosowane. Ponadto istotnym atutem sieci neuronowych jest wygoda ich programowania poprzez uczenie. Dzięki temu nie jest wymagane tworzenie skomplikowanych odwzorowań matematycznych rzeczywistości, a jedynie ogólne określenie parametrów, co pozwala na realizację skomplikowanych zadań niezależnie od rozpatrywanego problemu.

SZTUCZNE SIECI NEURONOWE DO MODELOWANIA RUCHU POJAZDU

Opis procesu modelowania

Zagadnienia modelowania i identyfikacji obiektów dynamicznych stanowią jeden z podstawowych problemów w wielu metodach syntezy układów sterowania. Rozwój teorii sztucznych sieci neuronowych oraz technologicznych możliwości ich praktycznej realizacji stworzyły w ostatnich latach nowe efektywne i uniwersalne narzędzie wykorzystywane dla potrzeb modelowania. Ogólną strukturę układu realizującego zadanie syntezy modelu neuronowego pokazano na rysunku 1., a przedstawiona sytuacja odpowiada procesowi uczenia sieci neuronowej, tj. dostrajania wartości jej współczynników.

Informacja o wartości błędu będącego różnicą pomiędzy wyjściem obiektu a wyjściem modelu neuronowego stanowi informację wejściową w algorytmie uczenia.



Rys. 1. Synteza modelu neuronowego

W układach sterowania procesy mają charakter dynamiczny, neuronowe modelowanie zatem wymaga stosowania specjalnych rozwiązań. Ostatnio uwidoczniły się tendencje nadawania charakteru dynamicznego sieciom neuronowym poprzez wprowadzenie dynamiki procesu do neuronu w taki sposób, aby aktywność neuronu zależała od jego wewnętrznych stanów.

Synteza neuronowego algorytmu modelowania

Model dynamicznego neuronu

Właściwości sieci neuronowych oraz ich możliwości zależą zarówno od architektury połączeń między neuronami, jak i od rodzaju samych neuronów. Wprowadzając do statycznego modelu matematycznego sztucznego neuronu różnego rodzaju sprzężenia zwrotne, można otrzymać model dynamiczny. Dynamika zostaje wprowadzona do neuronu w taki sposób, aby aktywność neuronu zależała od jego wewnętrznych stanów. Realizuje się to poprzez dodanie do struktury neuronu liniowego systemu dynamicznego [1], z wykorzystaniem którego każdy neuron odtwarza przeszłe wartości sygnałów, mając do dyspozycji dwa zbiory sygnałów: sygnały wejściowe $x_i(k)$, dla $i = 1, 2, \dots, N$ oraz sygnał wyjściowy $y(k)$ w chwilach bieżących i przeszłych (rys. 2.). W dynamicznym modelu neuronu można wyodrębnić trzy bloki:

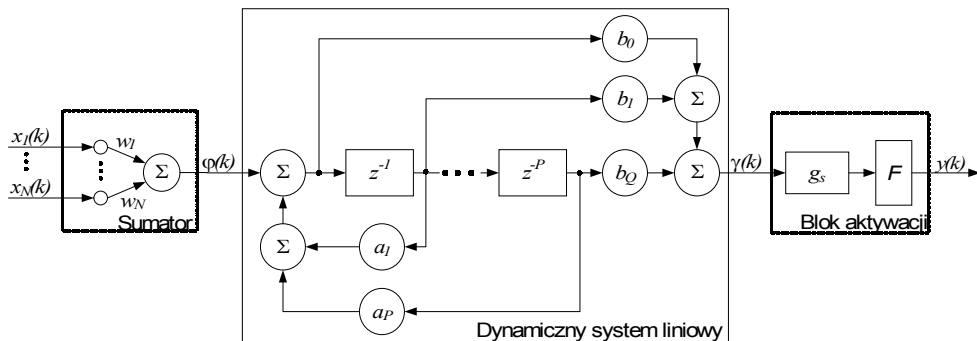
- sumator ważonych sygnałów wejściowych;
- dynamiczny system liniowy;
- nieliniowy blok aktywacji.

W bloku sumowania następuje obliczanie sumy ważonej informacji dochodzących do neuronu na podstawie zależności [1]:

$$\varphi(k) = \sum_{i=1}^N w_i(k)x_i(k) \quad (1)$$

gdzie: $w_i(k)$ – waga i -tego wejścia;
 $x_i(k)$ – i -ty sygnał wejściowy;
 N – liczba składowych sygnału wejściowego;
 k – indeks dyskretnego czasu.

Obliczona suma ważona jest następnie przetwarzana w dynamicznym systemie liniowym, który może być filtrem dowolnego rzędu. Struktura taka zawiera elementy opóźniające, sprzężenia zwrotne, połączenia jednokierunkowe oraz odpowiednio dopasowane wagi.



Rys. 2. Struktura dynamicznego modelu neuronu z N wejściami i jednym wyjściem

Jest ona opisana następującym równaniem różnicowym [1]:

$$\begin{aligned} \gamma(k) = & -a_1\gamma(k-1) - \dots - a_P\gamma(k-P) + \\ & + b_0\varphi(k) + b_1\varphi(k-1) + \dots + b_Q\varphi(k-Q) \end{aligned} \quad (2)$$

gdzie: $\varphi(k)$ – wejście bloku filtru w chwili k ;
 $\gamma(k)$ – wyjście filtru w chwili k ;
 $a = [a_1, \dots, a_P]$, $b = [b_0, \dots, b_Q]$ – wektory wag sprzężeń zwrotnych i połączeń jednokierunkowych;
 P, Q – wartości stałe.

Zgodnie ze strukturą dynamicznego modelu neuronu sygnał wyjściowy bloku dynamicznego systemu liniowego stanowi sygnał wejściowy bloku aktywacji. Ostatecznie sygnał wyjściowy neuronu będący sygnałem wyjściowym bloku aktywacji wyznaczany jest z zależności:

$$y(k) = F(g_s, \gamma(k)) \quad (3)$$

gdzie: $F(\cdot)$ – nieliniowa funkcja aktywacji;
 g_s – współczynnik nachylenia funkcji aktywacji.

Algorytm uczenia dynamicznego neuronu

Celem algorytmu jest wyznaczenie wartości parametrów dynamicznego modelu neuronu (wartości wag, wartości współczynników dynamicznego systemu liniowego oraz współczynnika nachylenia funkcji aktywacji) na bazie danego zbioru par wzorców wejściowych i wyjściowych. Ich wyznaczenia można dokonać poprzez rozwiązanie problemu optymalizacyjnego, w którym przyjmując błąd wyjściowy neuronu w postaci:

$$e(k) = y^d(k) - y(k) \quad (4)$$

gdzie: $y^d(k)$ – żądana odpowiedź układu;
 $y(k)$ – aktualna odpowiedź układu,

należy zminimalizować kryterium J mające postać:

$$J = \frac{1}{2} E\{e(k)^2\} \quad (5)$$

gdzie: E – operator wartości oczekiwanej.

Do rozwiązania tak sformułowanego problemu optymalizacyjnego i określenia optymalnych wartości parametrów neuronu zastosowano metodę gradientową największego spadku [3, 4, 5].

Niech $v = v(a, b, w, g_s)$ będzie uogólnionym parametrem sieci, $\frac{dJ}{dv}$ pochodną J względem v , a η współczynnikiem uczenia, to ogólny wzór na obliczenie wartości współczynników modelu neuronu ma postać:

$$\mathfrak{v}(k+1) = \mathfrak{v}(k) - \eta \nabla_{\mathfrak{v}} J \Big|_{\mathfrak{v}=\mathfrak{v}_k} \quad (6)$$

W wyniku prostych przekształceń otrzymuje się:

$$\frac{dJ}{d\mathfrak{v}} = E\{-e(k)F'(g_s\gamma(k))S_{\mathfrak{v}}(k)\} \quad (7)$$

gdzie: $S_{\mathfrak{v}}(k) = \frac{\partial\gamma(k)}{\partial\mathfrak{v}}$ – wektor wrażliwości sygnału $\gamma(k)$ na zmianę parametru \mathfrak{v} ;

$F'(\cdot)$ – pochodna funkcji aktywacji neuronu.

Ostatecznie zależność (6) przyjmuje postać:

$$\mathfrak{v}(k+1) = \mathfrak{v}(k) + \eta E\{e(k)F'(g_s\gamma(k))S_{\mathfrak{v}}(k)\} \quad (8)$$

Ponieważ dla ergodycznego procesu czasu dyskretnego wartość oczekiwana jest wartością średnią, to równanie (8) można zapisać w postaci:

$$\mathfrak{v}(k+1) = \mathfrak{v}(k) + \eta \sum_{i=1}^N e(i)F'(g_s\gamma(i))S_{\mathfrak{v}}(i) \quad (9)$$

Zgodnie z założeniami algorytmów uczenia sieci neuronowych [4] aktualizacja wartości parametrów modelu neuronowego następuje po przedstawieniu wszystkich wzorców uczących. W praktyce ich aktualizacje przeprowadza się po każdorazowej prezentacji wektora uczącego, stąd równanie (9) upraszcza się do postaci:

$$\mathfrak{v}(k+1) = \mathfrak{v}(k) + \eta e(k)F'(g_s\gamma(k))S_{\mathfrak{v}}(k) \quad (10)$$

Wyrażenia opisujące składowe wektora wrażliwości uogólnionego parametru \mathfrak{v} przedstawiają następujące zależności:

– współczynnik wrażliwości dla wartości wag

$$S_{w_i}(k) = g_s \frac{\partial\gamma(k)}{\partial w_i} = g_s \left(\sum_{q=0}^Q b_q x_i(k-q) - \sum_{p=1}^P a_p S_{w_i}(k-p) \right) \quad (11)$$

– współczynnik wrażliwości dla parametrów filtru

$$S_{a_p}(k) = g_s \frac{\partial\gamma(k)}{\partial a_p} = -g_s\gamma(k-p) \quad \text{dla } p = 1, \dots, P \quad (12)$$

$$S_{b_q}(k) = g_s \frac{\partial \gamma(k)}{\partial b_q} = -g_s \varphi(k-q) \quad \text{dla } q = 0, 1, \dots, Q \quad (13)$$

– współczynnik wrażliwości dla nachylenia funkcji aktywacji

$$S_{g_s}(k) = g_s \frac{\partial \gamma(k)}{\partial g_s} = \gamma(k) \quad (14)$$

Struktura dynamicznej sieci neuronowej

W praktyce elementarne operacje wykonywane przez pojedynczy dynamiczny neuron nie są zbyt interesujące, gdyż faktyczna moc obliczeń neuronowych wynika z połączenia wielu neuronów w sieć. Najczęściej struktura dynamicznej sieci neuronowej jest podobna do struktury statycznej jednokierunkowej sieci wielowarstwowej. Taka jej postać nie wprowadza globalnych sprzężeń zwrotnych, które komplikują architekturę sieci oraz zależności adaptacyjnego algorytmu uczenia.

Algorytm uczenia dynamicznej sieci neuronowej

W przypadku wielowarstwowej dynamicznej sieci neuronowej równania pojedynczego dynamicznego neuronu mogą być rozszerzone na przypadek całej sieci. Błąd obliczany na wyjściu sieci jest propagowany wstecz poprzez warstwy ukryte do warstwy wejściowej, podobnie jak w algorytmie wstecznej propagacji błędu [3, 4, 5].

Niech M oznacza liczbę warstw, s_m liczbę neuronów w m -tej warstwie, $y_i^m(k)$ wyjście i -tego neuronu położonego w m -tej warstwie w momencie k ($m = 0, 1, \dots, M$; $i = 0, 1, \dots, s_m$). Funkcja opisująca i -ty neuron w m -tej warstwie jest definiowana w postaci [6]:

$$\begin{aligned} y_i^m(k) &= F(g_{s_i}^m \gamma_i^m(k)) = \\ &= F(g_{s_i}^m [b_{0_i}^m \varphi_i^m(k) + b_{1_i}^m \varphi_i^m(k-1) + \dots + b_{n_i}^m \varphi_i^m(k-n) + \\ &\quad - a_{1_i}^m \gamma_i^m(k-1) - \dots - a_{n_i}^m \gamma_i^m(k-n)]) \end{aligned} \quad (15)$$

a ogólny błąd generowany przez ten neuron opisuje równanie [6]:

$$\delta_i^m(k) = -\frac{J(k)}{\partial \gamma_i^m(k)} = -\frac{J(k)}{\partial x_i^m(k)} \frac{\partial x_i^m(k)}{\partial g_{s_i}^m \gamma_i^m(k)} = -\frac{J(k)}{\partial x_i^m(k)} F'(g_{s_i}^m \gamma_i^m(k)) \quad (16)$$

Jego pierwsza część dla warstwy wyjściowej przyjmuje postać:

$$\frac{J(k)}{\partial x_i^M(k)} = \frac{\partial J(k)}{\partial y_i(k)} = -(y_i^d(k) - y_i(k)) = -e(k) \quad (17)$$

Natomiast dla warstw ukrytych wyznacza się ją następująco:

$$\begin{aligned} \frac{J(k)}{\partial x_i^m(k)} &= \sum_{j=1}^{s_{m+1}} \frac{\partial J(k)}{\partial g_{s_j}^{m+1} \gamma_j^{m+1}(k)} \frac{\partial g_{s_j}^{m+1} \gamma_j^{m+1}(k)}{\partial x_j^m(k)} \\ &= \sum_{j=1}^{s_{m+1}} \frac{\partial J(k)}{\partial g_{s_j}^{m+1} \gamma_j^{m+1}(k)} g_{s_j}^{m+1} b_{0j}^{m+1} w_{ij}^{m+1} = \sum_{j=1}^{s_{m+1}} -\delta_j^{m+1}(k) g_{s_j}^{m+1} b_{0j}^{m+1} w_{ij}^{m+1} \end{aligned} \quad (18)$$

Z powyższego wynika, iż ogólny błąd generowany przez neurony może być zapisany w postaci:

- dla warstwy wyjściowej

$$\delta_i^M(k) = e(k) F'(\gamma_i^M(k)) \quad (19)$$

- dla warstwy ukrytej

$$\delta_i^m(k) = \sum_{j=1}^{s_{m+1}} (\delta_j^{m+1}(k) g_{s_j}^{m+1} b_{0j}^{m+1} w_{ij}^{m+1}) F'(\gamma_j^m(k)) \quad (20)$$

Stąd zmiana parametrów i -tego neuronu w m -tej warstwie w postaci ogólnej przedstawiona jest następująco [6]:

$$v_i^m(k+1) = v_i^m(k) + \eta \delta_i^m(k) S_{v_i}^m(k) \quad (21)$$

gdzie: $S_{v_i}^m(k)$ – wektor wrażliwości na zmianę parametru v obliczany zgodnie z wyrażeniami (11) – (14).

BADANIA EKSPERYMENTALNE NA OBIEKCIE RZECZYWISTYM

Warunki prowadzenia badań

Badania w warunkach eksploatacyjnych przeprowadzono dla bezzałogowego zdalnie sterowanego pojazdu podwodnego typu „Ukwiął” w Porcie Wojennym Gdynia [6]. Bazę dla pojazdu stanowił okręt-niszczyciel min ORP „Flaming”. W czasie prowadzenia badań eksperymentalnych warunki hydrometeorologiczne były dobre.

Wektor sygnałów wejściowych (sterowań) miał następującą postać:

$$u_s = [\tau_X, \tau_Z, \tau_N] \quad (22)$$

gdzie: τ_X – siła naporu wzdłuż podłużnej osi pojazdu;
 τ_Z – siła naporu wzdłuż normalnej osi pojazdu;
 τ_N – moment obrotowy wokół normalnej osi pojazdu.

Natomiast wektor sygnałów wyjściowych (stanu) był w postaci:

$$x_e = [d, \psi] \quad (23)$$

gdzie: d – głębokość zanurzenia pojazdu mierzona z dokładnością do 0.1 m;
 ψ – kąt kursowy pojazdu mierzony z dokładnością do 0.5°.

Neuronowy algorytm modelowania ruchu pojazdu

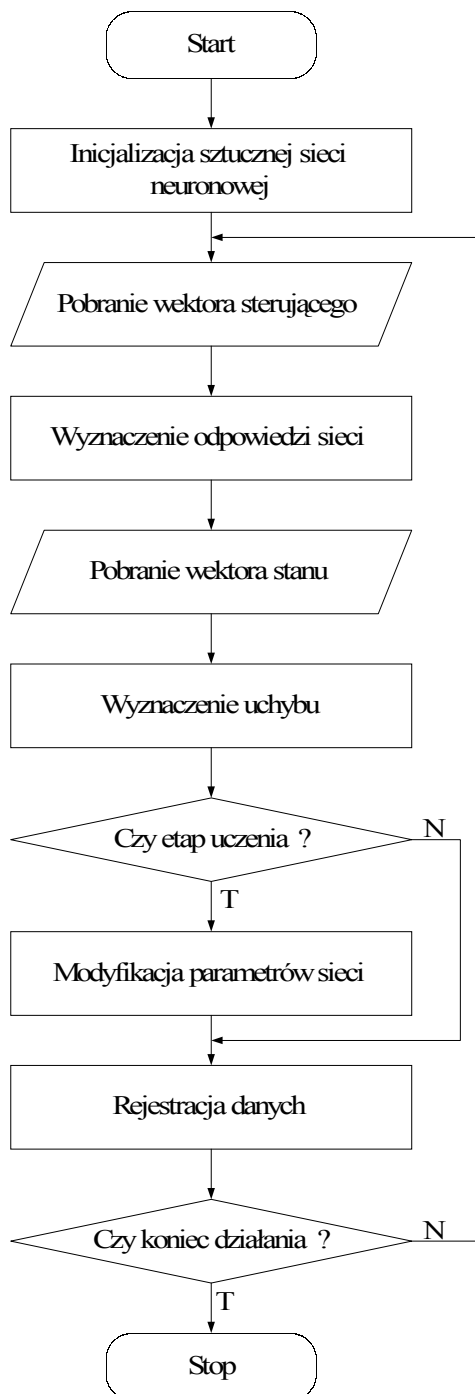
Dla potrzeb wyznaczania modelu dynamiki pojazdu, na podstawie przedstawionych w poprzednim rozdziale zależności, został opracowany moduł wykorzystujący dynamiczne sieci neuronowe, którego algorytm działania przedstawiono na rysunku 3 [6].

Na pierwszym etapie następuje inicjalizacja sztucznej sieci neuronowej składająca się z następujących czynności:

- wygenerowanie dynamicznych neuronów o strukturach opisanych wzorem (4);
- ustalenie początkowych wartości parametrów każdego neuronu, tj. wag wejściowych, współczynników dynamicznego systemu liniowego oraz kąta nachylenia funkcji aktywacji;
- wykonanie połączeń pomiędzy neuronami poprzez przypisanie każdemu neuronowi jego następników.

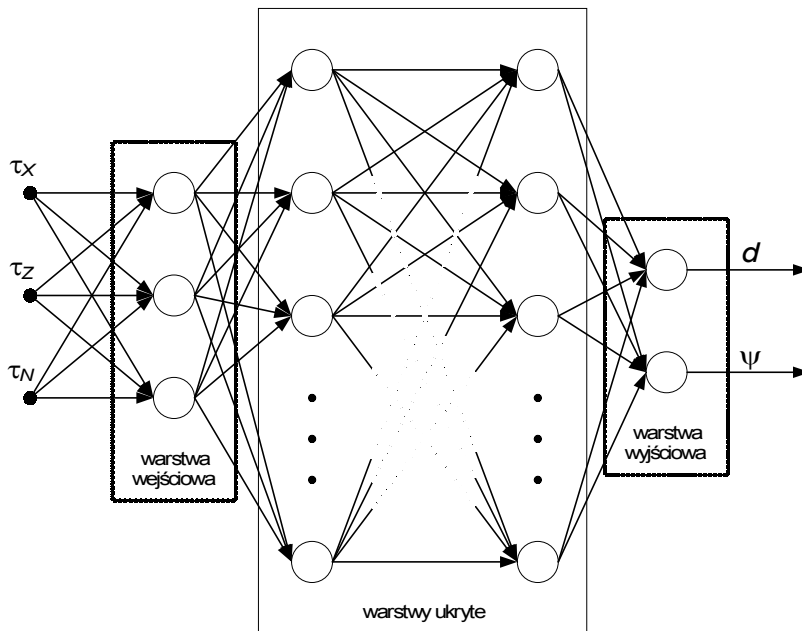
Jako funkcję aktywacji dla wszystkich neuronów przyjęto funkcję tangensoidalną.

Liczba neuronów w warstwie wejściowej uzależniona jest od rozmiaru wektora sterującego. W rozpatrywanym przypadku warstwa ta składała się z trzech neuronów. Podobnie rozmiar warstwy wyjściowej uzależniony był od rozmiaru wektora stanu i na etapie badań na obiekcie rzeczywistym były to dwa neurony. Natomiast liczbę neuronów w warstwach ukrytych dobrano w oparciu o miarę Vapnika-Chervonenkisa [4].



Rys. 3. Schemat blokowy algorytmu wyznaczania modelu dynamiki pojazdu podwodnego

Neurony powiązane na zasadzie każdy z każdym pomiędzy dwoma sąsiednimi warstwami bez połączeń wewnątrz warstwy oraz pomijających warstwę. Postać przyjętej do badań sieci neuronowej przedstawia rysunek 4.



Rys. 4. Postać sieci neuronowej przyjętej do badań eksperymentalnych

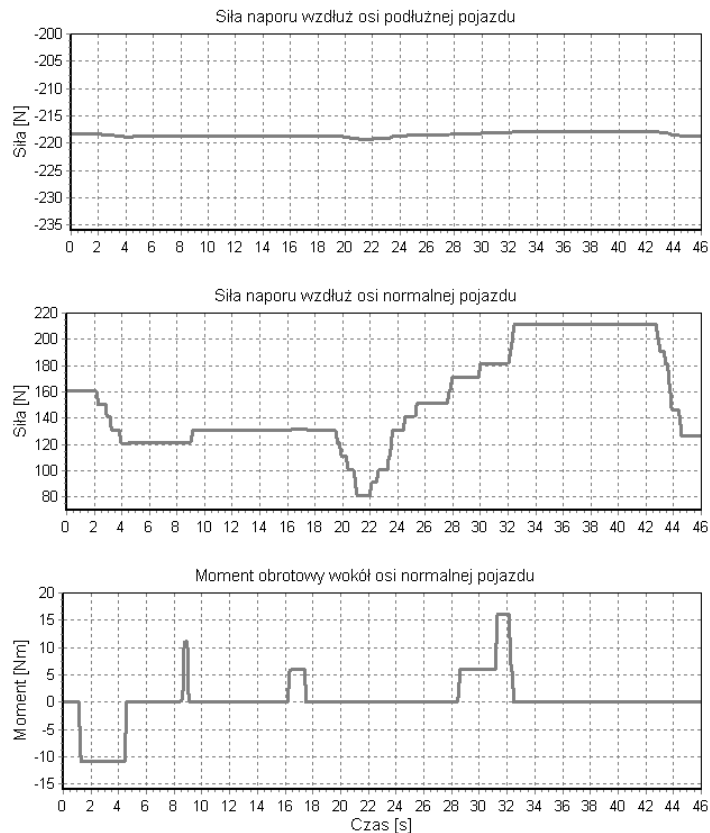
Drugim etapem algorytmu jest pobranie danego wektora sterującego, jego normalizacja, a następnie obliczenie odpowiedzi sieci neuronowej na zadane wymuszenie, zgodnie z równaniem (15). Na podstawie odpowiedzi sieci oraz pomiaru wartości współczynników wektora stanu modelowanego obiektu następowo wyznaczenie uchybu odpowiedzi sieci.

Kolejnym etapem było przeprowadzenie modyfikacji wartości parametrów każdego z neuronów. W tym celu błędy warstwy wyjściowej propagowano na warstwy poprzedzające, zgodnie ze wzorami (19) i (20), a następnie przeprowadzono modyfikację parametrów neuronów, zgodnie z zależnością (21). Tak wyznaczona odpowiedź sieci neuronowej była rejestrowana dla potrzeb dalszej analizy. Proces modelowania uważany był za zakończony po osiągnięciu określonego poziomu uchybu, który jest wskaźnikiem dopasowania modelu do obiektu. W przypadku niedopasowania algorytm wracał do etapu drugiego.

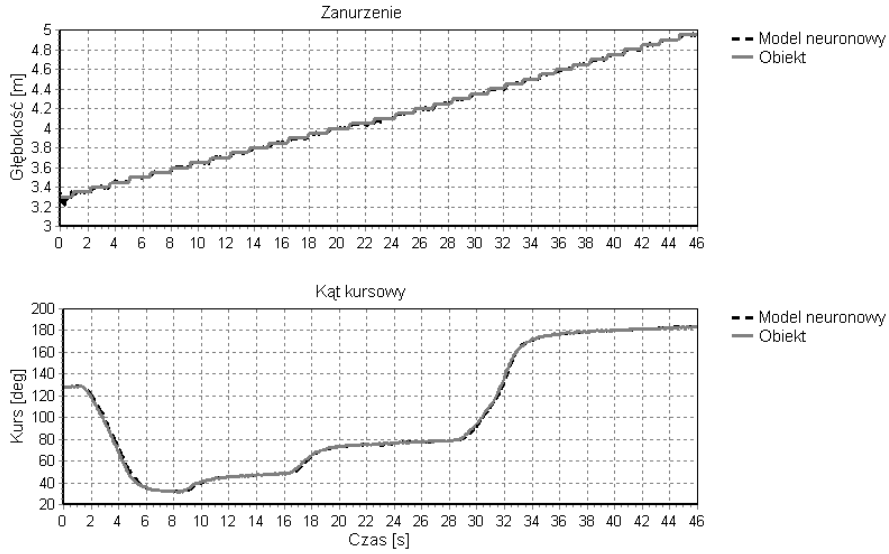
Wyniki badań

W pierwszej fazie badań eksperymentalnych wyznaczono neuronowy model ruchu pojazdu. Model ten działał jako układ predykcji, a jego parametry podlegały modyfikacjom zgodnie z wcześniej przedstawionym algorytmem. Etap ten zrealizowano w trakcie wykonywania zadania przejścia pojazdu w określony rejon działania, tj. z punktu do punktu. Wyniki z tej fazy badań przedstawiono na rysunkach od 5. do 7.

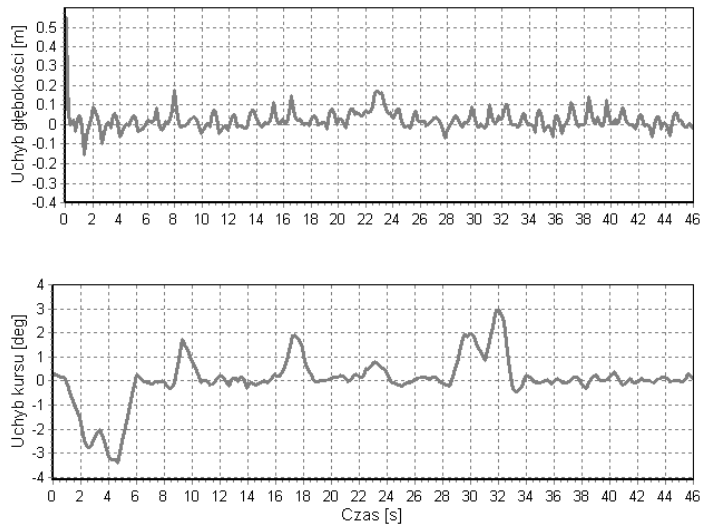
W kolejnej fazie eksperymentu układ modelowania przechodził w tryb symulacji, czyli wyznaczone parametry modelu nie podlegały modyfikacjom. Na tym etapie pojazd realizował zadanie podejścia do konstrukcji podwodnej. Wyniki z tego etapu badań przedstawiono na rysunkach od 8 do 10. Uzyskany niewielki uchyb odpowiedzi obiektu i utworzonego modelu neuronowego świadczy o dobrej jakości dopasowania modelu do obiektu.



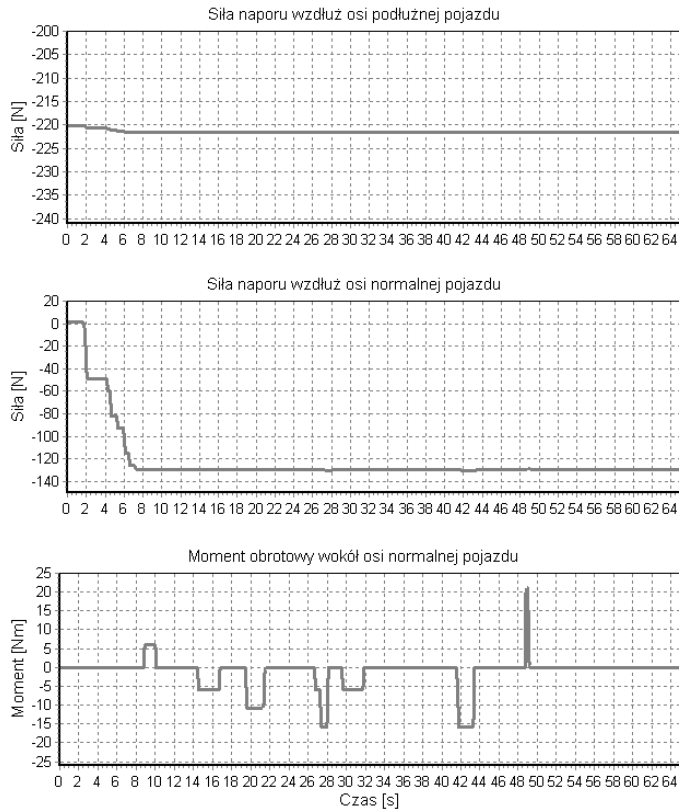
Rys. 5. Przebiegi sygnałów sterujących przy realizacji zadania przejścia pojazdu w zadany rejon działań



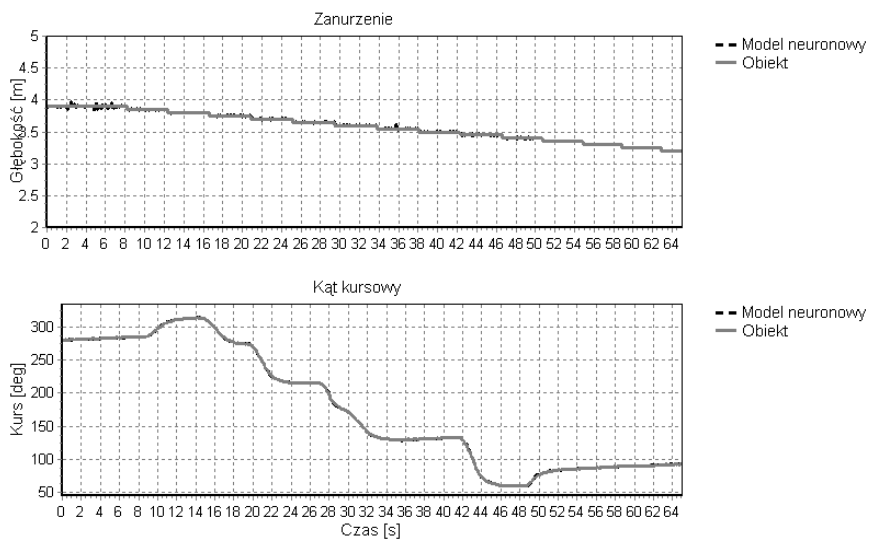
Rys. 6. Przebiegi współrzędnych wektora stanu pojazdu przy realizacji zadania przejścia pojazdu w zadany rejon działań



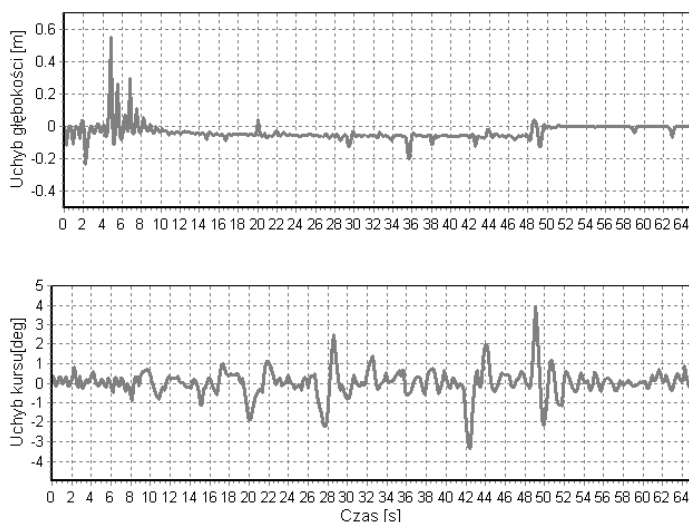
Rys. 7. Uchylb odpowiedzi modelu neuronowego i pojazdu przy realizacji zadania przejścia pojazdu w zadany rejon działań



Rys. 8. Przebiegi sygnałów sterujących przy realizacji zadania podejścia pojazdu do konstrukcji podwodnej



Rys. 9. Przebiegi współrzędnych wektora stanu pojazdu przy realizacji zadania podejścia pojazdu do konstrukcji podwodnej



Rys. 10. Uchyb odpowiedzi modelu neuronowego i pojazdu przy realizacji zadania podejścia pojazdu do konstrukcji podwodnej

PODSUMOWANIE

Przedmiotem badań było opracowanie neuronowego modelu ruchu pojazdu podwodnego. W czasie prowadzenia badań napotkano na wiele trudności, które były związane między innymi z doбором architektury dynamicznej sieci neuronowej realizującej syntezę układu wielowymiarowego, ograniczonym czasem prowadzenia uczenia sieci, pomiarem wektorów sterującego i stanu pojazdu podwodnego. Ich rozwiązanie pozwoliło na przeprowadzenie badań eksperymentalnych i wyznaczenie neuronowego modelu ruchu badanego obiektu.

Na podstawie uzyskanych rezultatów można stwierdzić, że modelowanie z wykorzystaniem sztucznych sieci neuronowych pozwala na prowadzenie identyfikacji dynamiki obiektu w czasie rzeczywistym, a zastosowanie do tego celu dynamicznych sieci neuronowych charakteryzuje się małym błędem modelowania.

Opisany w pracy sposób uzyskania modelu bezzałogowego pojazdu podwodnego może być wykorzystany zarówno w procesie syntezy układu sterowania, jak i do tworzenia symulatora ruchu – trenażera pilotażu bezzałogowego pojazdu podwodnego. Metoda ta może być również zastosowana do modelowania ruchu innych wielowymiarowych obiektów dynamicznych.

BIBLIOGRAFIA

- [1] Back A. D., Tsoi A. C., *FIR and IIR synapses. A new neural network architecture for time series modeling*, „Neural Computation”, 1991, Vol. 3, pp. 375 – 385.
- [2] Gutenbaum J., *Modelowanie matematyczne systemów*, Akademicka Oficyna Wydawnicza Exit, Warszawa 2003.
- [3] Koniński R. A., *Sztuczne sieci neuronowe. Dynamika nieliniowa i chaos*, WNT, Warszawa 2002.
- [4] Ossowski S., *Sieci neuronowe*, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa 1996.
- [5] Tadeusiewicz R., Duch W., Korbicz J., Rutkowski L., *Sieci neuronowe*, Akademicka Oficyna Wydawnicza Exit, Warszawa 2004.
- [6] Żak A.: *Identyfikacja dynamiki bezzałogowego pojazdu podwodnego w warunkach eksploatacyjnych*, rozprawa doktorska, AMW, Gdynia 2006.
- [7] Żak A., *Modelowanie dynamiki robotów podwodnych*, Materiały XV Krajowej Konferencji Automatyki, Warszawa 2005, s. 295 – 298.
- [8] Żak A., *Neural Algorithm of Underwater Vehicle's Dynamics Identification*, WSEAS „Transactions on Computers Research”, 2006, Vol. 1, No 2, pp. 89 – 94.

ABSTRACT

The paper presents the method used to model motion of an unmanned underwater vehicle with artificial neural networks. It describes neural networks, including the dynamic model of artificial neuron and adaptive method used to calculate its parameters the method for learning of multilayer dynamic neural network. It shows an algorithm used to model motion of an underwater vehicle and the results of experimental research.

Recenzent kmdr dr hab. inż. Jerzy Garus, prof. AMW