

Jerzy ROJ

POLITECHNIKA ŚLĄSKA, INSTYTUT METROLOGII, ELEKTRONIKI I AUTOMATYKI,
ul. Akademicka 10, 44-100 Gliwice

Neuronowa korekcja błędów dynamicznych przetwornika II-go rzędu

Dr inż. Jerzy ROJ



Absolwent Wydziału Elektrycznego Politechniki Śląskiej. Od 1989 r. pracownik Instytutu Metrologii, Elektroniki i Automatyki. Stopień doktora nauk technicznych uzyskał w roku 1998. Obecnie obszar jego zainteresowań obejmuje opracowywanie modeli błędów, modeli niepewności przetworników próbkujących, badania metrologiczne właściwości sieci neuronowych oraz możliwości ich wykorzystania do korekcji błędów statycznych i dynamicznych przetworników pomiarowych.

e-mail: jerzy.roj@polsl.pl

Streszczenie

W artykule przedstawiono wyniki wstępnych badań dotyczących możliwości wykorzystania jednokierunkowych sieci neuronowych do korekcji błędów dynamicznych wprowadzanych przez przetworniki opisane liniowym równaniem różniczkowym II-go rzędu. Oceniono zasadność stosowania tego rodzaju podejścia do zagadnienia korekcji błędów dynamicznych. Wnioski sformułowano w oparciu o wyniki badań symulacyjnych.

Słowa kluczowe: sztuczna sieć neuronowa, korekcja błędów dynamicznych, niepewność wyniku pomiaru.

Correction of second order converter dynamic errors implementing neural networks

Abstract

Dynamic properties of second order transducers are usually modelled by the linear differential equation (1) which can be converted to the discrete equation of state (6). Recursive solving of this equation for the input quantity (Eqs. 8 and 9) is a dynamic error correction algorithm. This algorithm can be written in the form of equations (10 and 11) which can be solved by simple, feed-forward neural networks of structures shown in Fig.1. Fig. 2 illustrates the use of neural networks for realisation of the dynamic correction recursive algorithm. The possibility of applying neural networks to dynamic error correction was investigated by simulations in the Matlab Neural Toolbox environment. There were taken the following assumptions concerning the transducer model: $\xi = 0,7$, $\omega_0 = 10 \text{ rad/s}$, $k_s = 1$ and the discretization period $T_d = 0,01 \text{ s}$. The network was learned using a 200 – element learning set generated on a basis of relation (14). The network was tested with a 200 000 – element testing set. The test results of both networks showed error – free implementation of (10) and (11) (errors of 10^{-15} order). At the next stage the learning sets were quantized with 12 – bit resolution. The influence of the discretization period on the accuracy of correction realisation was also investigated. Fig. 7 presents the results as a dependency of the output results Δ on the discretization period T_d .

Keywords: artificial neural network, dynamic error correction, uncertainty of a measurement result.

1. Wprowadzenie

Proces przetwarzania zachodzący w przetwornikach służących do cyfrowego pomiaru wartości chwilowych wielkości zmiennych w czasie składa się z reguły z kilku etapów. W pierwszym następuje przetwarzanie analogowe wielkości mierzonej na sygnał elektryczny za pomocą czujnika i odpowiednich układów kondycjonowania. Następnie realizowane jest próbkowanie i kwantowanie, mające na celu uzyskanie wartości cyfrowych proporcjonalnych do wartości chwilowych wielkości wyjściowych toru analogowego. W ostatnim etapie następuje przetwarzanie programowe,

umożliwiające uzyskanie odpowiednio dokładnych wyników pomiaru wielkości wejściowej w dyskretnych chwilach czasu. Przetwarzanie programowe, realizowane najczęściej przez mikroprocesor, polega na wykonywaniu działań określonych przez odpowiednie algorytmy korekcji błędów statycznych i dynamicznych, których właściwości współdecydują o dokładności całego toru przetwarzania. Implementacja tego rodzaju działań numerycznych zakłada znajomość odwrotnego modelu opisującego tor przetwarzania od jego wejścia do wyjścia przetwornika A/C.

W wielu sytuacjach identyfikacja dostatecznie dokładnego odwrotnego modelu przetwarzania analogowego jest procesem bardzo złożonym, bądź wręcz niemożliwym do przeprowadzenia. Wówczas można próbować zastępować wybrane algorytmy lub ich łańcuchy przez sieci neuronowe, które w procesie uczenia same „tworzą” właściwe modele odwrotne. Tego rodzaju właściwości sieci neuronowych wydają się mieć istotne znaczenie w odniesieniu do zagadnień związanych z korekcją błędów dynamicznych, których źródłem są z reguły właściwości dynamiczne czujników pomiarowych. Klasyczne algorytmy korekcji błędów dynamicznych charakteryzują się dużą złożonością działań numerycznych, zwłaszcza w przypadku opisu dynamiki czujników za pomocą równań różniczkowych wyższego rzędu.

2. Model przetwornika II-go rzędu

Właściwości dynamiczne przetworników II-go rzędu modeluje się z reguły za pomocą liniowego równania różniczkowego [1]

$$\ddot{y} + 2\xi\omega_0\dot{y} + \omega_0^2 y = \omega_0^2 k_s x, \quad (1)$$

gdzie: ξ jest tłumieniem względnym, ω_0 - pulsacją drgań swobodnych nietłumionych, k_s - współczynnikiem wzmocnienia statycznego. Zależność (1) można zapisać w postaci 2 równań stanu:

$$\begin{aligned} \dot{y}_1 &= y_2, \\ \dot{y}_2 &= -2\xi\omega_0 y_2 - \omega_0^2 y_1 + \omega_0^2 k_s x, \end{aligned} \quad (2)$$

gdzie \dot{y}_1 , \dot{y}_2 są zmiennymi stanu, przy czym $y_1 = y$.

Przedstawiając układ równań (2) w postaci macierzowej uzyskuje się

$$\dot{\mathbf{y}} = \mathbf{F}\mathbf{y} + \mathbf{G}x, \quad (3)$$

gdzie:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} 0 & 1 \\ -\omega_0^2 & -2\xi\omega_0 \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} 0 \\ \omega_0^2 k_s \end{bmatrix}. \quad (4)$$

Rozwiązaniem równania (3) jest wyrażenie

$$\mathbf{y}(t_{k+1}) = e^{\mathbf{F}T_d} \mathbf{y}(t_k) + e^{\mathbf{F}T_d} \int_{t_k}^{t_{k+1}} e^{-\mathbf{F}\tau} \mathbf{G}x(\tau) d\tau, \quad (5)$$

gdzie $T_d = t_{k+1} - t_k$ jest okresem dyskretyzacji. Zakładając, że wartości y nie zmieniają się między punktami próbkowania (zmiany y następują skokowo tylko w chwilach dyskretyzacji) [2], zależność (3) można zapisać w postaci dyskretnego równania stanu

$$\mathbf{y}(k+1) = \Phi \cdot \mathbf{y}(k) + \Psi \cdot x(k), \quad (6)$$

gdzie:

$$y(k) = \begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix},$$

$$\Phi = e^{F T_d} = \begin{bmatrix} \varphi_{11} & \varphi_{12} \\ \varphi_{21} & \varphi_{22} \end{bmatrix},$$

$$\Psi = G \cdot \int_0^{T_d} e^{F(T_d-\tau)} d\tau = \begin{bmatrix} \psi_1 \\ \psi_2 \end{bmatrix}. \quad (7)$$

3. Neuronowa implementacja algorytmu korekcji błędów dynamicznych

Jedną z metod korekcji błędów dynamicznych polega na rekurencyjnym rozwiązywaniu dyskretnego równania stanu (6) ze względu na wielkość wejściową [2] w wyniku czego uzyskuje się:

$$\hat{x}(k) = \frac{1}{\psi_1} [y(k+1) - \varphi_{11}y(k) - \varphi_{12}\hat{y}_2(k)], \quad (8)$$

$$\hat{y}_2(k+1) = \varphi_{21}y(k) + \varphi_{22}\hat{y}_2(k) + \psi_2\hat{x}(k), \quad (9)$$

gdzie $\hat{x}(k)$ stanowi ocenę wielkości wejściowej x w chwili k , $\hat{y}_2(k)$ jest oceną zmiennej stanu y_2 .

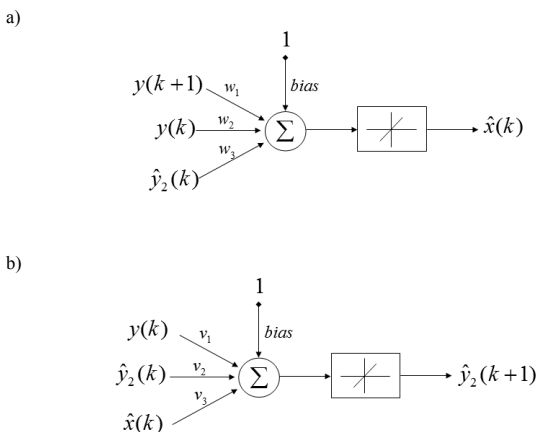
Równania (8) i (9) można zapisać jako:

$$\hat{x}(k) = w_1 \cdot y(k+1) + w_2 \cdot y(k) + w_3 \cdot \hat{y}_2(k), \quad (10)$$

$$\hat{y}_2(k+1) = v_1 \cdot y(k) + v_2 \cdot \hat{y}_2(k) + v_3 \cdot \hat{x}(k), \quad (11)$$

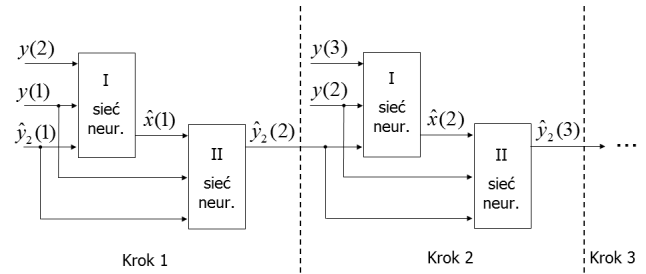
gdzie współczynniki $w_1 \dots w_3$ oraz $v_1 \dots v_3$ mają stałe wartości i są funkcjami $\varphi_{11} \dots \varphi_{22}, \psi_1, \psi_2$.

Działania opisane zależnościami (10) i (11) mogą być zrealizowane za pomocą prostych jednokierunkowych sieci neuronowych o 3 wejściach i liniowej funkcji aktywacji, których strukturę przedstawiono na rys. 1.



Rys. 1. Struktury sieci neuronowych realizujących działania opisane zależnościami a) (10), b) (11)
Fig. 1. Structure of neural networks implementing the steps described by a) Eq. (10), b) Eq. (11)

Wykorzystanie sieci neuronowych do realizacji rekurencyjnego algorytmu korekcji dynamicznej zilustrowano na rys. 2.



Rys. 2. Ilustracja neuronowej implementacji algorytmu korekcji dynamicznej. I sieć neuronowa ma strukturę z rys. 1 a), II sieć neuronowa ma strukturę z rys. 1 b)
Fig. 2. Illustration of neural implementation of dynamic correction algorithm. I Neural network has the structure of Fig. 1 a), II Neural network - the structure of Fig. 1 b)

4. Badania symulacyjne

Badanie możliwości wykorzystania sieci neuronowej do korekcji błędów dynamicznych przeprowadzono w sposób symulacyjny, wykorzystując do tego celu bibliotekę Neural Toolbox środowiska Matlab. Jednak przed przystąpieniem do ich realizacji należało przyjąć pewne założenia dotyczące parametrów modelu samego przetwornika, zarówno w postaci ciągłej, jak i dyskretniej. Bowiem wartości współczynników $\varphi_{11} \dots \varphi_{22}, \psi_1, \psi_2$ algorytmu korekcji błędów dynamicznych (8), (9) zależą od parametrów modelu przetwornika opisanego równaniem (1) oraz okresu dyskretyzacji T_d . W celu przeprowadzenia wstępnych badań przyjęto, że wartość współczynnika tłumienia $\xi = 0,7$, wartość pulsacji drgań własnych $\omega_0 = 10 \text{ rad/s}$, natomiast współczynnik wzmocnienia statycznego $k_s = 1$. Wybrane wartości parametrów charakteryzują zatem przetwornik II-go rzędu oscylacyjny. Przyjęto również, że okres dyskretyzacji wynosi $T_d = 0,01 \text{ s}$.

Proces uczenia sieci neuronowej wymaga zastosowania zbioru wzorców uczących [3]. Zbiór ten wygenerowano korzystając z zależności opisującej odpowiedź przetwornika oscylacyjnego na wymuszenie skokowe wielkości wejściowej od wartości x_0 do x_m

$$y(t) = (x_m - x_0) \cdot \left[1 - \frac{e^{-\xi\omega_0 t}}{\sqrt{1-\xi^2}} \cdot \sin(\omega t + \varphi_p) \right] + x_0, \quad (12)$$

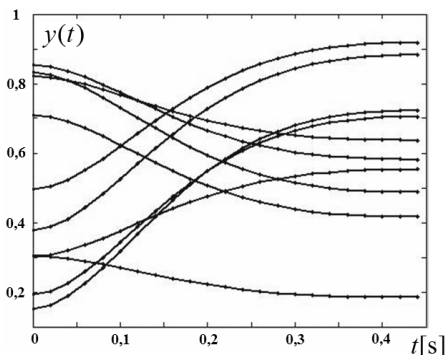
gdzie

$$\varphi_p = \arctg\left(\frac{\omega}{\xi\omega_0}\right). \quad (13)$$

Każdy element zbioru wzorców uczących sieci I powinien składać się zatem z wektora $w e_{ij}^I = [y_{ij}(t_j + T_d), y_{ij}(t_j), \hat{y}_{2ij}(t_j)]^T$ podawanego na wejścia sieci oraz wartości $w y_i^I = x_{mi}$, dla której wyznaczono wartości wektora $w e_{ij}^I$, podawanego na wyjście sieci. Indeks i oznacza kolejne wymuszenie, j – kolejną chwilę czasową. Podobnie dla sieci II - wektor wartości wejściowych $w e_{ij}^{II} = [x_{mj}, y_{ij}(t_j), \hat{y}_{2ij}(t_j)]^T$ oraz $w y_i^{II} = \hat{y}_{2ij}(t_j + T_d)$. Przygotowanie tego rodzaju zbiorów uczących wymaga znajomości m.in. kolejnych wartości zmiennej stanu \hat{y}_{2ij} , czyli wymaga wyznaczenia pochodnych. Dla celów badań symulacyjnych pochodne wyznaczono jedną z prostszych metod, korzystając z tzw. „różnicy centralnej”

$$\hat{y}_{2ij}(t_j) \cong \frac{y_i(t_j + T_d) - y_i(t_j - T_d)}{2T_d} \quad (14)$$

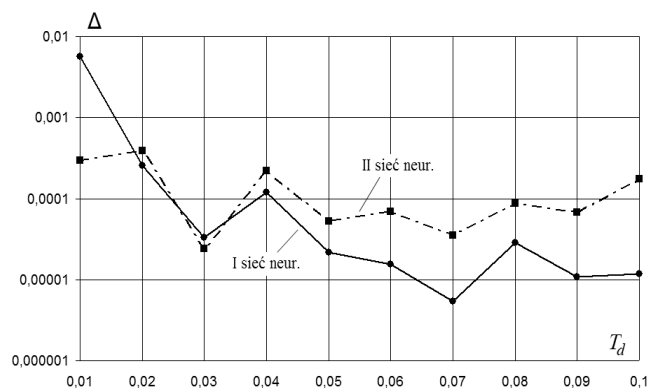
W procesie uczenia wykorzystano algorytm Levenberga-Marquardta [3] oraz 200-elementowe zbiory uczące. Dane wygenerowane na podstawie zależności (12) i (13), wykorzystane następnie do utworzenia zbiorów uczących przedstawiono na rys. 3.



Rys. 3. Dane wykorzystane do utworzenia zbiorów uczących sieci neuronowej

Fig. 3. The data used to create neural network learning sets

W celu sprawdzenia stopnia wytrenowania obu sieci przeprowadzono proces testowania za pomocą 200 000-elementowego zbioru testującego, którego wartości wygenerowano w sposób analogiczny jak zbioru uczącego. Wyniki testowania obu sieci wykazały praktycznie bezbłędną realizację działań (10) oraz (11) (błędy rzędu 10^{-15}). W kolejnym etapie badań zbiory uczące poddano kwantowaniu z 12-bitową rozdzielczością. Wyniki testowania sieci przedstawiono na rys. 4 jako zależność niepewności wyników wyjściowych Δ w funkcji okresu dyskretyzacji T_d . Sposób wyznaczania niepewności w tego rodzaju sytuacjach pomiarowych opisano m.in. w publikacjach [4, 5].



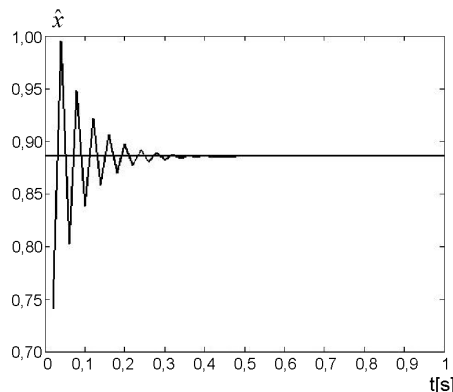
Rys. 4. Niepewność Δ wyniku na wyjściu sieci neuronowych realizujących działania (10) i (11) w funkcji okresu dyskretyzacji T_d

Fig. 4. Uncertainty Δ of the output result of neural networks implementing (10) and (11) vs. the digitization period T_d

5. Uwagi końcowe

W oparciu o przeprowadzone badania symulacyjne sformułowano następujące wnioski. Neuronową korekcję błędów dynamicznych można przeprowadzać dwojako. Wykorzystując rekurencyjne działanie łańcucha dwóch sieci – w takim przypadku nie

jest konieczna znajomość kolejnych wartości zmiennej stanu \hat{y}_2 . W pierwszym kroku można przyjąć \hat{y}_2 równą zero. Powoduje to wystąpienie stanu nieustalonego algorytmu, którego przykładowy przebieg przedstawiono na rys. 5.



Rys. 5. Przebieg stanu nieustalonego algorytmu korekcji realizowanego przez łańcuch 2 sieci neuronowych

Fig. 5. Transient state of the correction algorithm implemented by a chain of two neural networks

Tego rodzaju działanie może być wykorzystywane w bieżącej, ciągłej korekcji dynamicznej. Drugim sposobem jest wykorzystanie tylko pierwszej sieci i wyznaczanie wartości zmiennej stanu (pochodnej) na bieżąco – działanie nierekurencyjne. Z danych przedstawionych na rys. 4 wynika, że dla okresu dyskretyzacji $T_d > 0,03s$ niepewność wyników po korekcji jest mniejsza od 10^{-4} . Oznacza to, że w takim przypadku wynik można uzyskać już po czasie 0,06s od rozpoczęcia pomiarów (przy braku stosowania korekcji po czasie 1,2s).

Przeprowadzone badania mają wstępny charakter i nie uprawniają do uogólnień. Jednak można wskazać istotne zalety neuronowej implementacji opisanego algorytmu korekcji błędów dynamicznych. Po pierwsze nie jest konieczne przeprowadzanie identyfikacji parametrów przetwornika oraz wyznaczanie współczynników algorytmu korekcji. Sieć realizuje te działania w procesie uczenia. Po wtóre, opisana neuronowa korekcja błędów dynamicznych działa dla wymuszenia o dowolnym przebiegu.

6. Literatura

- [1] Hagel R., Zakrzewski J.: Miernictwo dynamiczne. WNT, Warszawa, 1984.
- [2] Jakubiec J.: Bieżące programowe odtwarzanie wartości chwilowych dynamicznych przebiegów wejściowych nieliniowych przetworników pomiarowych. Zeszyty Naukowe Pol. Śl., z. 964, Gliwice 1988.
- [3] Osowski S.: Sieci neuronowe do przetwarzania informacji. Oficyna Wydawnicza Politechniki Warszawskiej. Warszawa 2006.
- [4] Roj J.: Właściwości metrologiczne jednokierunkowych sieci neuronowych. Zeszyty Naukowe Pol. Śl., seria ELEKTRYKA, z. 205, str. 87-100, 2008.
- [5] Jakubiec J., Makowski P., Roj J.: Error Model Application in Neural Reconstruction of Nonlinear Sensor Input Signal. IEEE Transactions on Instrumentation and Measurement, Vol. 58, no. 3, str. 649-656, 2009.

otrzymano / received: 28.07.2010

przyjęto do druku / accepted: 01.10.2010

artykuł recenzowany