

Wpływ informacji o zmiennych stanu obiektu na jakość sterowania przez neurosterownik

The impact of information about the state variables on the quality control of neurodriver

Marcin Lis, Piotr Kozierski, *Politechnika Poznańska*

Abstract

In the paper there has been shown the impact of information of the state variables on the quality control using neural network. It is shown how from one initial configuration of weights, the network may evolve differently, using various input signals. There has been proposed 3 different systems for simulations, each with different inputs to the neural network (fig. 1). Each of this 3 systems have a correspond waveform of signal evolution in time (fig. 3 – 5). As one can see, adding state variable to the control loop greatly increases neural network performance. Neural network is learning using *Adaptive Interaction* method. The first chapter is an introduction to the subject of the article. Chapter 2 discusses the entire system, the object, the observer and neurodriver. Chapter 3 presents the simulation results and conclusions.

Streszczenie

W pracy został omówiony wpływ informacji o zmiennych stanu obiektu na jakość sterowania za pomocą sieci neuronowej. Pokazano, w jaki sposób z jednej, początkowej konfiguracji wag, sieć za pomocą różnych zestawów sygnałów wejściowych może różnie ewoluować. Rozdział pierwszy stanowi wprowadzenie do tematyki artykułu. W rozdziale 2 omówiono cały układ, tj. obiekt, neurosterownik i obserwator. Rozdział 3 przedstawia wyniki symulacyjne oraz wnioski.

1. Wprowadzenie

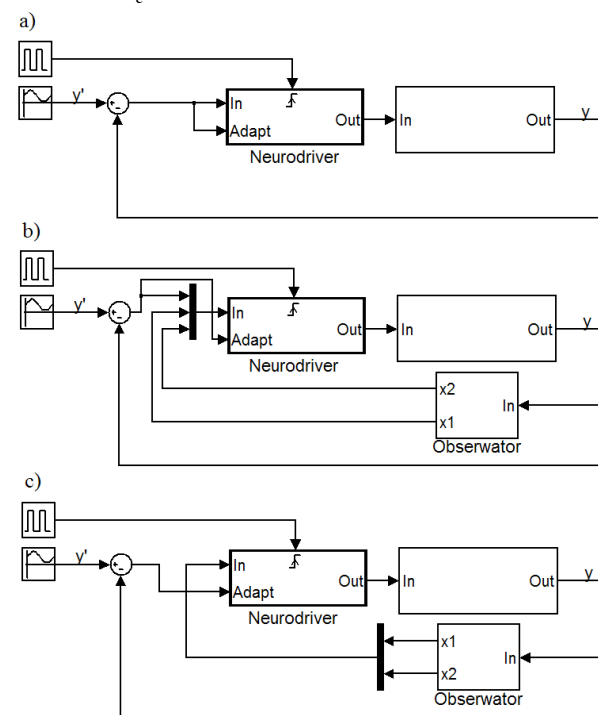
W poprzednich publikacjach [3], [4] rozważano zastosowanie sieci neuronowej do sterowania obiektem w pętli sprzężenia zwrotnego. Pokazano wpływ poszczególnych parametrów na jakość nadążania odpowiedzi z obiektu za sygnałem referencyjnym. Wykorzystując tą wiedzę, podjęto próbę rozszerzenia sterowania, sprawdzając wpływ podania zmiennych stanu na wejście sieci.

W dobie ogólnodostępnych obserwatorów stanu (filtr Kalmana i jego ulepszenia (EKF, UKF) [5], [6],

filtr cząsteczkowy [2], itp.), dojście do poszczególnych zmiennych stanu jest możliwe. Dzięki temu można wykorzystać te zmienne jako dodatkowe wejścia do sieci, uzyskując o wiele lepsze sterowanie np. jeżeli chodzi o jakość nadążania za sygnałem referencyjnym.

2. Układ

Schematy układów, które zostały poddane badaniom symulacyjnym znajdują się na rys. 1. Różnią się one przede wszystkim sygnałem podawanym na wejście sieci neuronowej. Opis tych układów znajduje się w podrozdziale opisującym sieć neuronową.



Rys.1. Schematy układów do symulacji.

Fig.1. Circuit diagrams for simulation.

2.1 Obiekt sterowania

Obiektem badań jest nieliniowy układ zapisany za pomocą dyskretnych równań stanu:

$$\begin{cases} x_1(k+1) = 0.01 \cdot (x_1(k) \cdot x_2(k) + e^{x_1(k)} \cdot e^{-x_2(k)}) \\ x_2(k+1) = 0.01 \cdot (x_1(k) + x_2(k)) + u \\ y(k) = (x_1(k))^{\frac{2}{3}} + (x_2(k))^{\frac{2}{3}} \end{cases} \quad (1)$$

Układ zachowuje się stabilnie dla określonego zakresu wymuszeń. W tym zakresie można wyznaczyć jego odpowiedź na skok jednostkowy, a podając na wejście szum – zidentyfikować go.

2.2 Obserwator

W ramach uproszczenia zostało założone, że istnieje dostęp do zmiennych stanu tego obiektu. Obserwator, który miałby pełnić taką rolę został przyjęty jako jednostkowe wzmocnienie, a sygnał jest pobierany bezpośrednio ze wzoru obiektu. Aby zachować konwencję, na rys. 1 został on przedstawiony jako osobny blok wykonujący zadanie identyfikacji.

2.3 Sygnał referencyjny

Celem sterowania jest nadążanie wyjścia z obiektu za sygnałem referencyjnym, który ma postać:

$$y' = 0.3 \cdot \sin(t - \frac{\pi}{2}) + 0.5 \quad (2)$$

Należy zwrócić uwagę na to, że sygnał sterujący został dobrany tak, aby nigdy nie osiągnąć wartości mniejszych niż 0. Ma to związek ze wzorem wyjściowym obiektu, niezależnie od zmiennych stanu jest ono zawsze nieujemne.

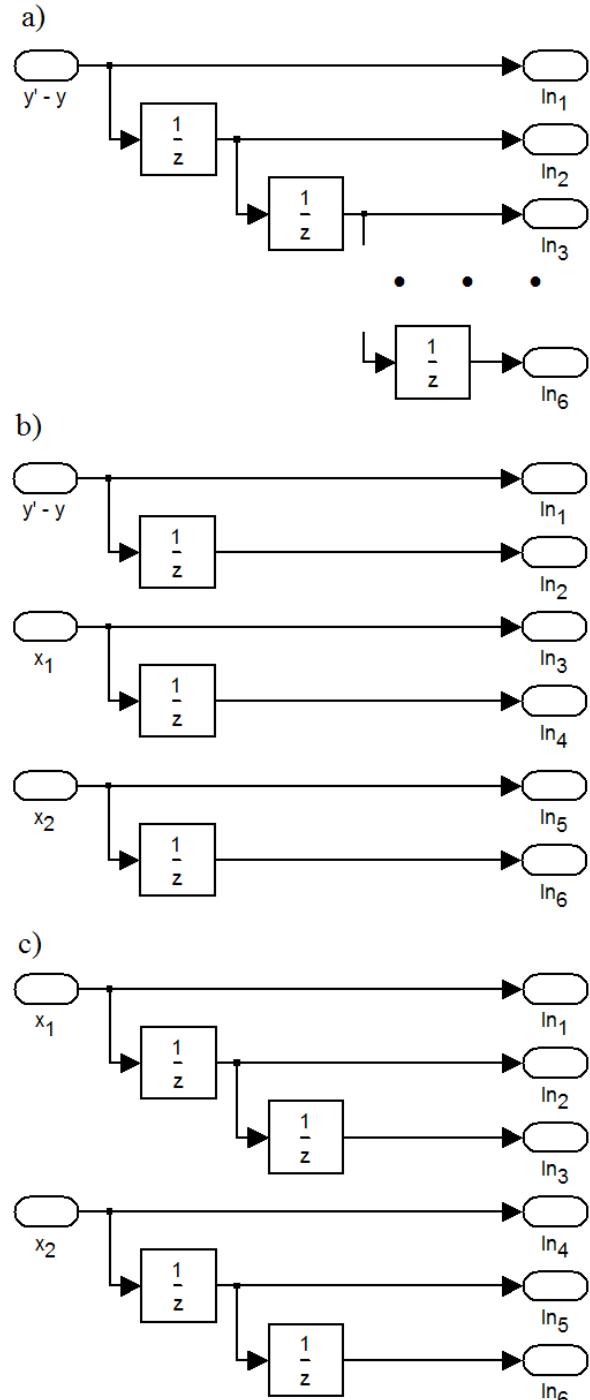
2.4 Sieć neuronowa i założenia symulacji

Sieć neuronowa wykorzystana jako regulator ma strukturę *feed-forward*. Ilość sygnałów wejściowych i neuronów w warstwie ukrytej przyjęto jako 6 (chyba, że podano inaczej). Metoda uczenia sieci: *Adaptive Interaction* [1]. Podczas jednej symulacji generowane zostają trzy zestawy danych, dla trzech różnych konfiguracji wartości wejściowych do sieci neuronowej. Adaptacja odbywa się według sygnału podanego na wejście *Adapt*, sygnałem tym zawsze jest uchyb $y' - y$.

W przypadku rys. 1a) na pierwsze wejście sieci podawany jest uchyb $y' - y$, na każde kolejne opóźnienie tejże wartości według schematu z rys. 2a). Podanie na wejście sieci neuronowej sygnału oryginalnego i jego pięciokrotnego opóźnienia pozwala lepiej odzwierciedlić dynamikę obiektu, co przyczynia się do dokładniejszych rezultatów w sterowaniu.

W przypadku rys. 1b) do sieci podawane są 3 sygnały wejściowe: jeden z uchybu, oraz dwa ze zmiennych stanu, jak pokazano na rys. 2b).

Dla rozwiązania z rys. 1c) na wejścia sieci podawane są dwa sygnały wygenerowane przez obserwator, tak jak pokazano na rys. 2c). Należy zwrócić uwagę na to, że uchyb $y' - y$ nie bierze bezpośredniego udziału w generowaniu sygnału sterującego obiektem. Jest on wykorzystywany tylko w procesie adaptacji.



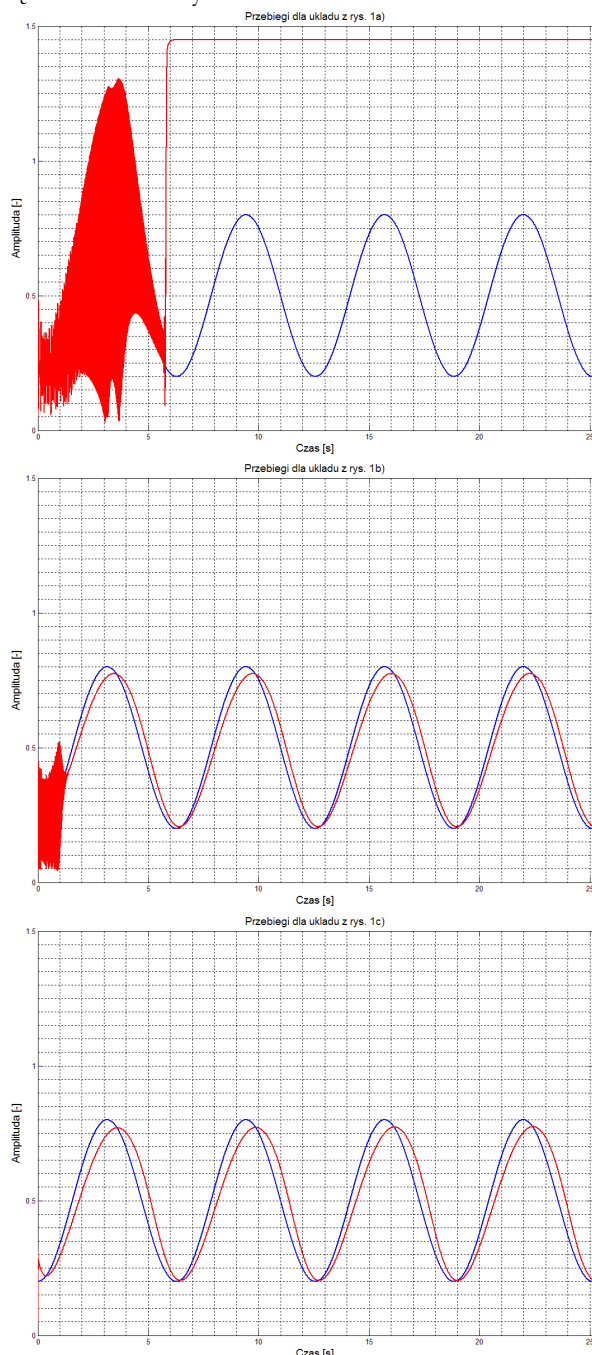
Rys.2. Opóźnienie sygnałów dla układów z rys. 1.
Fig.2. Signal delay for systems from fig. 1.

Dodatkowo, sieci są kopiami siebie, tzn. wylosowane wagi w sieci dla rys. 1a) odpowiadają wagom w sieciach na rys. 1b) oraz 1c). Parametry takie jak lr oraz Tp (wartość współczynnika adaptacji oraz czasu pomiędzy kolejnymi obliczeniami sieci) są

stałe dla całej symulacji. Na 10 odpowiedzi układu sieć odpowiada raz. Funkcje aktywacji: pomiędzy warstwą ukrytą a wyjściową: logsig , za neuronem wyjściowym: $150 \cdot \text{tansig}$.

3. Wyniki symulacyjne i wnioski

Do każdej grupy eksperymentów symulacyjnych były losowane nowe wagi sieci. Otrzymane przebiegi są widoczne na rys. 3 – 5.

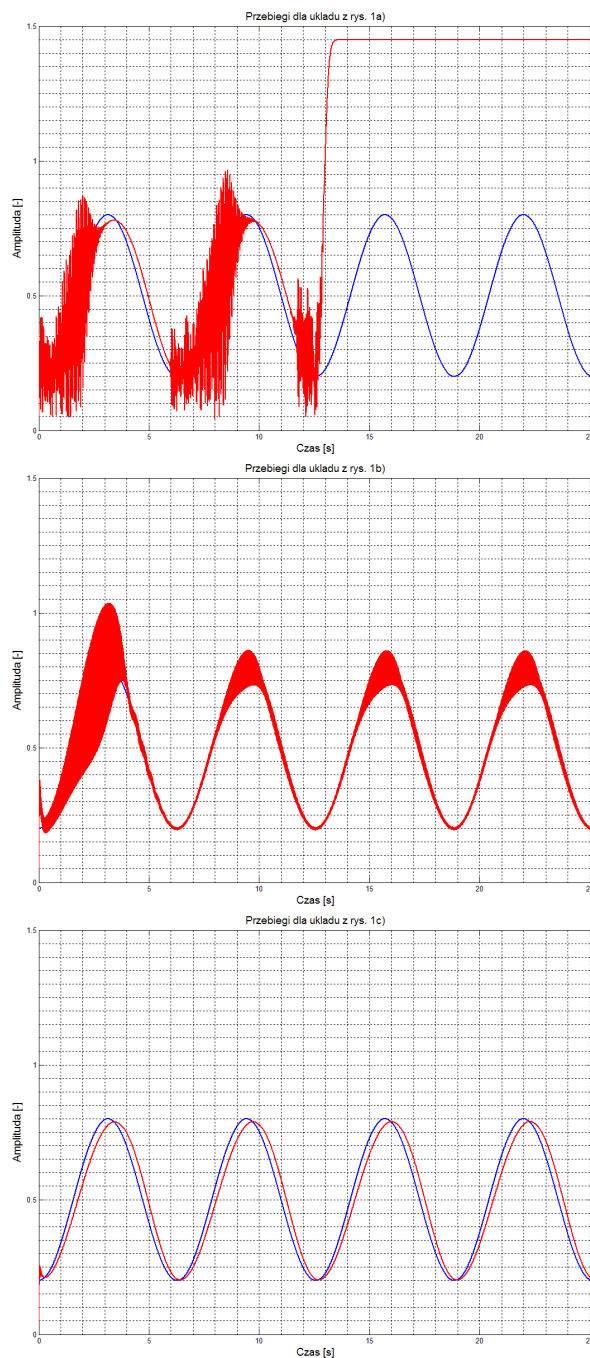


Rys.3. Przebieg sygnału referencyjnego i odpowiedzi obiektu.

Fig.3. Reference waveform and response of the object.

Jak można zaobserwować na przebiegach, dodanie sprzężenia od zmiennej stanu w znakomitej większości wpływa pozytywnie na działanie układu. Wychodząc z tego samego punktu początkowego

(wag), sieć jest w stanie ewoluować w różnych kierunkach, od braku sterowania obiektem do idealnego nadążania za sygnałem referencyjnym.

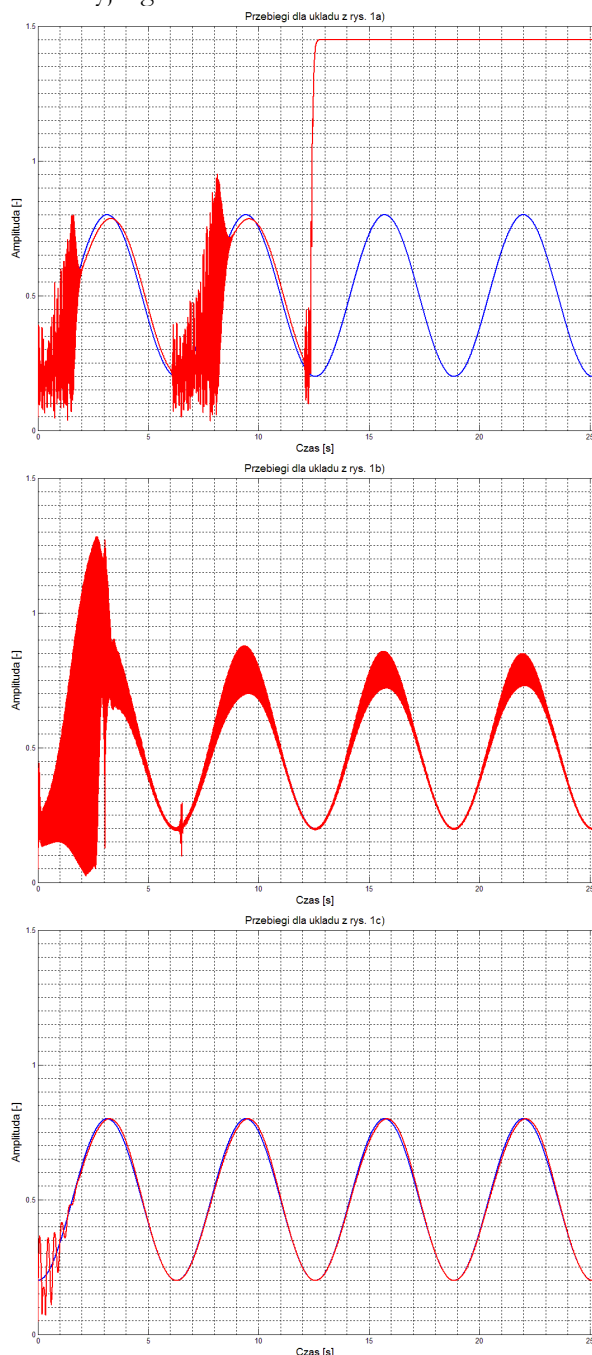


Rys.4. Przebieg sygnału referencyjnego i odpowiedzi obiektu.

Fig.4. Reference waveform and response of the object.

Wyniki nie są jednak jednoznaczne. Oprócz tych, opisanych w tym artykule wygenerowano kilkadziesiąt innych przebiegów dla takich układów. W znakomitej większości dla założonych warunków sieć sterowana według rys. 1a) nie była w stanie wysterować obiektem, natomiast sieć z rys. 1b) radziła sobie lepiej (mniejsze przeregulowania) niż ta z rys. 1c). Sieć z rys. 1c) wykazywała tendencję, podobnie jak ta z rys. 1a) do błędnego sterowania obiektem.

Jeżeli zdarzy się, że w pewnym momencie mechanizm adaptacji zawiedzie, dla tego obiektu każda z zaproponowanych sieci przestaje realizować swoje zadania. Wagi każdej sieci ulegają zmianom przez cały czas trwania symulacji. Na rys. 6 zaprezentowano przebiegi zmian dwóch wag (dwie pierwsze wagi neuronu wyjściowego). Widać, jak wagi oscylują wokół pewnych wartości. Oscylacje te mają znamiona częstotliwości sygnału referencyjnego.

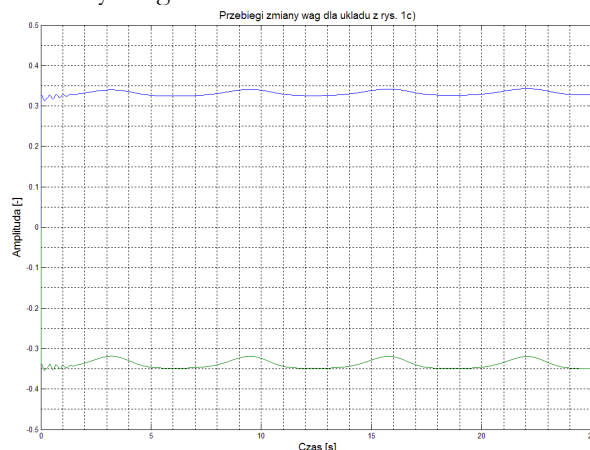


Rys.5. Przebieg sygnału referencyjnego i odpowiedzi obiektu, ilość neuronów w warstwie ukrytej = 10.

Fig.5. Reference waveform and response of the object, number of hidden neurons increased to 10.

Zauważono również następującą zależność: zwiększenie ilości sygnałów wejściowych (więcej jak 12) jak i neuronów w warstwie ukrytej (więcej jak 10)

powoduje brak umiejętności sieci do efektywnego sterowania obiektem.



Rys.6. Przebieg ewolucji wag w sieci neuronowej, ilość neuronów w warstwie ukrytej = 10.

Fig.6. Course of the evolution of weights in the neural network, number of hidden neurons increased to 10.

Literatura

1. Brandt R. D., Feng Lin, *Adaptive interaction and its application to neural networks*, Information Sciences, Volume 121, Issues 3–4, 2 December 1999
2. Doucet A., Johansen A.M.: *A Tutorial on Particle Filtering and Smoothing: Fifteen years later*, handbook of Nonlinear Filtering 2009/12, pp.656-704.
3. Lis Marcin: *Sieć neuronowa jako regulator obiektu dynamicznego*, Poznań University of Technology Academic Journals: Electrical Engineering, 2012 Issue 71, s. 17 – 24
4. Lis Marcin: *Wpływ czasu odpowiedzi neurosterownika na jakość regulacji*, Materiały konferencyjne, Warsztaty Doktoranckie 2012, lipiec 2012, Lublin
5. Maria Isabel Ribeiro: *Kalman and Extended Kalman Filters: Concept, Derivation and Properties*, Institute for Systems and Robotics, February 2004
6. Rambabu Kandepu, Bjarne Foss, Lars Imsland: *Applying the unscented Kalman filter for nonlinear state estimation*, Journal of Process Control, November 2007

Autorzy:



Mgr inż. Marcin Lis
Politechnika Poznańska,
Wydział Elektryczny
ul. Piotrowo 3A
60-965 Poznań
email:
mail.dla.studenta@gmail.com



Mgr inż. Piotr Koziński
Politechnika Poznańska
Wydział elektryczny
Ul. Piotrowo 3A
60-965 Poznań
email:
piotr.koziński@gmail.com