

WPŁYW CZASU ODPOWIEDZI NEUROSTEROWNIKA NA JAKOŚĆ REGULACJI

Marcin Lis

Politechnika Poznańska, Wydział Elektryczny

Streszczenie. W publikacji umieszczono wyniki dotyczące badania jakości sterowania siecią neuronową pracującą z różnym czasem wykonywania obliczeń. Celem jest wyznaczenie (dla założonego obiektu) granicy częstotliwości, z jaką musi pracować sieć aby sterowanie spełniało założone wskaźniki jakości. Do uczenia sieci neuronowej wykorzystano algorytm adaptacyjnej interakcji.

Słowa kluczowe: sieć neuronowa, adaptive interaction, jakość regulacji, czas odpowiedzi, neurosterownik

RESPONSE TIME OF NEURODRIVER VS QUALITY OF REGULATION

Abstract. In the article reader can find the results of testing the quality of neural network control with varying calculations time. The aim is to determine (for the assumed object) the limit frequency with which it must work to meet established quality indicators. Neural network learning algorithm uses the Adaptive Interaction method.

Keywords: neural network, adaptive interaction, quality of regulation, response time, neurodriver

Wstęp

W publikacji [2] przedstawiono wyniki eksperymentów symulacyjnych dla sieci neuronowej sterującej obiektem, realizującej zadanie nadążania za sygnałem referencyjnym. Przyjęto tam założenie, iż sieć wykonuje obliczenia (wartości wyjściowej oraz adaptacji) co krok symulacyjny. Tematem tego artykułu jest zbadanie wpływu ograniczenia ilości operacji w stosunku do ilości kroków, które wykonuje symulacja oraz zbadanie, jak wpływa to na jakość sterowania obiektem.

Stwierdzenie, że sieć neuronowa może wykonywać swoje obliczenia stosunkowo „rzadziej” jest bardzo pomocne w kontekście używania jej w systemach mikroprocesorowych. Przy sterowaniu przekształtnikami wystarczy raz na okres kluczowania elementu mocy zmieniać jego wypełnienie (w przypadku metody PWM), a co za tym idzie, tylko raz wykonać obliczenia sieci neuronowej. Przy ogólnosiłowym nastawieniu na jak najbardziej energooszczędne sterowania (mikroprocesor pobiera o wiele mniej energii kiedy nie liczy) jest to bardzo ważna cecha.

W pierwszym rozdziale umieszczono założenia odnośnie sieci neuronowej oraz symulacji. W drugim rozdziale zamieszczono informacje odnośnie obiektu sterowania. Rozdziały trzeci i czwarty przedstawiają wyniki symulacyjne, które pokazują w jaki sposób częstotliwość wykonywania obliczeń wpływa na jakość regulacji, dla dwóch różnych sygnałów sterujących.

1. Założenia odnośnie sieci neuronowej i symulacji

Przy projektowaniu sieci neuronowej zostały nałożone ograniczenia na ilość neuronów w warstwie ukrytej oraz liczbę sygnałów opóźnień wejścia do sieci. Symulacje były przeprowadzane dla 3 neuronów (2 w warstwie ukrytej, 1 w wyjściowej), funkcję aktywacji użyto: *logsig* dla warstwy ukrytej oraz *tansig* dla warstwy wyjściowej. Przyjęto brak wejścia *bias* na każdy z neuronów. Sieć jest typu *feedforward*.

Sieć jest uczona metodą adaptacyjnych interakcji. Wzory opisujące algorytm uczenia mają postacie:

a) dla warstwy wyjściowej:

$$\dot{w}_i = \gamma \cdot e_i \cdot r_i \quad (1)$$

gdzie: w – waga, i – numer wagi w warstwie wyjściowej, γ – współczynnik adaptacji, e_i – sygnał wejściowy do sieci, bez opóźnień, r_i – sygnał wyjściowy z neuronu warstwy ukrytej do i -tej wagi w warstwie wyjściowej,

b) dla warstwy ukrytej:

$$\dot{w}_j = r_j \cdot \Phi_i \cdot f(-p_k) \quad (2)$$

gdzie: j – numer wagi w warstwie wejściowej, k – numer neuronu warstwy ukrytej, r_j – sygnał doprowadzony do wagi j , p_k – suma sygnałów k -tego neuronu, przed podaniem na funkcję aktywacji,

sygnałów k -tego neuronu, przed podaniem na funkcję aktywacji,

$$\Phi_i = \dot{w}_i \cdot w_i \quad (3)$$

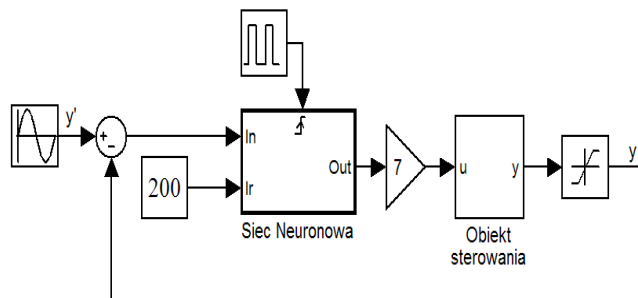
Początkowe wagi sieci zostały dobrane metodą losową, z uwzględnieniem zależności:

$$\text{if } (\text{sign}(w_a) == \text{sign}(w_b)) \Rightarrow \text{sign}(w_a) = -\text{sign}(w_b) \quad (4)$$

gdzie: a oraz b są to odpowiadające sobie neurony ze względu na opóźnienie sygnału. Zastosowanie dla przynajmniej dwóch par neuronów w warstwie ukrytej.

Nowością jeżeli chodzi o przyjęte założenia (w porównaniu do [2]) jest wprowadzenie czasu próbkowania. Co określony czas T_p następuje proces obliczenia wartości wyjściowej sieci oraz zmiany parametrów wartości wag poszczególnych neuronów. W okresie między T_p , wartość wyjściowa z sieci jest zatrzymywana i równa ostatniemu obliczeniu.

Wszystkie symulacje zostały przeprowadzone w programie Matlab/Simulink. Krok, z którym zostały one wykonane to $1 \cdot 10^{-6}$ s (niezmienny podczas całej symulacji). Użyta metoda całkowania: *ode3*. Schemat układu do symulacji został przedstawiony na rys. 1. Na rysunku widać generator funkcji prostokątnej. Na zbczce narastające fali prostokątnej są wykonywane wszystkie obliczenia (wartości wyjściowej oraz adaptacji). Zastosowany trigger narzuca wewnątrz bloku Sieć Neuronowa osobny krok, synchronizując z nim obliczanie wszystkich wewnętrznych całek. Całki obliczane są metodą trapezów, którą w prosty sposób można zaimplementować na mikrokontrolerze. Wzmocnienie służy do rozciągnięcia zakresu sygnału wyjściowego z sieci (funkcją *tansig* generuje wartości z zakresu $<-1, 1>$). Struktura sieci neuronowej (bez sprzężeń zwrotnych wewnątrz sieci oraz bez użycia członu różniczkującego) przypomina regulator PI działający w układzie z ujemnym sprzężeniem zwrotnym.



Rys. 1. Schemat blokowy do symulacji sieci z badanym obiektem

2. Obiekt sterowania

Obiekt, który został poddany sterowaniu przez sieć neuronową można opisać za pomocą następujących równań stanu (pominięto indeksy zależności od czasu):

$$\begin{cases} \dot{x} = x \cdot e^{-0.5x} + (1 + x^2 + 5 \cdot v) \cdot u \\ y = x \end{cases} \quad (5)$$

gdzie: $v \sim N(0, 1)$.

Obiekt z punktu widzenia sieci neuronowej jest czarną skrzynką, tj. jedynymi znanymi sygnałami obiektu przyjęto jego wejście oraz wyjście.

3. Wyniki symulacyjne dla sygnału $y = 3 \cdot \sin(x)$

Dla wylosowanych wag w sieci neuronowej przeprowadzono symulację, w której sygnałem sterującym był: $y = 3 \cdot \sin(x)$. Na rys. 2 przedstawiono wybrane przebiegi dla tego doświadczenia. Linia ciągłą został przedstawiony przebieg odpowiedzi z obiektu, linią przerywaną – sygnał zadany.

Tabela 1. Wpływ czasu T_p na jakość sterowania

Odstęp czasowy pomiędzy obliczeniami [s]	$\frac{1}{N} \sum_{n=1}^N (y'_i - y_i)^2$
$> 6,55 \cdot 10^{-3}$	> 20 – brak stabilności
$6,55 \cdot 10^{-3}$	0.0479
$6 \cdot 10^{-3}$	0.0370
$5 \cdot 10^{-3}$	0.0101
$3,5 \cdot 10^{-3}$	0.0067
$2,5 \cdot 10^{-3}$	0.0053
$1 \cdot 10^{-3}$	0.0039
$1 \cdot 10^{-4}$	0.0032
$1 \cdot 10^{-5}$	0.0031
$1 \cdot 10^{-6}$	0.0031

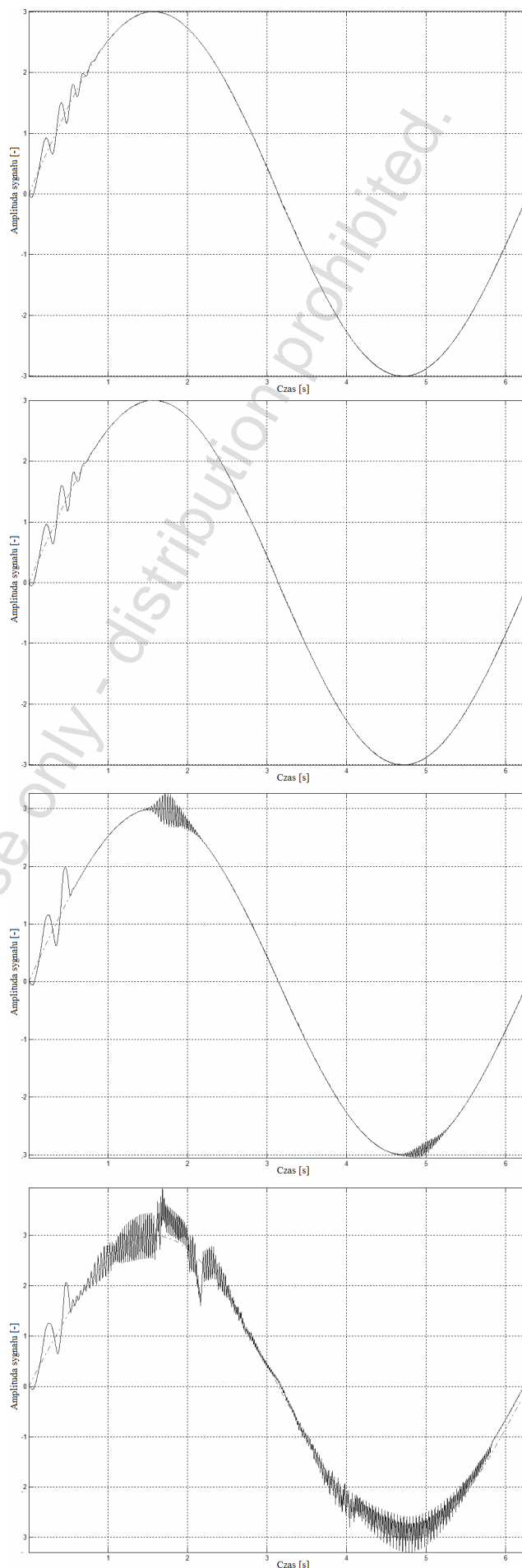
Analizując zawartość tabeli 1 można stwierdzić dwie rzeczy: zbyttnia minimalizacja czasu obliczeń sieci nie robi większej różnicy w kontekście jakości sterowania. Przy porównaniu wartości i przebiegów dla $1 \cdot 10^{-6}$ s oraz $1 \cdot 10^{-3}$ s można stwierdzić, że nie następują żadne znaczące zmiany. Sieć neuronowa po dobraniu odpowiednich wag prawidłowo steruje obiektem. Drugi wniosek: istnieje pewna granica czasu obliczeń wartości wyjściowej z sieci, poniżej której układ staje się niestabilny. Dla opisanego tutaj obiektu oraz wylosowanych wag, ta granica znajduje się około $6,55 \cdot 10^{-3}$ [s]. Na wykresie dla tego czasu obliczeń odpowiedzi na rys. 2 można zauważyć, iż sieć posiada duże trudności z dopasowaniem wyjścia z obiektu do sygnału zadanego. Niewielkie zmniejszenie tej wartości, np. do $6,56 \cdot 10^{-3}$ s powoduje niestabilną pracę układu.

Im bardziej parametr T_p zbliża się do wartości $6,55 \cdot 10^{-3}$ s, tym bardziej jakość sterowania ulega pogorszeniu, jednakże układ jest w stanie nadążać za sygnałem referencyjnym o zadanym kształcie.

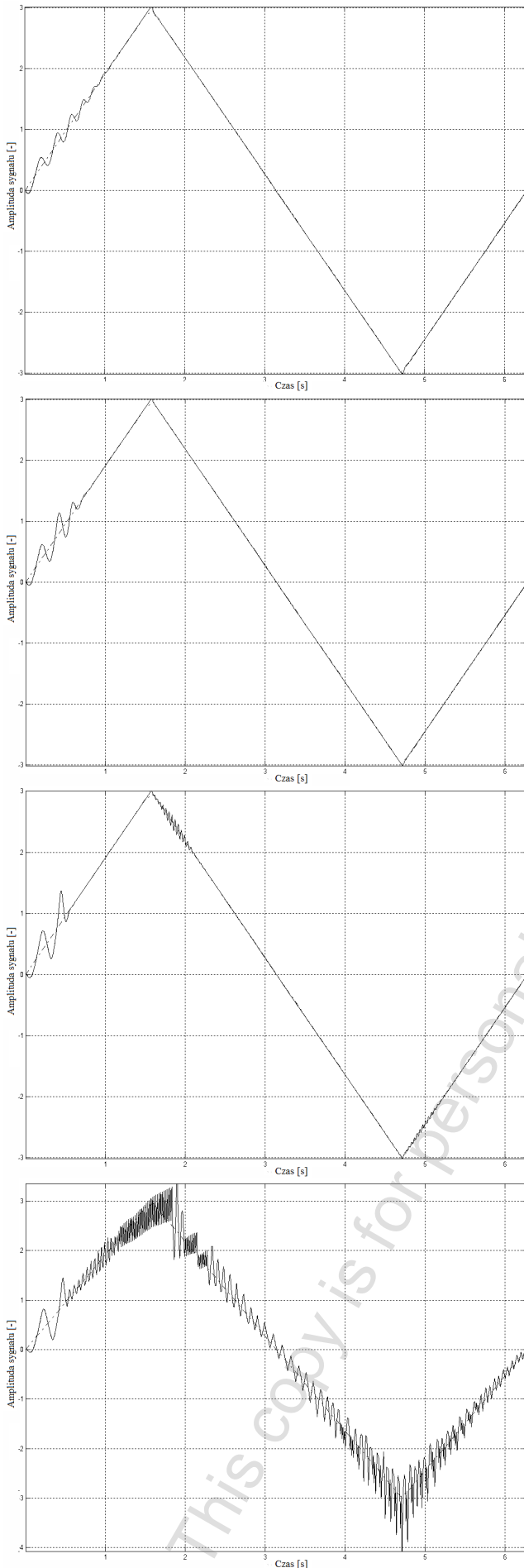
4. Wyniki symulacyjne dla sygnału $y = 3 \cdot \text{tri}(x)$

Podobnie jak powyżej, dla wylosowanych wag w sieci neuronowej przeprowadzono symulację, w której sygnałem sterującym był: $y = 3 \cdot \text{tri}(x)$. Na rys. 3 przedstawiono wybrane przebiegi dla tego doświadczenia. Linia ciągłą został przedstawiony przebieg odpowiedzi z obiektu, linią przerywaną – sygnał zadany.

Odstęp czasowy pomiędzy obliczeniami [s]	$\frac{1}{N} \sum_{n=1}^N (y'_i - y_i)^2$
$1,5 \cdot 10^{-2}$	> 20
$1 \cdot 10^{-2}$	0,0455
$8 \cdot 10^{-3}$	0,0333
$7 \cdot 10^{-3}$	0,0044
$5 \cdot 10^{-3}$	0,0032
$2,5 \cdot 10^{-3}$	0,0023
$1 \cdot 10^{-3}$	0,0017
$1 \cdot 10^{-4}$	0,0013
$1 \cdot 10^{-5}$	0,0012
$1 \cdot 10^{-6}$	0,0012



Rys. 2. Przebieg odpowiedzi obiektu dla czasu obliczeń odpowiedzi sieci na poziomie (od góry): $1 \cdot 10^{-6}$, $1 \cdot 10^{-3}$, $5 \cdot 10^{-3}$, $6,650 \cdot 10^{-3}$ [s]



Rys.3. Przebieg odpowiedzi obiektu dla czasu obliczeń odpowiedzi sieci na poziomie (od góry): $1 \cdot 10^{-6}$, $3,5 \cdot 10^{-3}$, $7 \cdot 10^{-3}$, $1 \cdot 10^{-2}$ [s]

Porównując wyniki do sygnału sinusoidalnego można stwierdzić, że użyteczny zakres czasu odpowiedzi sieci neuronowej zależy bardzo mocno od sygnału, za którym sieć ma nadążyć. Jeżeli sygnał jest odcinkowo liniowy, sieć bardzo dobrze radzi sobie ze sterowaniem obiektu nawet dla czasów większych niż $7 \cdot 10^{-3}$ s. Granica, przy której cały układ staje się niestabilny w przypadku sygnału trójkątnego wynosi około $1,5 \cdot 10^{-2}$ s.

5. Wpływ wartości lr na jakość nadążania przy zwiększonym czasie odpowiedzi sterownika

W omawianych przypadkach wartość lr była ustalona na 200. Dzięki takiej wartości, po dopasowaniu wag do obiektu, sygnał odpowiedzi był bardzo zbliżony do sygnału referencyjnego. Zauważono zależność tej wartości z maksymalnym czasem odpowiedzi neurosterownika.

Aby wykonać porównanie wartości lr zostały wylosowane nowe wagi dla sieci neuronowej. Niezależnie od sygnału sterującego, wartość pomiędzy odpowiedzią sieci na poziomie $1,5 \cdot 10^{-2}$ s powodowała niestabilność układu. Dla takich warunków zmieniano parametr lr , a wyniki tych zmian zostały przedstawione w tabeli 3. Wyniki wygenerowano dla dwóch okresów sygnału, graficzna reprezentacja została przedstawiona na rys. 4. Na rys 5 został pokazany sygnał sterujący obiektem dla różnych wartości parametrów lr .

Tabela 3. Wpływ lr na jakość sterowania

lr	$\frac{1}{N} \sum_{n=1}^N (y'_i - y_i)^2$ $y = 3 \cdot \sin(x)$	$\frac{1}{N} \sum_{n=1}^N (y'_i - y_i)^2$ $y = 3 \cdot \text{tri}(x)$
200	>41	>35
100	0,1532	0,0907
75	0,1693	0,0056
50	0,0120	0,0065
10	0,0342	0,0227
1	0,2329	0,2079
<0.1	Niestabilność po pierwszej ćwiartce pierwszego okresu, >400	

Z rys. 4 oraz rys. 5 wyciągnięto następujące wnioski:

- istnieje stan, z którego sieć nie może powrócić do nadążania za sygnałem referencyjnym ($lr = 200$), gdy tak się stanie, sygnał sterujący przyjmuje wartość równą 0,
- istnieje jedna wartość parametru lr , dla której można znaleźć sygnał sterujący, zapewniający minimalizację wskaźnika $\frac{1}{N} \sum_{n=1}^N (y'_i - y_i)^2$ Przyпуска się, iż wartość ta będzie zmienną

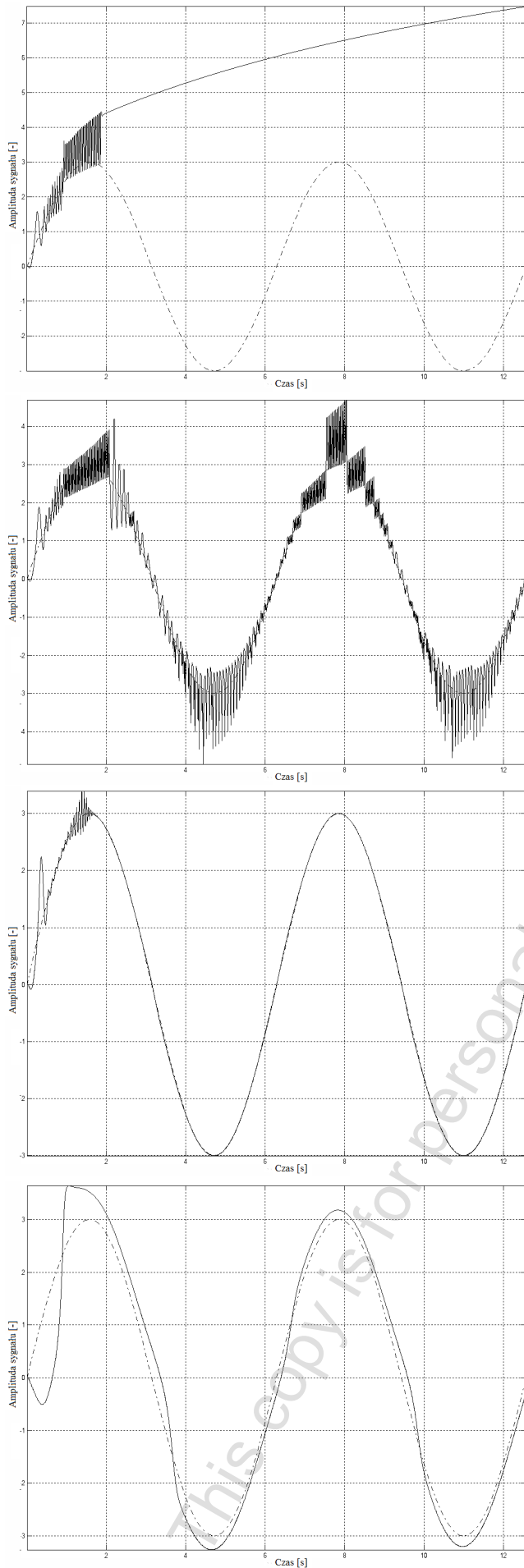
między innymi w kontekście czasu odpowiedzi sieci. Będzie można znaleźć funkcję $lr(T_p, y)$ dla zapewnienia prawie idealnego odwzorowania sygnału wyjściowego z obiektu,

- należy zwrócić uwagę na kształt sygnału sterującego dla $lr = 50$ oraz $lr = 100$ – w drugim przypadku sygnał sterujący przybiera postać sygnału z regulatora on/off, co nie wpływa pozytywnie na jakość regulacji.

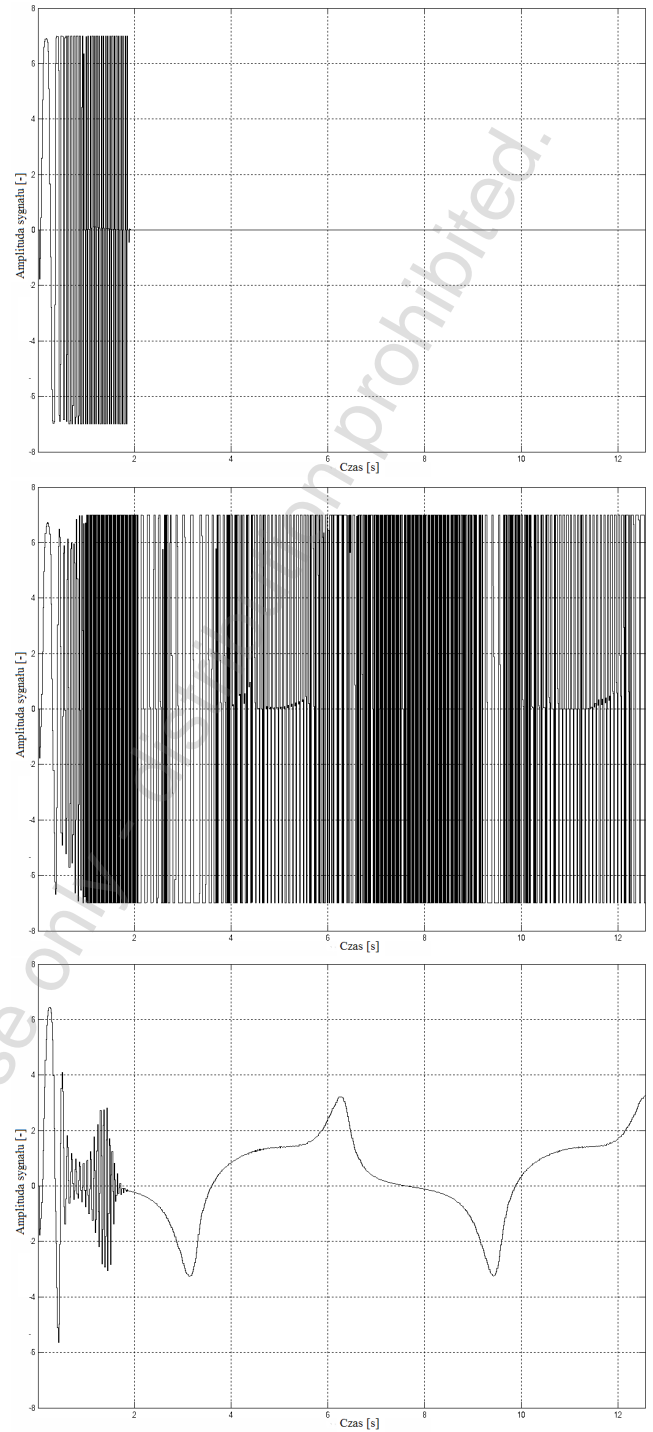
6. Podsumowanie

Zaproponowano neurosterownik, generujący taki sygnał sterujący, że obiekt nadąża za sygnałem referencyjnym. Pokazano wpływ dwóch parametrów na jakość regulacji: czas pomiędzy kolejnym obliczeniem wartości wyjściowej z sieci T_p oraz lr oraz opisano ich powiązanie między sobą.

Dalsze badania będą skupiać się na znalezieniu sposobu obliczania parametru lr , wzorem lub za pomocą mechanizmu adaptacyjnego i jego wpływu na zachowanie się neurosterownika. Oprócz tego, zagadnieniem wymagającym analizy jest użycie neurosterownika do generacji sygnału sterującego, gdy sygnał referencyjny posiada pewne przesunięcie bias.



Rys. 4. Przebieg odpowiedzi obiektu dla wartości lr (od góry): 200, 100, 50, 1



Rys. 5. Sygnał sterujący obiektem dla wartości lr (od góry): 200, 100, 50

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] Lis M.: *Sieć neuronowa jako regulator obiektu dynamicznego*. Poznań University of Technology Academic Journals: Electrical Engineering, 2012 Issue 71, s. 17 – 24

Marcin Lis

e-mail: mail.dla.studenta@gmail.com

Student studiów III-go stopnia na Politechnice Poznańskiej od 2011 roku. Działalność naukowa obejmuje zastosowanie sieci neuronowych do sterowania obiektami o nieznannej strukturze wewnętrznej, z ukierunkowaniem na sterowanie przekształtnikami energoelektronicznymi.

