

*Local Relative Sensitivity Index pruning,  
neuronowe estymatory zmiennych stanu,  
układ dwumasowy, napęd elektryczny*

Marcin KAMIŃSKI\*

## **ESTYMACJA ZMIENNYCH STANU UKŁADU DWUMASOWEGO ZA POMOCĄ MODELI NEURONOWYCH**

W niniejszym artykule przedmiotem badań są estymatory neuronowe zastosowane w celu odtwarzania zmiennych stanu układu napędowego z połączeniem sprzężystym. Analizie poddano modele estymujące prędkość maszyny roboczej oraz momentu skrętnego napędu. W trakcie doboru współczynników wagowych sieci neuronowych zastosowano metodę *Levenberga–Marquardta*. Ten etap procesu projektowania zmodyfikowano poprzez wprowadzenie redukcji struktury zastosowanych modeli. W tym celu zastosowano współczynnik istotności połączeń – LRSI (*Local Relative Sensitivity Index*). Zaprojektowane estymatory zmiennych stanu charakteryzując się wysoką dokładnością odtwarzania dla szerokiego zakresu zmian zadanej trajektorii prędkości oraz przełączania momentu obciążającego, a także w obecności zmian parametrów napędu. Wyniki badań symulacyjnych zostały potwierdzone w eksperymencie wykonanym na stanowisku laboratoryjnym.

### 1. WSTĘP

W ostatnich latach obserwowana jest rosnąca liczba aplikacji algorytmów sztucznej inteligencji w zastosowaniach naukowo-technicznych. Zjawisko takie jest związane z gwałtownym rozwojem energoelektroniki oraz elektronicznych układów programowalnych, umożliwiających praktyczną implementację analizowanych struktur. Należy podkreślić, że są to najczęściej modele inspirowane logiką rozmytą, oparte na sieciach neuronowych lub wykorzystujące algorytmy genetyczne, zatem wymagające znaczących mocy obliczeniowych. Opisywana tendencja dotyczy również napędu elektrycznego. Aplikowane struktury, których elementarne zasady działania zaczerpnięto z dziedziny sztucznej inteligencji, stosowane są w celu sterowania [1], odtwarzania zmiennych stanu [2], diagnostyki [3], itd.

---

\* Politechnika Wroclawska, Instytut Maszyn, Napędów i Pomiarów Elektrycznych, ul. Smoluchowskiego 19, 50-370 Wrocław, e-mail: marcin.kaminski@pwr.wroc.pl.

W artykule opisywana jest aplikacja, której obiektem wykonawczym jest układ napędowy, w którym połączenie pomiędzy silnikiem a maszyną roboczą jest zrealizowane za pomocą długiego elastycznego wału. Taka konstrukcja stanowiska reprezentuje zjawiska zachodzące w napędach spotykanych w:

- mechanizmach maszyn papierniczych [4],
- manipulatorach stosowanych w medycynie [5],
- napędach robotów [6], [7],
- napędach walcowniczych [8].

Specyficznym zjawiskiem, wynikającym z ograniczonej sztywności połączenia w napędzie, jest występowanie momentu skrętnego, utrudniającego w sposób istotny precyzyjne sterowanie prędkością wału. W efekcie, w układzie pojawiają się oscylacje zmiennych stanu. Jedną z efektywnych metod tłumienia drgań skrętnych układu dwumasowego jest rozwinięcie klasycznych struktur sterowania, poprzez wprowadzenie dodatkowych sprzężeń zwrotnych od zmiennych stanu, które często są trudne do zmierzenia w rzeczywistym układzie [9]. Aby zredukować koszty wykonania napędu oraz zwiększyć jego niezawodność stosowane są układy estymujące w celu uzyskania informacji o niedostępnych zmiennych stanu.

Wśród najpopularniejszych układów stosowanych w odtwarzaniu zmiennych stanu układu napędowego z połączeniem sprężystym należy wymienić: obserwator Luenbergera [10] oraz filtr Kalmana [11]. Zakładając bardzo dokładną znajomość obiektu, jego model matematyczny oraz parametry, można uzyskać za pomocą wspomnianych układów wysoką dokładność odtwarzania. Jednak w rzeczywistych aplikacjach przemysłowych jest to często warunek trudny do spełnienia. Jednym z alternatywnych rozwiązań opracowywanych w ośrodkach naukowych jest zastosowanie modeli neuronowych jako estymatorów zmiennych stanu napędów elektrycznych.

Sieci neuronowe są złożonymi, nieliniowymi strukturami, z (najczęściej) bardzo dużą liczbą parametrów, które są optymalizowane w procesie treningu. Po odpowiednim zaprojektowaniu istnieje możliwość odwzorowywania skomplikowanych zależności pomiędzy sygnałami wejściowymi oraz wyjściowymi modelu. W artykule analizowana jest struktura perceptronu wielowarstwowego (*Multi Layer Perceptron*). Przygotowanie modelu neuronowego (w trybie *off-line*) do realizacji założonego zadania można podzielić na trzy główne etapy:

- zapewnienie odpowiedniego zbioru danych treningowych, reprezentatywnych dla danego zadania,
- dobór struktury sieci neuronowej (należy wyznaczyć liczbę warstw ukrytych, a następnie określić liczbę neuronów w poszczególnych warstwach ukrytych, typy funkcji aktywacji neuronów).
- ostatni etap jest związany z algorytmem optymalizacyjnym zastosowanym do wyznaczenia współczynników sieci neuronowej (występuje konieczność wybo-ru metody treningu oraz jego parametrów).

Jednym z najbardziej problematycznych zagadnień wpływających w sposób istotny na realizację zadania, jest dobór struktury sieci neuronowej. Optymalizacja struktury sieci neuronowej jest jednym z działań mających na celu poprawę właściwości generalizacyjnych modelu. Wśród stosowanych rozwiązań należy wymienić:

- algorytmy konstruktywistyczne,
- oraz metody *pruningu*.

Algorytmy konstruktywistyczne zakładają w fazie początkowej treningu minimalną strukturę, która jest następnie rozbudowywana według określonego kryterium [12], [13]. Odminną koncepcją jest założenie relatywnie dużej struktury początkowej sieci, a następnie usuwanie połączeń międzyneuronowych. Generalnie, takie założenie jest również bardzo korzystne dla praktycznej implementacji modelu w procesorze sygnałowym lub matrycy FPGA. Redukcja struktury modelu neuronowego, oznacza uproszczenie (zmniejszenie liczby elementarnych) obliczeń. Zatem układ programowalny może wykonywać obliczenia z większą częstotliwością.

Metody optymalizacji struktury sieci neuronowych, mające na celu eliminację poszczególnych wag, koncentrują się głównie na wskazaniu połączeń zbędnych. Najprostszym sposobem jest analiza wartości bezwzględnych wag sieci [14]. Do najsukuczniejszych metod zalicza się algorytmy wrażliwościowe, w których badana jest czułość funkcji celu na usunięcie poszczególnych połączeń ze struktury sieci neuronowej. Najistotniejsze metody w tej grupie to *Optimal Brain Damage* [15] oraz *Optimal Brain Surgeon* [16], [17]. Charakterystyczną cechą wyznaczania współczynników istotności metod OBD oraz OBS jest obliczanie macierzy drugich pochodnych funkcji celu względem współczynników wagowych. W przypadku dużej liczby analizowanych połączeń sieci jest to czynność czasochłonna oraz wymaga znacznej mocy obliczeniowej. Wśród metod zmniejszających strukturę sieci neuronowych znajdują również zastosowanie algorytmy genetyczne [18].

W niniejszym artykule analizowana jest jedna z metod mających na celu redukcję struktury modelu neuronowego, wykorzystująca współczynnik *Local Relative Sensitivity Index*. Jedną z podstawowych zalet związanych z obliczeniami parametru LRSI dla poszczególnych połączeń sieci neuronowej, w porównaniu z algorytmami OBD oraz OBS, jest wykorzystanie jedynie pierwszej pochodnej funkcji celu względem wartości wag [19]. Współczynnik LRSI stanowi rozwinięcie idei przedstawionej w artykule [20], gdzie zdefiniowano współczynnik istotności *Sensitivity Value*, który bazuje na wrażliwości funkcji celu na zmiany wartości wag oraz uwzględnia stopień zmiany wagi dla danego połączenia sieci (w efekcie procesu treningu). Współczynnik SV uwzględnia jedynie globalne znaczenie danej wagi sieci neuronowej, natomiast LRSI zakłada również że połączenie może być istotne dla przetwarzania w fragmencie sieci neuronowej. W analizowanym przypadku zastosowano dwie sieci neuronowe do estymacji poszczególnych zmiennych stanu, zatem każdy model posiadał jedno wyjście. Wobec problemu z decyzją o liczbie usuwanych połączeń, przyjęto założenie zakładające usuwanie jednego połączenia w warstwie wyjściowej. Takie podejście

eliminuje wspomniany problem, ponadto eliminacja wagi akurat w tej warstwie prowadzi do wyłączenia znacznie większej części struktury, związanej z obliczeniami wejściowymi dla usuniętego neuronu drugiej warstwy ukrytej. W treningu zastosowano algorytm *Levenberga-Marquardta*.

Przedstawiony artykuł składa się z sześciu rozdziałów. W pierwszym z nich zaprezentowano ogólnie problematykę związaną ze sterowaniem napędami z połączeniem elastycznym oraz neuronowymi estymatorami zmiennych stanu. Następnie przedstawiono model matematyczny układu sterowania oraz obiektu. W dalszej części przedstawiono strukturę estymatorów neuronowych oraz algorytmy zastosowane w ich projektowaniu, w tym algorytm umożliwiający redukcję struktury zastosowanego modelu. Zaprojektowane estymatory prędkości obciążenia oraz momentu skrętnego zostały poddane testom symulacyjnym a także eksperymentalnym. Pracę kończy krótkie podsumowanie.

## 2. OPIS MATEMATYCZNY OBIEKTU ORAZ STRUKTURY STEROWANIA

Efektywna analiza struktur sterowania lub estymatorów zmiennych stanu napędów elektrycznych wymaga wyboru właściwego modelu układu wykonawczego. W literaturze przedstawiono kilka modeli układu dwumasowego, wśród których można wybrać ten, który wystarczająco dobrze opisuje obiekt i badane zjawisko, a jednocześnie zapewni możliwie prosty opis matematyczny przydatny do dalszej analizy. Do modelowania układu napędowego, w opisywanym przypadku zastosowano model z bezinercyjnym połączeniem sprzężystym opisany równaniami [9]:

$$\begin{aligned} T_1 \frac{d\omega_1(t)}{dt} &= m_e(t) - m_s(t), \\ T_2 \frac{d\omega_2(t)}{dt} &= m_s(t) - m_L(t), \\ T_c \frac{dm_s(t)}{dt} &= \omega_1(t) - \omega_2(t) \end{aligned} \quad (1)-(3)$$

gdzie:

- $T_1$  – mechaniczna stała czasowa silnika,
- $T_2$  – mechaniczna stała czasowa obciążenia,
- $T_c$  – stała czasowa elementu sprzężystego,
- $\omega_1$  – prędkość silnika,
- $\omega_2$  – prędkość maszyny roboczej,
- $m_e$  – moment elektromagnetyczny silnika,
- $m_s$  – moment skrętny,
- $m_L$  – moment obciążenia.



### 3. ESTYMATORY NEURONOWE

Opisywane w artykule estymatory zostały oparte o sieci neuronowe MLP, które stanowią kombinację elementarnych neuronów w strukturę przetwarzającą dane równolegle. Dane są przesyłane jednokierunkowo od wejść do wyjścia, bez sprzężeń zwrotnych [13]. Współczynniki skalujące dla każdego z połączeń dobierane są w procesie treningu. Adaptacja poszczególnych wartości wagowych  $w_{ij}$  odbywa się w kolejnych iteracjach, zatem można zapisać:

$$w_{ij}(k+1) = w_{ij}(k) + \Delta w_{ij}(k). \quad (9)$$

Obliczenia, w poszczególnych epokach treningu, na wartości wag jest realizowane w celu zmniejszania różnic między wartościami wyjściowymi sieci neuronowej a elementami wzorcowymi. Matematycznie można takie działanie przedstawić jako minimalizację funkcji celu, która najczęściej jest definiowana jako błąd średniokwadratowy między wartością estymowaną a rzeczywistą. W przypadku jednej pary uczącej przyjmuje następującą postać:

$$E = \frac{1}{2} \sum_{j=1}^M (y_j(k) - d_j(k))^2 \quad (10)$$

gdzie:

$y_j$  – artość wyjściowa sieci neuronowej,

$d_j$  – wartość wzorcowa.

Ciągłość i różniczkowalność funkcji celu umożliwia zastosowanie metod gradientowych do ich minimalizacji. Punktem wyjścia w takich metodach jest rozwinięcie funkcji celu w szereg Taylora w otoczeniu aktualnego rozwiązania:

$$\Delta E = \sum_i g_i \Delta w_i + \frac{1}{2} \left[ \sum_i h_{ii} [\Delta w_{ii}]^2 + \sum_{i \neq j} h_{ij} \Delta w_i \Delta w_j \right] + O(\|\Delta w\|^3), \quad (11)$$

gdzie:

$\Delta w_i$  – zmiany  $i$ -tej wagi,

$$g_i = \frac{\partial E}{\partial w_i},$$

$$h_{ij} = \frac{\partial^2 E}{\partial w_i \partial w_j}.$$

Powyższy zapis prezentuje zależność zamian funkcji celu od pierwszych oraz drugich pochodnych funkcji celu, obliczanych względem zmian współczynników wagowych. Można również zauważyć, że w celu wprowadzenia poprawki współczynników

wagowych warto skorzystać z macierzy gradientu, natomiast dodatkową dokładność obliczeń wprowadza uwzględnienie macierzy hesjanu pochodnych cząstkowych. Jednym z najpopularniejszych oraz najefektywniejszych gradientowych algorytmów minimalizacyjnych jest metoda *Levenberga-Marquardta*. Wykorzystanie algorytmu, bazującego na macierzy hesjanu w opisywanej aplikacji ma istotne znaczenie dla dokładności otrzymywanych wyników [21]. Korekcja wag sieci neuronowej zgodnie z metodą LM jest opisana równaniem:

$$\Delta \mathbf{w} = -(\mathbf{J}(\mathbf{w})^T \mathbf{J}(\mathbf{w}) + \mu \mathbf{I})^{-1} \mathbf{J}^T(\mathbf{w}) \mathbf{e}(\mathbf{w}), \quad (12)$$

gdzie  $\mu$  – współczynnik uczenia.

W powyższym równaniu macierz  $\mathbf{J}$  jest zdefiniowana następująco:

$$\mathbf{J}(\mathbf{w}) = \begin{bmatrix} \frac{\partial e_1}{\partial w_1} & \frac{\partial e_1}{\partial w_2} & \dots & \frac{\partial e_1}{\partial w_n} \\ \frac{\partial e_2}{\partial w_1} & \frac{\partial e_2}{\partial w_2} & \dots & \frac{\partial e_2}{\partial w_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial e_M}{\partial w_1} & \frac{\partial e_M}{\partial w_2} & \dots & \frac{\partial e_M}{\partial w_n} \end{bmatrix} \quad (13)$$

gdzie  $\mathbf{e}(\mathbf{w}) = \begin{bmatrix} e_1(\mathbf{w}) \\ e_2(\mathbf{w}) \\ \vdots \\ e_M(\mathbf{w}) \end{bmatrix}$

oraz  $e_j(\mathbf{w}) = [y_j(\mathbf{w}) - d_j]$ .

W wielu przypadkach w obliczeniach algorytmu LM macierz Hessego jest aproksymowana:

$$\mathbf{H} \cong \mathbf{J}^T(\mathbf{w}) \mathbf{J}(\mathbf{w}). \quad (14)$$

W opisywanej aplikacji sieci neuronowych wprowadzono dodatkowo *pruning*. W celu selekcji mniej istotnych wag zastosowano współczynnik LRSI. Metoda ta jest rozwinięciem algorytmu zaproponowanego w [20], gdzie współczynnik istotności zdefiniowano następująco:

$$SV_{ij} = - \sum_0^{N-1} \left[ \frac{\partial E}{\partial w_{ij}}(k) \Delta w_{ij}(k) \right] \frac{w_{ij}^f}{w_{ij}^f - w_{ij}^i}, \quad (15)$$

gdzie:

- $w_{ij}^f$  – wartość współczynnika  $w_{ij}$  po treningu,  
 $w_{ij}^i$  – wartość współczynnika  $w_{ij}$  przed treningiem,  
 $N$  – liczba epok,  
 $k$  – numer iteracji.

Jednak w celu zwiększenia skuteczności poprzez uwzględnienie lokalnych oddziaływań współczynników wagowych, wprowadzono modyfikację [19]:

$$LRSI_{ij} = \frac{|SV_{ij}|}{\sum_{m=1}^n |SV_{im}|} \quad (16)$$

gdzie  $n$ -liczba połączeń wejściowych danego neuronu.

Poszczególne etapy postępowania w opisywanym algorytmie *pruningu* można przedstawić następująco:

1. Opracowanie danych treningowych, spośród których należy wyznaczyć zbiór walidacyjny.
2. Przyjęcie początkowej struktury sieci oraz wstępny trening.
3. Obliczenia parametrów SV dla każdego z połączeń sieci neuronowej.
4. Wyznaczenie współczynników LRSI dla poszczególnych połączeń zastosowanego modelu.
5. Usuwanie połączeń w oparciu o LRSI. Eliminuje się określoną liczbę połączeń lub wagi dla których współczynnik LRSI jest poniżej założonej wartości. Ponadto, zakłada się usuwanie maksymalnie jednego wejścia dla poszczególnego neuronu.
6. Realizowane jest douczanie sieci neuronowej.
7. Następuje porównanie wyników – ewentualne zakończenie algorytmu lub skok do punktu 3.

Jednak w opisywanej w artykule implementacji sieci neuronowych wprowadzono modyfikację do opisanej powyżej metody. Trening, zgodnie z opisaną metodą (LM), dla każdego z estymatorów został wykonany dwukrotnie. Przed drugą częścią obliczeń wyeliminowano część sieci neuronowej, usuwając połączenie wejściowe neuronu wyjściowego, w oparciu o współczynnik LRSI. W ten sposób zredukowano część sieci neuronowej, a jednocześnie wyeliminowano konieczność podjęcia decyzji o liczbie usuwanych połączeń lub poziomu wartości współczynnika LRSI decydującego o eliminacji połączenia sieci oraz uproszczono obliczenia.

#### 4. BADANIA SYMULACYJNE

Estymatory neuronowe zaprojektowane według procedury opisanej w poprzednich częściach artykułu poddano badaniom symulacyjnym. Obliczenia wykonano w pro-

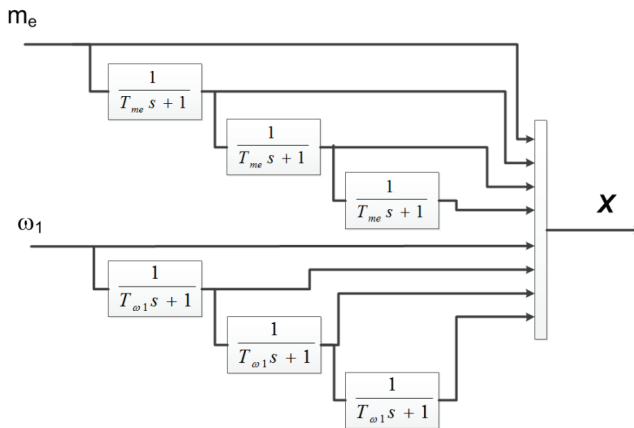


gramie Matlab\Simulink. Analizowano sieci neuronowe odtwarzające prędkość obciążenia oraz moment skrętny. Dla każdej z estymowanych zmiennych stanu zastosowano sieć o strukturze początkowej:

$$NN = \{8-7-8-1\}. \quad (17)$$

W powyższym oznaczeniu kolejne wartości liczbowe opisują kolejno złożoność (liczbę neuronów): wektora wejściowego, pierwszej warstwy ukrytej, drugiej warstwy ukrytej, warstwy wyjściowej.

Jako elementy wektora wejściowego  $X$  sieci neuronowych zastosowano mierzalne zmienne stanu, które najczęściej zakładane są jako dostępne w strukturach sterowania napędu z połączeniem sprzężystym. Wykorzystano moment elektromagnetyczny oraz prędkość silnika napędowego. Zatem można stwierdzić, że wybór sygnałów wejściowych został podyktowany praktycznymi aspektami opisywanej aplikacji. Oddzielnym zagadnieniem jest decyzja dotycząca postaci wektora wejściowego. W pracy zastosowano statyczne modele neuronowe nie zawierające dodatkowych, wewnętrznych sprzężeń zwrotnych, zatem nie uwzględniające w trakcie przetwarzania dynamiki sygnałów. Należy jednak zauważyć, że odtwarzane są dynamicznie zmieniające się w czasie sygnały. W związku z tym, w celu poprawienia dokładności odtwarzania zmiennych stanu, w wielu przypadkach wprowadza się historyczne próbki sygnałów wejściowych. Dzięki temu, sztucznie zwiększa się wrażliwość estymatora na szybkość zmian sygnałów wejściowych. Postać wektora wejściowego estymatorów obu zmiennych stanu przedstawiono na rysunku 2.



Rys. 2. Wektor wejściowy estymatorów zmiennych stanu

Opóźnienia pomiędzy poszczególnymi próbkami sygnałów wejściowych wprowadzane są za pomocą filtrów dolnoprzepustowych. W ten sposób zapewnione jest dodatkowe filtrowanie sygnałów pomiarowych oraz wprowadzana stała wartość opóźnienia

pomiędzy danymi wejściowymi. Wartości stałych czasowych dobrano eksperymentalnie:  $T_{me} = 5$  ms,  $T_{ol} = 1$  ms.

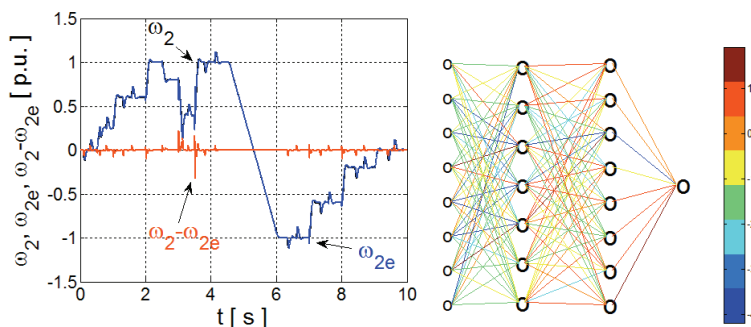
W celu oceny jakości odtwarzania zmiennych stanu za pomocą testowanych estymatorów neuronowych wyznaczano jest błąd zdefiniowany następująco:

$$Err = \frac{\sum_{i=1}^p |x_i - x_i^e|}{p} \cdot 100, \quad (18)$$

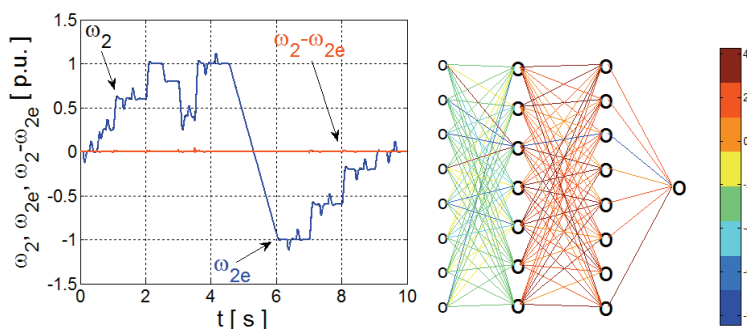
gdzie:

- $x_i$  – wartość rzeczywista,
- $x_i^e$  – wartość estymowana,
- $p$  – liczba próbek przebiegu.

a)b)



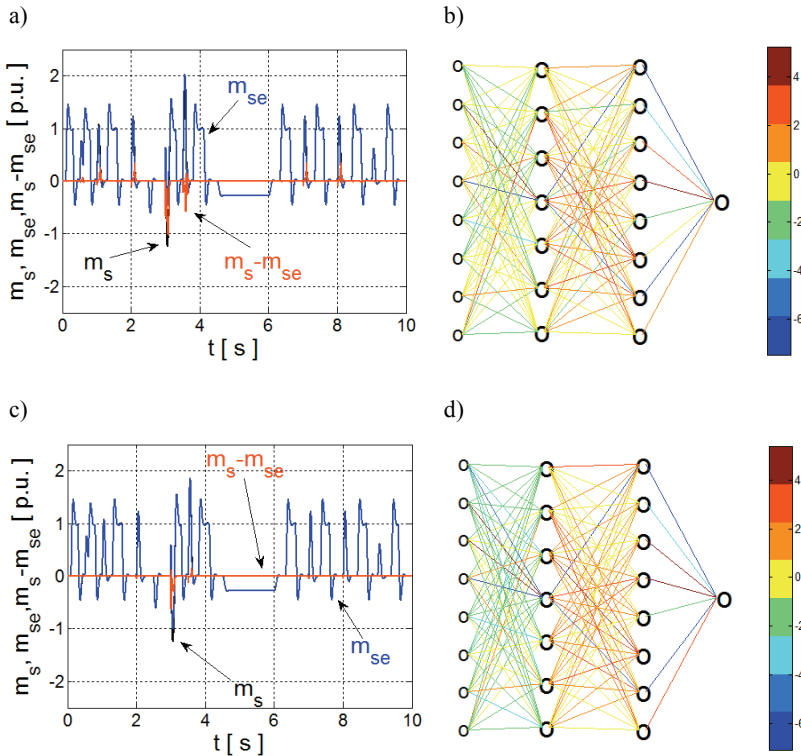
c)d)



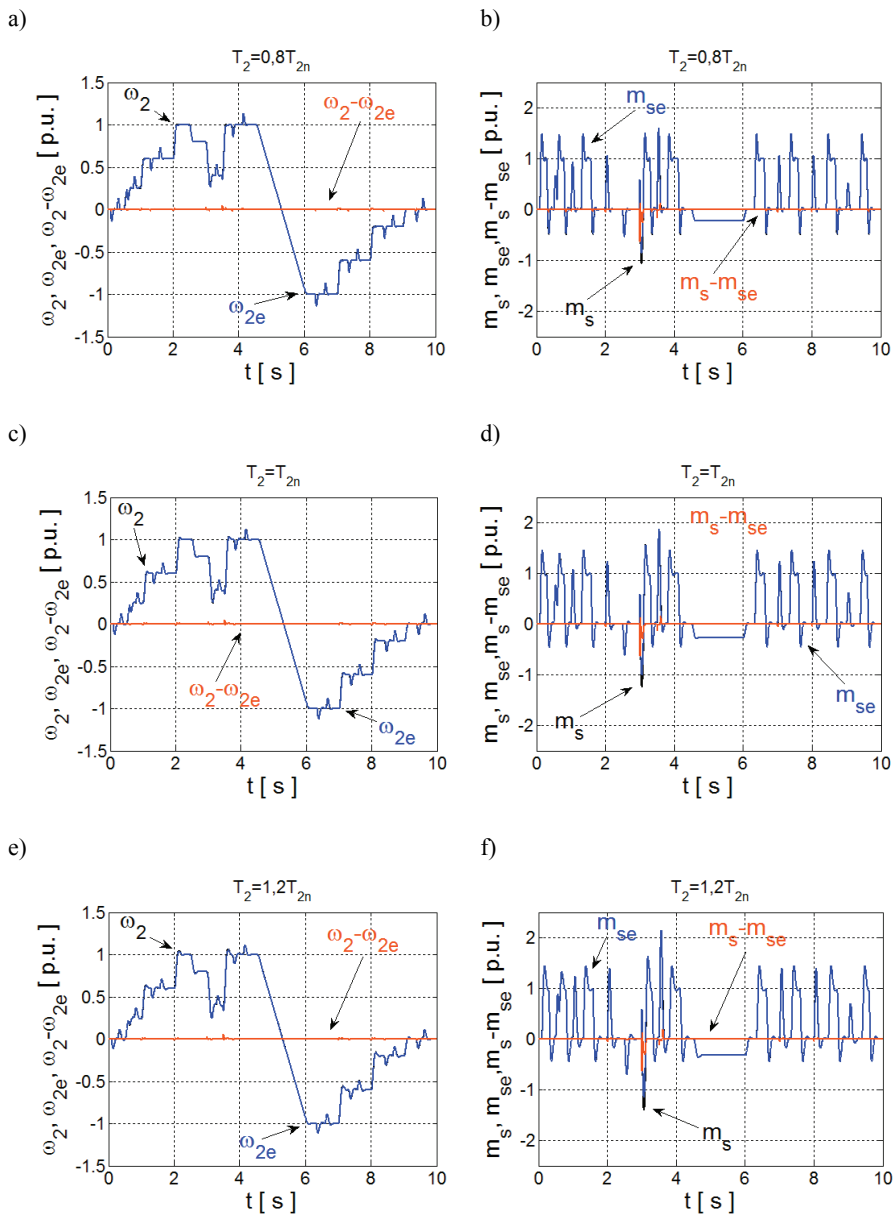
Rys. 3. Przebiegi rzeczywiste, estymowane oraz różnica wartości chwilowych prędkości obciążenia (a,c) oraz struktura zastosowanej sieci (b,d) - przed (a, b) oraz po (c, d) zastosowaniu algorytmu LRSI

Wstępne testy prezentują wpływ zastosowania redukcji struktury na dokładność odtwarzania obu zmiennych stanu. W tym celu wykonano badania, przedstawiające

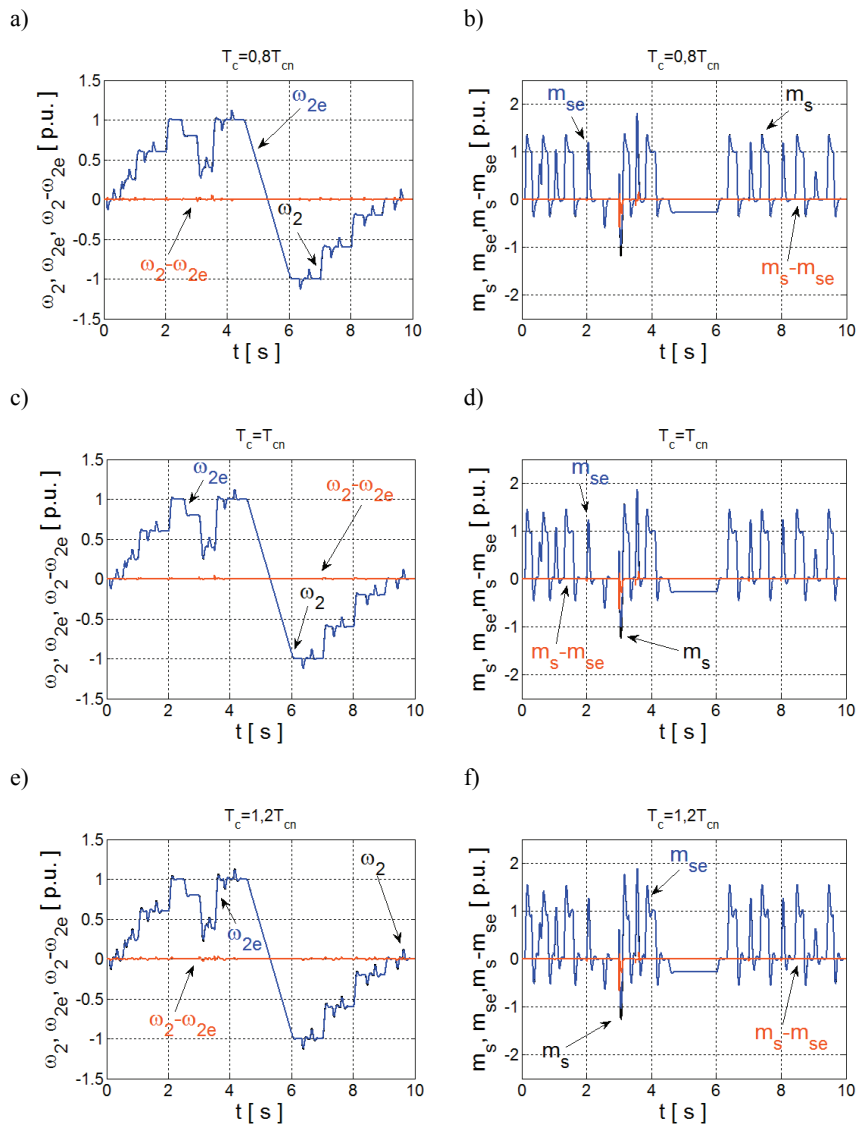
rezultaty estymacji obu zmiennych stanu przed usunięciem części struktury sieci neuronowej (po pierwszym treningu modelu) oraz po zastosowaniu *pruningu* (po dodatkowym douczaniu sieci). Wybrane wyniki badań zamieszczono na rysunku 3 oraz 4. Estymatory neuronowe testowano zbiorem danych nie wykorzystywanych w procesie treningu. Usunięcie jednego połączenia w wyjściowej części sieci neuronowej powoduje wyeliminowanie całego neuronu drugiej warstwy ukrytej. Należy podkreślić, że w ten sposób usunięcie jednej wagi w rzeczywistości pozwala na zredukowanie większej liczby obliczeń (w części wejściowej eliminowanego neuronu oraz funkcji aktywacji). Po powtórnym procesie treningu uzyskano zwiększenie dokładności estymacji obu zmiennych stanu, co jest obserwowalne szczególnie w stanach przejściowych. Błąd odtwarzania prędkości obciążenia przed usunięciem wybranego połączenia sieci wynosił  $Err\omega_2 = 7,86$ , natomiast po końcowych obliczeniach  $Err\omega_2 = 1,57$ . Dla momentu skrotnego uzyskano zmniejszenie wartości błędu z  $Errm_s = 16,96$  do wartości  $Errm_s = 3,60$ .



Rys. 4. Przebiegi rzeczywiste, estymowane oraz różnica wartości chwilowych momentu skrotnego (a, c) oraz struktura zastosowanej sieci (b, d) – przed (a, b) oraz po (c, d) zastosowaniu algorytmu LRSI



Rys. 5. Przebiegi rzeczywiste, estymowane oraz różnica wartości chwilowych prędkości obciążenia (a, c, e) oraz momentu skrętnego (b, d, f) dla różnych wartości parametru  $T_2$



Rys. 6. Przebiegi rzeczywiste, estymowane oraz różnica wartości chwilowych prędkości obciążenia (a, c, e) oraz momentu skrętnego (b, d, f) dla różnych wartości parametru  $T_c$

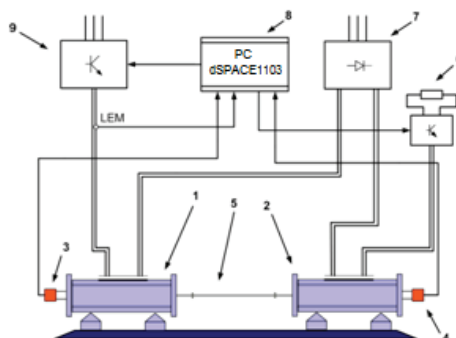
Tabela 2. Zestawienie błędów estymatorów neuronowych

	$T_2=T_{2n}; T_c=T_{cn}$	$T_2=0,8T_{2n}; T_c=T_{cn}$	$T_2=1,2T_{2n}; T_c=T_{cn}$	$T_c=0,8T_{cn}; T_2=T_{2n}$	$T_c=1,2T_{cn}; T_2=T_{2n}$
$Err_{\omega_2}$	1,57	1,78	1,61	3,10	3,81
$Err_{m_s}$	3,60	3,75	4,54	3,89	3,94

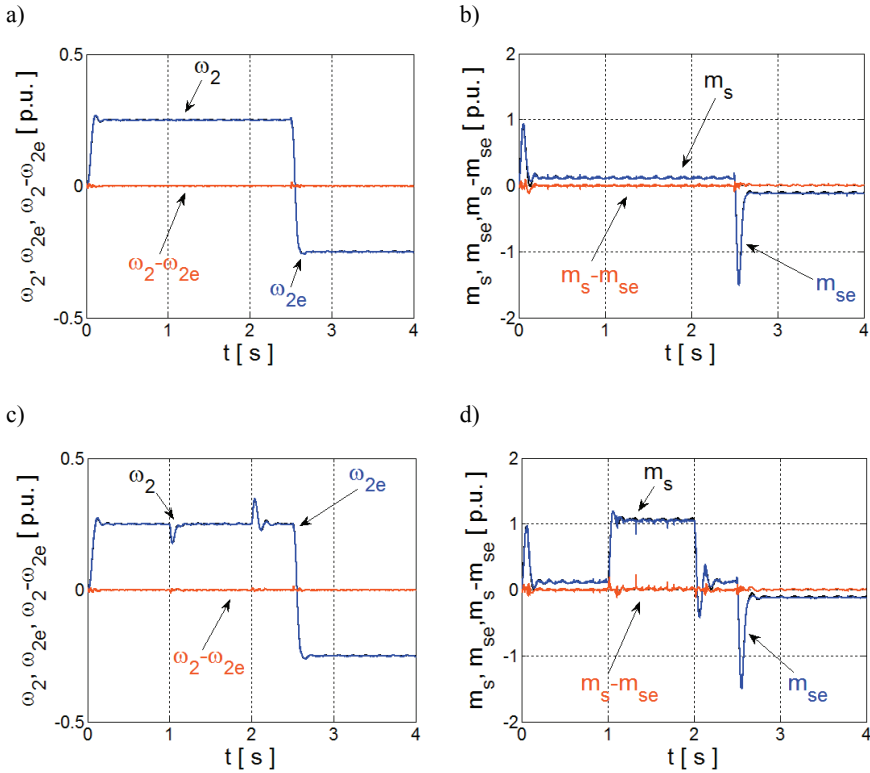
Następnie estymatory neuronowe zostały poddane testom, w których zmieniano wartości parametrów napędu. Wprowadzane zmiany nie były uprzednio uwzględniane w procesie treningu. Wyniki zaprezentowano na rysunku 5 oraz 6. Badania dotyczyły: stałej czasowej obciążenia  $T_2$  oraz stałej czasowej elementu sprzęgającego  $T_c$ . Wartości wymienionych parametrów wprowadzano w zakresie  $\pm 20\%$  w stosunku do wartości znamionowej. Zestawienie błędów zamieszczono w tabeli 1. W każdym z przeprowadzonych testów symulacyjnych, w układzie zadawano złożoną trajektorię prędkości zadanej, w stanach ustalonych prędkości przełączano moment (załączano oraz odłączano) moment obciążenia. Odtwarzane zmienne stanu są obliczane z dużą dokładnością. Wprowadzane zmiany parametrów napędu mają niewielki wpływ na działanie estymatorów neuronowych.

## 5. EKSPERYMENT

Kolejnym etapem prowadzonych badań były testy eksperymentalne analizowanych estymatorów neuronowych. W tym celu opisywane modele neuronowe zostały zaimplementowane oraz poddane testom na stanowisku laboratoryjnym. Schemat struktury stanowiska badawczego zamieszczono na rysunku 7. Algorytm związany z układem sterowania oraz estymatorami neuronowymi został zaimplementowany w procesorze sygnałowym karty dSPACE1103. Obliczenia realizowano na podstawie sygnałów pomiarowych z enkoderów prędkości oraz prądu. Sygnały sterujące struktury laboratoryjnej wyprowadzono do sterowalnych układów energoelektronicznych. Wirtualny panel operatorski oraz rejestracja danych pomiarowych została zrealizowana za pomocą oprogramowania ControlDesk. Układ dwumasowy został wykonany za pomocą dwóch silników prądu stałego połączonych długim wałem. Krok obliczeniowy skompiłowanej aplikacji wyniósł 0,5 ms.



Rys. 7. Schemat stanowiska laboratoryjnego: 1 – silnik, 2 – maszyna obciążająca, 3,4 – enkodery, 5 – wał łączący, 6 – rezystor hamujący, 7 – prostownik diodowy, 8 – układ sterujący, 9 – przetwornik tranzystorowy



Rys. 8. Przebiegi rzeczywiste, estymowane oraz różnica wartości chwilowych prędkości obciążenia (a, c) oraz momentu skrętnego (b, d) – przykładowe przebiegi

Na rysunku 8 zaprezentowano wybrane wyniki badań, prezentujące działanie estymatorów neuronowych na obiekcie rzeczywistym. Zaaplikowano sieci neuronowe, których topologia początkowa, wektory wejściowe, metodyka doboru współczynników wagowych, zostały opracowane zgodnie z opisem przedstawionym w poprzednich rozdziałach artykułu. Pierwsze testy wykonano dla napędu, w którym nie załączano obciążenia (rysunek 8a,b). Przedstawiono działanie układu dla skokowej zmiany prędkości zadanej do 25% prędkości znamionowej, w czasie  $t = 2,5$  s następuje nawrót napędu. Zarówno prędkość obciążenia jak moment skrętny jest odtwarzany bardzo dokładnie. Następnie testy powtórzono w obecności zmian momentu obciążenia, które wprowadzono w czasie  $t = 1$  s oraz  $t = 2$  s. Charakterystyczne jest precyzyjne estymowanie obu zmiennych stanu w trakcie występowania chwilowych zakłóceń spowodowanych przełączaniem obciążenia. W analizowanym przypadku błąd estymacji prędkości obciążenia wynosił  $Err\omega_2 = 0,66$ , błąd odtwarzania momentu skrętnego:  $Errm_s = 6,61$ .

## 6. PODSUMOWANIE

W artykule opisano aplikację sieci neuronowych w strukturze napędowej o złożonej części mechanicznej. Zadaniem zastosowanych modeli było odtwarzanie prędkości obciążenia oraz momentu skrętnego napędu z połączeniem sprężystym. Uzyskano bardzo dużą dokładność estymacji. Opracowane estymatory charakteryzują się niewielką wrażliwością na zmiany parametrów napędu. W trakcie obliczeń współczynników modelu neuronowego zastosowano metodę *Levenberga–Marquardta*. W procesie uczenia przeprowadzono redukcję struktury sieci neuronowej w oparciu o współczynnik istotności *Local Relative Sensitivity Index*. Usunięcie części obliczeń modelu neuronowego wpłynęło na zmniejszenie błędu odtwarzania obu zmiennych stanu. Należy zaznaczyć, że założono usunięcie tylko jednego neuronu ze struktury każdego z estymatorów. W związku z tym wyeliminowano konieczność dalszej analizy procesu redukcji sieci neuronowej (skrócono czas obliczeń), osiągnięty efekt jest poprawą dokładności odtwarzania zmiennych stanu, nie jest optymalną strukturą modelu neuronowego. Opisane estymatory zmiennych stanu charakteryzują się specyficzną metodyką projektowania, w której nie wymagany jest model matematyczny obiektu oraz jego parametry. Bardzo precyzyjne sygnały estymowane otrzymano również w eksperymencie na stanowisku laboratoryjnym.

## LITERATURA

- [1] GRZESIAK L.M., MEGNACK V., SOBOLEWSKI J., UFNALSKI B., *Genetic Algorithm for Parameters Optimization of ANN-based Speed Controller*, Proc. of The International Conference on Computer as a Tool, 2007, 1700–1705.
- [2] KAŻMIERKOWSKI M.P., ORŁOWSKA-KOWALSKA T., *NN State Estimation and Control in Converter-Fed Induction Motor Drives*, Chapter 2 in a book: “Soft Computing in Industrial Electronics”, Physica–Verlag, Springer, 2002, Heilderberg, Germany, 45–94.
- [3] SITAO W., CHOW T.W.S., *Intelligent machine fault detection using SOM based RBF neural networks*, IEEE International Joint Conference on Neural Networks, 2004, Vol. 3, 2077–2082.
- [4] VALENZUELA M.A., BENTLEY J.M., LORENZ R.D., *Evaluation of torsional oscillations in paper machine sections*, IEEE Transaction on Industrial Applications, 2005, Vol. 41, No. 2, 493–501.
- [5] ENGLERT M., TRAPP R., DE KLERK R., *Neural Control of a Nonlinear Elastic Two-Mass System*, Computational Intelligence, Theory and Applications, International Conference, 5th Fuzzy Days, 1997, Vol. 1226, 553–553.
- [6] TOMEI P., *A simple PD controller for robots with elastic joints*, IEEE Transactions on Automatic Control, 1991, Vol. 36, No. 10, 1208–1213.
- [7] BATTILOTTI S., LANARI L., *On optimal controllers for elastic joint robots*, Proceedings of the 34th IEEE Conference on Decision and Control, 1995, Vol. 3, 2818–2822.
- [8] PARK T.S., SHIN E.C., OH W.H., YOO J.Y., *Robust speed control for torsional vibration suppression of rolling mill drive system*, Proceedings of the 29th Annual Conference of the IEEE Industrial Electronics Society IECON’03, 2003, 66–71.
- [9] SZABAT K., *Struktury sterowania elektrycznych układów napędowych z połączeniem sprężystym*, Prace Naukowe Instytutu Maszyn, Napędów i Pomiarów Elektrycznych Politechniki Wrocławskiej, Nr 61, Seria: Monografie Nr 19, Oficyna Wydawnicza Politechniki Wrocławskiej, 2008.



- [10] PAUL P. MUSZYŃSKI R., *Zmodyfikowany obserwator w układzie napędowym z połączeniem sprzężystym*, Materiały konferencji SENE'03, 2003, 399–404.
- [11] SZABAT K., ORŁOWSKA-KOWALSKA T., *Optimal design of the extended Kalman filter for the two-mass system using genetic algorithm*, Polish Academy of Sciences, Electrical Engineering Committee, 2006, Vol. 55, No. 3–4, 237–254.
- [12] FAHLMAN S.E., LEBIERE C., *The Cascade-Correlation Learning Architecture*, Advances in Neural Information Processing Systems 2, D. S. Touretzky (ed.), Morgan-Kaufmann, Los Altos CA, 1990.
- [13] OSOWSKI S., *Sieci neuronowe do przetwarzania informacji*, Oficyna Wydawnicza Politechniki Warszawskiej, 2006.
- [14] KAMIŃSKI M., ORŁOWSKA-KOWALSKA T., *Odtwarzanie zmiennych stanu układu napędowego z połączeniem sprzężystym za pomocą sieci neuronowych z redukcją połączeń synaptycznych*, Prace Naukowe Instytutu Maszyn, Napędów i Pomiarów Elektrycznych Politechniki Wrocławskiej. Studia i Materiały, 2008, Vol. 62, Nr 28, 2008, 396–405.
- [15] LE CUN Y., DENKER J.S., SOLLA S.A., *Optimal Brain Damage*, AT&T Bell Laboratories, 1990.
- [16] HASSIBI B., STORK D.G., WOLFF G.J., *Optimal brain surgeon and general network pruning*, CRC-TR-9235, RICOH California Research Center, 1992.
- [17] HASSIBI B., STORK D.G., *Second order derivatives for network pruning: Optimal brain surgeon*, Advances in Neural Information Processing Systems 5, Morgan Kaufmann, 1993.
- [18] HANCOCK P.J.B., *Pruning Neural Nets by Genetic Algorithm*, Proceedings of the International Conference on Artificial Neural Networks, Elsevier, 1992, 991–994.
- [19] PONNAPALLI P.S., HO K.C., THOMSON M., *A Formal Selection and Pruning Algorithm for Feedforward Artificial Neural Network Optimization*, IEEE Trans. Neural Netw., 1999, Vol. 10, No. 4, 964–968.
- [20] KARNIN E.D., *A Simple Procedure for Pruning Backpropagation Trained Neural Networks*, IEEE Trans. Neural Networks, 1990, Vol. 1, No. 2, 239–242.
- [21] KAMIŃSKI M., ORŁOWSKA-KOWALSKA T., *Analysis of chosen training algorithms in practical design process of neural network based state estimators of two-mass drive system*, Electromagnetic phenomena in nonlinear circuits, EPNC 2012, XXII symposium, proceedings, Pula, Croatia, June 26–June 29, 2012. Rijeka, Faculty of Engineering, University of Rijeka, Croatia, Poznań, Polskie Towarzystwo Elektrotechniki Teoretycznej i Stosowanej, 2012, 85–86.

#### ESTIMATION OF STATE VARIABLES OF TWO-MASS SYSTEM USING NEURAL MODELS

This paper is focused on neural models applied for estimation of state variables drive with elastic connection. Models that estimate speed of the load and shaft torque are analyzed. For calculation of weight coefficients of neural networks the *Levenberg–Marquardt* algorithm was used. This stage of the whole design process was modified by application of the structure reduction of neural models (based on *Local Relative Sensitivity Index*). For this purpose significance factors of the connections were used. Prepared models of state variables have high precision of estimation for wide range of changes of reference speed and switching of the load, and also for changes of the parameters of two-mass system. Simulation results are confirmed by experimental tests.