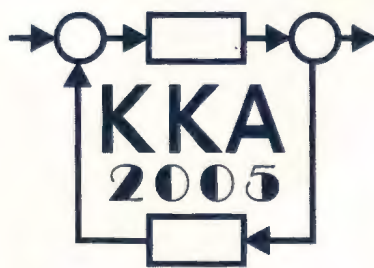


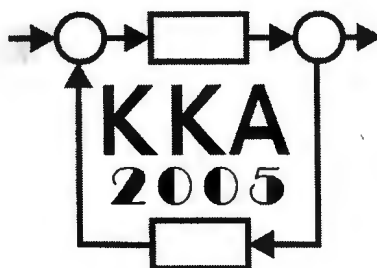
XV Krajowa Konferencja Automatyki

Tom II



**Redaktorzy:
Zdzisław Bubnicki
Roman Kulikowski
Janusz Kacprzyk**

XV Krajowa Konferencja Automatyki Tom II



Redaktorzy:
Zdzisław BUBNICKI
Roman KULIKOWSKI
Janusz KACPRZYK

ORGANIZATOR

Komitet Automatyki i Robotyki Polskiej Akademii Nauk
Instytut Badań Systemowych Polskiej Akademii Nauk

WSPÓLORGANIZATORZY

Politechnika Warszawska

Przemysłowy Instytut Automatyki i Pomiarów

Polskie Stowarzyszenie Pomiarów, Automatyki i Robotyki

ORGANIZATOR

Komitet Automatyki i Robotyki Polskiej Akademii Nauk
Instytut Badań Systemowych Polskiej Akademii Nauk

WSPÓLORGANIZATORZY

Politechnika Warszawska
Przemysłowy Instytut Automatyki i Pomiarów
Polskie Stowarzyszenie Pomiarów, Automatyki i Robotyki

KOMITET PROGRAMOWY

Przewodniczący	Zdzisław BUBNICKI
Zastępca Przewodniczącego	Roman KULIKOWSKI

CZŁONKOWIE

Stanisław BAŃKA	Michał BIAŁKO
Mikołaj BUSŁOWICZ	Władysław FINDEISEN
Ryszard GESSING	Henryk GÓRECKI
Jakub GUTENBAUM	Jerzy JÓZEFczyk
Stanisław KACZANOWSKI	Tadeusz KACZOREK
Janusz KACPRZYK	Jerzy KLAMKA
Józef KORBICZ	Zbigniew KOWALSKI
Krzysztof KOZŁOWSKI	Juliusz L. KULIKOWSKI
Krzysztof KUŹMIŃSKI	Kazimierz MALANOWSKI
Krzysztof MALINOWSKI	Wojciech MITKOWSKI
Antoni NIEDERLIŃSKI	Władysław PEŁCZEWSKI
Tadeusz PUCHAŁKA	Leszek RUTKOWSKI
Stanisław SKOCZOWSKI	Roman SŁOWIŃSKI
Jerzy ŚWIĄTEK	Andrzej ŚWIERNIAK
Ryszard TADEUSIEWICZ	Piotr TATJEWSKI
Krzysztof TCHOŃ	Leszek TRYBUS
Jan WĘGLARZ	Andrzej P. WIERZBICKI

KOMITET ORGANIZACYJNY

Przewodniczący	Roman KULIKOWSKI
Zastępcy Przewodniczącego	Janusz KACPRZYK
	Stanisław KACZANOWSKI
	Tadeusz KACZOREK
	Krzysztof MALINOWSKI
Członkowie	Roman OSTROWSKI
	Tadeusz PUCHAŁKA
	Dariusz WAGNER
Sekretarze naukowci	Jan STUDZIŃSKI
	Jan W. OWSIŃSKI

ISBN 83-89475-01-4

Copyright © Instytut Badań Systemowych Polskiej Akademii Nauk
All rights reserved

Druk: ARGRAF, Warszawa

ROBOTY

IDENTYFIKACJA MODELU ROBOTA ZA POMOCĄ REKURENCYJNYCH SIECI NEURONOWYCH – SYNTEZA UKŁADU STEROWANIA ROBOTA

Cezary WILDNER, Jerzy E. KUREK

Politechnika Warszawska, Wydział Mechatroniki, Instytut Automatyki i Robotyki,
ul. Św. Andrzeja Boboli 8, 02-525 Warszawa,
e-mail: c.wildner@mchtr.pw.edu.pl, j.kurek@mchtr.pw.edu.pl

Streszczenie: Matematyczny model robota jest dobrze znany np. w postaci równań Lagrange'a-Eulera. Parametry fizyczne robota wykorzystywane w modelu matematycznym są jednak trudne do wyznaczenia w sposób analityczny dla rzeczywistego robota (momenty inercyjne, masy, wymiary członów itd.). W pracy przedstawiono wyniki otrzymane przy aproksymacji nieliniowych funkcji modelu robota za pomocą sieci neuronowych bez demontażu robota. Przedstawiono także wyniki uzyskane w czasie syntezy układu sterowania robota zgodnie z algorytmem Iteracyjnie Uczącego się Sterowania (IUS) na podstawie wyznaczonego neuronowego modelu robota.

Słowa kluczowe: Model robota Lagrange'a-Eulera, sieci neuronowe, robot Puma 560, iteracyjnie uczące się sterowanie.

1. MATEMATYCZNY MODEL ROBOTA

Równania opisujące własności dynamiczne robota z n stopniami swobody mogą być wyznaczone w oparciu o równanie Lagrange'a-Eulera [2] w następującej postaci

$$M(\theta)\ddot{\theta} + V(\theta, \dot{\theta}) + G(\theta) = \tau \quad (1)$$

gdzie $\theta \in R^n$ wektor współrzędnych uogólnionych, $\tau \in R^n$ wektor momentów napędowych, $M(\theta) \in R^{n \times n}$ macierz momentów bezwładności robota, $V(\theta, \dot{\theta}) \in R^n$ wektor momentów zależnych od sił odśrodkowych i sił Coriolisa, $G(\theta) \in R^n$ wektor momentów zależnych od sił grawitacji.

Dokonując następujących aproksymacji pochodnych:

$$\dot{\theta}(t) \approx \frac{\theta(t) - \theta(t - T_p)}{T_p} \quad (2)$$

$$\ddot{\theta}(t) \approx \frac{\dot{\theta}(t + T_p) - \dot{\theta}(t)}{T_p} \quad (3)$$

gdzie T_p jest okresem próbkowania, bazując na (1) można wyznaczyć model robota z czasem dyskretnym w postaci

$$\theta(k+1) = 2\theta(k) - \theta(k-1) + A[\theta(k), \theta(k-1)] + B[\theta(k)] + C[\theta(k)]\tau(k) \quad (4)$$

gdzie k jest czasem dyskretnym, $t = kT_p$ oraz $A[\]$, $B[\]$ i $C[\]$ są macierzami, których elementy a_i^* , b_i^* , c_{ij}^* są funkcjami nieliniowymi zmiennych $\theta(k)$ i $\theta(k-1)$ i zależą od parametrów robota takich jak momenty inercyjne, masy, wymiary członów:

$$A[\theta(k), \theta(k-1)] = -T_p^2 M^{-1}[\theta(k)]V[\theta(k), \theta(k-1)] \quad (5)$$

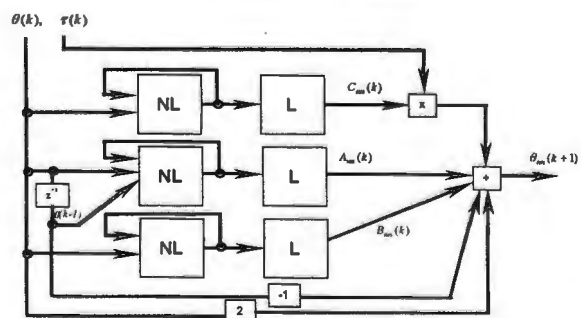
$$B[\theta(k)] = -T_p^2 M^{-1}[\theta(k)]G[\theta(k)] \quad (6)$$

$$C[\theta(k)] = T_p^2 M^{-1}[\theta(k)] \quad (7)$$

W pracy przedstawiono wyniki aproksymacji nieliniowych funkcji elementów macierzy A , B , C modelu robota (4) za pomocą rekurencyjnych sieci neuronowych bez demontażu robota oraz wyniki uzyskane w czasie syntezy układu sterowania robota zgodnie z algorytmem IUS [3, 4] na podstawie wyznaczonego modelu (4).

2. NEURONOWY MODEL ROBOTA

Do aproksymacji nieliniowych funkcji elementów macierzy A , B , C modelu robota (4) wykorzystano przedstawiony na rys. 1 układ trzech dwuwarstwowych sieci neuronowych rekurencyjnych Elmana.



Rys. 1. Układ sieci neuronowych.

Równanie neuronu ma następującą postać:

$$y(k) = f(v(k)) = f\left(\sum_{i=1}^h w_i u_i(k) + b\right) \quad (8)$$

gdzie w_i oznacza wagę, b oznacza przesunięcie, u_i jest sygnałem wejściowym do neuronu oraz $f(\cdot)$ jest funkcją aktywacji neuronu. Każda z sieci składa się z ukrytej warstwy NL zbudowanej z neuronów o nieliniowej funkcji aktywacji opisanej równaniem (9) i wyjściowej warstwy L , która składa się z neuronów o liniowej funkcji aktywacji (10).

$$f_{NL}(v_i) = \text{tansig}(v_i) = \frac{2}{1 + e^{-2v_i}} - 1 \quad f \in [-1,1] \quad (9)$$

$$f_L(v_i) = v_i \quad (10)$$

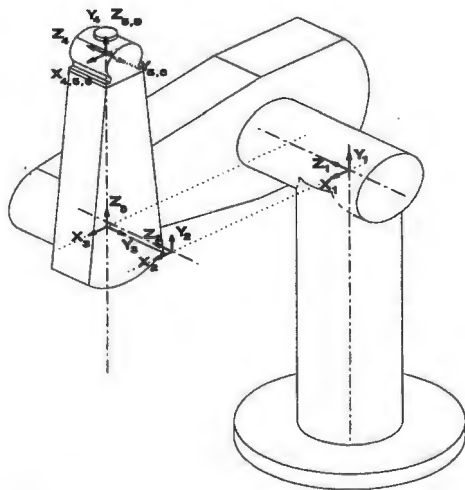
Każda z sieci jest kompletnie połączona, tzn. sygnały wejściowe do kolejnych warstw połączone są ze wszystkimi neuronami w tych warstwach. Przyjęto następujący wskaźnik jakości uczenia sieci neuronowych

$$J(d) = \frac{1}{2} \sum_{k=1}^v \sum_{j=1}^n [\theta_j(k) - \theta_{nnj}(k, d)]^2 \quad (11)$$

gdzie d oznacza numer iteracji uczącej, v oznacza ilość próbek uczących, θ_{nnj} wektor wyjść z sieci aproksymujący θ_j . Do nauki sieci wykorzystano metodę wstecznej propagacji błędów [7].

2.1. Wyniki symulacji komputerowych

Proponowany sposób aproksymacji nieliniowych funkcji modelu robota za pomocą sieci neuronowych wykorzystano do obliczeń parametrów matematycznego modelu robota Puma 560, rys. 2. Na rysunku pokazano również osie lokalnych układów współrzędnych wyznaczone wg. algorytmu Denavita-Hartenberga (D-H) [2]. Parametry robota przedstawiono w tab. 1, 2 i 3. Dane uczące i testujące uzyskano z komputerowych symulacji robota Puma 560 o $n=6$ stopniach swobody i przegubach obrotowych.



Rys. 2. Robot Puma 560 [1].

Tab. 1. Parametry lokalnych układów współrzędnych D-H robota Puma 560 [1] (α_i – kąt skręcenia, a_i – długość członu, θ_i – kąt konfiguracji, d_i – odsunięcie członów).

Przegub	α_i [°]	a_i [m]	θ_i [°]	d_i [m]
1	90	0	0	0
2	0	-0.4318	0	0
3	-90	-0.0203	0	0.15005
4	90	0	0	0.4318
5	-90	0	0	0
6	0	0	0	0

Tab. 2. Parametry inercyjne robota Puma 560 [1]. (I_{xx} , I_{yy} , I_{zz} – masowe momenty bezwładności członów względem ich środków mas, I_{xy} , I_{yz} , I_{zx} – masowe momenty dewiacji członów względem ich środków mas).

Przegub	I_{xx} [kg m ²]	I_{yy} [kg m ²]	I_{zz} [kg m ²]	$I_{xy}=I_{yz}=I_{zx}$ [kg m ²]
1	0	0.35	0	0
2	0.13	0.524	0.539	0
3	0.066	0.086	0.0125	0
4	0.0018	0.0013	0.0018	0
5	0.0003	0.0004	0.0003	0
6	0.00015	0.00015	0.00004	0

Tab. 3. Parametry inercyjne robota Puma 560 [1]. (M – masa członu, $r=[r_x, r_y, r_z]^T$ – jednorodny wektor współrzędnych środka masy).

Przegub	M [kg]	r_x [m]	r_y [m]	r_z [m]
1	0	0	0	0
2	17.4	-0.3638	0.006	0.2275
3	4.8	-0.0203	-0.0141	0.070
4	0.82	0	0.019	0
5	0.34	0	0	0
6	0.08	0	0	0.032

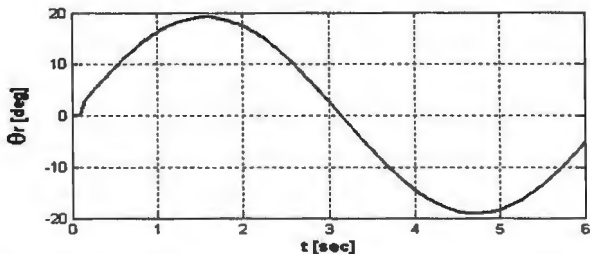
Do nauki sieci wygenerowano sekwencję $v=600$ danych uczących wykorzystując model (4). Wszystkie przeguby robota miały poruszać się po zadanej trajektorii (12) w czasie $t=6$ [sec] z okresem próbkowania $T_p=0.05$ [sec]. Następnie w celu sprawdzenia możliwości nauczonych sieci neuronowych przeprowadzono także próby dla innej trajektorii niż trajektoria ucząca.

$$\theta_r(t) = \frac{\sin(t)}{3} \quad (12)$$

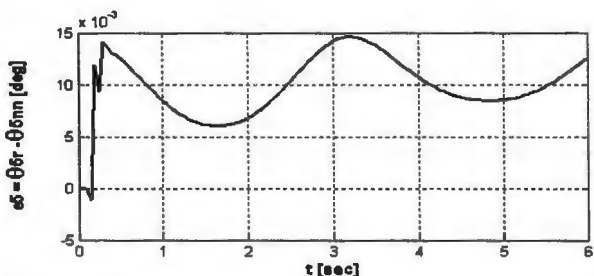
Przykładowe wyniki badań dotyczących wyznaczania neuronowego modelu robota przedstawiono na rys. 3, 4, 5 i 6 oraz w tab. 4 i 5. Na rys. 3 i 4 przedstawiono wyniki uzyskane podczas uczenia sieci neuronowych, a na rys. 5 i 6 wyniki podczas testowania sieci neuronowych

przy danych innych niż dane uczące. W tab. 4 przedstawiono maksymalny błąd między zadaną trajekcją uczącą θ_r i wyjściem θ_{nnj} uzyskanym z modelu neuronowego po 500 iteracjach uczących, a w tab. 5 pomiędzy trajekcją testową θ_t (13) dla wszystkich przegubów i wyjściem θ_{nnj} sieci neuronowej, dla której zakończono proces uczenia.

$$\theta_r(t) = \frac{\sin(3t)}{2} \quad (13)$$



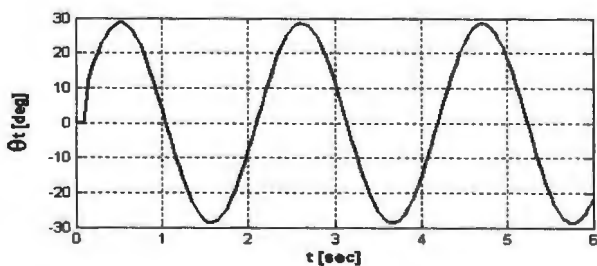
Rys. 3. Dane uczące: sygnał wyjściowy robota Puma 560.



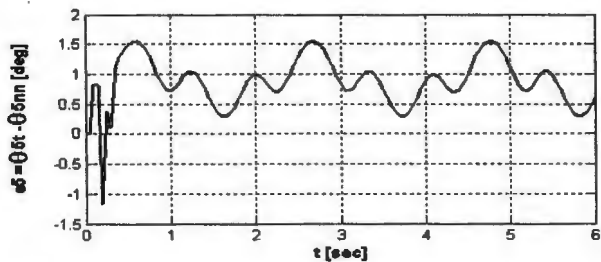
Rys. 4. Różnica pomiędzy zadaną trajekcją uczącą θ_r i wyjściem $\theta_{nn,5}$ uzyskanym z modelu neuronowego dla piątego przegubu robota.

Tab. 4. Maksymalny błąd $|\theta_r - \theta_{nnj}|_{\max}$ po 500 iteracjach uczących.

Przegub	$ \theta_r - \theta_{nnj} _{\max}$
1	0.083 [°]
2	0.069 [°]
3	0.098 [°]
4	0.106 [°]
5	0.014 [°]
6	0.115 [°]



Rys. 5. Dane testowe: zadaną trajekcją θ_t dla wszystkich przegubów robota.



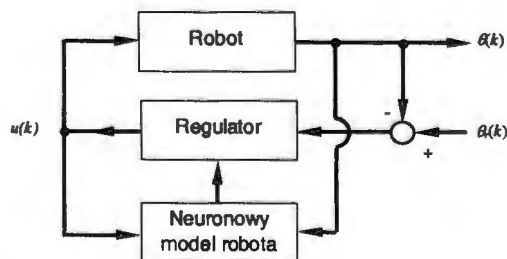
Rys. 6. Różnica pomiędzy zadaną trajekcją testową θ_t i wyjściem $\theta_{nn,5}$ uzyskanym z modelu neuronowego dla piątego przegubu.

Tab. 5. Maksymalny błąd $|\theta_t - \theta_{nnj}|_{\max}$ dla danych testowych.

Przegub	$ \theta_t - \theta_{nnj} _{\max}$
1	2.067 [°]
2	2.745 [°]
3	1.907 [°]
4	2.003 [°]
5	1.548 [°]
6	3.098 [°]

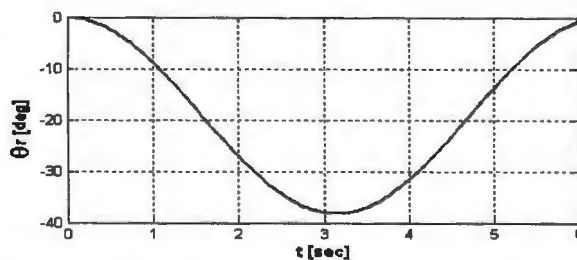
2.2. Wyniki syntezy układu sterowania robota

Na podstawie otrzymanego neuronowego modelu dokonano syntezy układu sterowania robota zgodnie z algorytmem IUS. Schemat układu sterowania robota przedstawiono na rys. 7. Sygnał zadany (14) dla wszystkich stopni swobody robota przedstawiony jest na rys 8. Trajektorja (14) jest inna niż trajektorje zastosowane wcześniej jako uczące i testowe sygnały zadane dla sieci neuronowych.



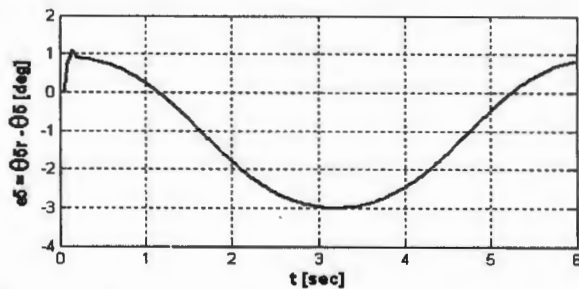
Rys. 7. Układ sterowania robota.

$$\theta_r(t) = \frac{-1 + \cos(t)}{3} \quad (14)$$

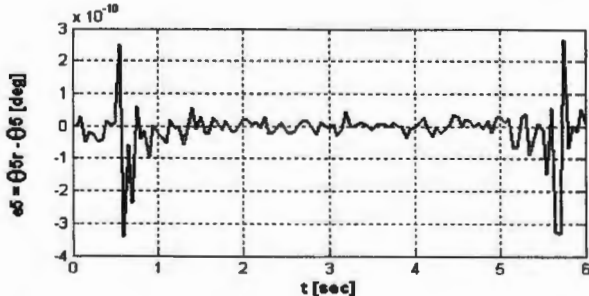


Rys. 8. Sygnał zadany θ_r dla wszystkich przegubów robota Puma 560.

Na rys. 9, 10, 11 i 12 przedstawiono wyniki otrzymane po 1 iteracji IUS dla piątego przegubu robota. Na rys. 9 pokazano różnicę między zadaną trajektorią θ_r i wyjściem θ_j robota Puma 560. W przypadku, gdy sterowanie było wyznaczane na podstawie dokładnego modelu robota, otrzymany błąd dla wszystkich przegubów był rzędu 10^{-10} [°], np. rys. 10 (rzęd błądu wynikający z dokładności obliczeń numerycznych). Sygnały sterujące robota po 1 iteracji IUS przedstawiono odpowiednio na rys. 11 i 12. Widać, że sygnał sterujący dla przypadku aproksymacji nieliniowych funkcji modelu robota za pomocą sieci neuronowych różni się od sygnału sterującego dla dokładnego modelu robota po 1 iteracji IUS. W tab. 6 przedstawiono maksymalny błąd między zadaną trajektorią θ_r i wyjściem θ_j robota Puma 560 po 1 iteracji IUS dla wszystkich przegubów.



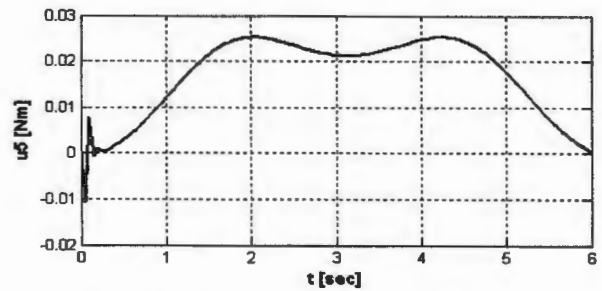
Rys. 9. Różnica pomiędzy zadaną trajektorią θ_r i wyjściem θ_j robota Puma 560 po 1 iteracji IUS dla piątego przegubu - aproksymacja nieliniowych funkcji modelu robota za pomocą sieci neuronowych.



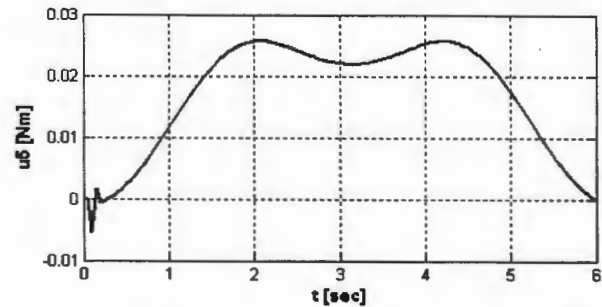
Rys. 10. Różnica pomiędzy zadaną trajektorią θ_r i wyjściem θ_j robota Puma 560 po 1 iteracji IUS dla piątego przegubu - dokładny model robota.

Tab. 6. Maksymalny błąd $|\theta_r - \theta_j|_{\max}$ po 1 iteracji IUS - aproksymacja nieliniowych funkcji modelu robota za pomocą sieci neuronowych.

Przegub	$ \theta_r - \theta_j _{\max}$
1	1.591 [°]
2	2.413 [°]
3	4.890 [°]
4	1.774 [°]
5	3.004 [°]
6	0.711 [°]

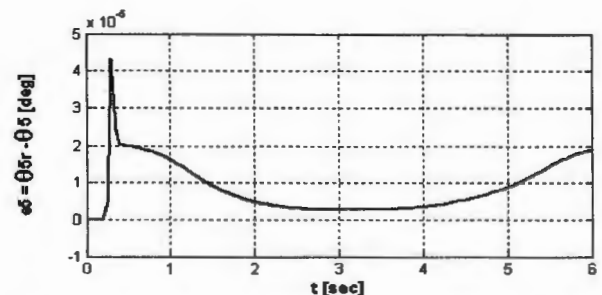


Rys. 11. Sygnał sterujący $\tau=u$ robota Puma 560 po 1 iteracji IUS dla piątego przegubu - aproksymacja nieliniowych funkcji modelu robota za pomocą sieci neuronowych.

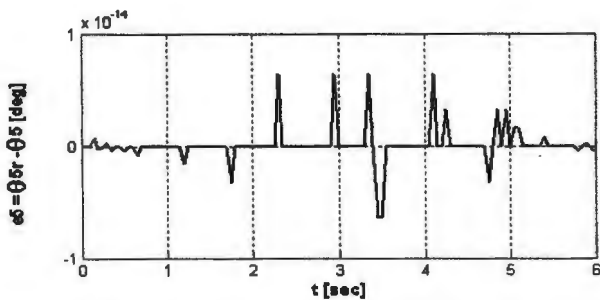


Rys. 12. Sygnał sterujący $\tau=u$ robota Puma 560 po 1 iteracji IUS dla piątego przegubu - dokładny model robota.

Na rys. 13, 14, 15 i 16 przedstawiono również wyniki po 5 iteracjach IUS dla piątego przegubu robota. Na rys. 13 pokazano różnicę między zadaną trajektorią θ_r i wyjściem θ_j robota Puma 560. Widać, że błąd został zmniejszony, rzędu 10^{-5} [°]. W przypadku, gdy sterowanie było wyznaczane na podstawie dokładnego modelu robota, otrzymany błąd dla wszystkich przegubów był rzędu 10^{-14} [°], np. rys. 14. Sygnały sterujące robota przedstawiono odpowiednio na rys. 15 i 16. Widać, że sygnały sterujące dla obu przypadków są praktycznie takie same po 5 iteracjach IUS. W tab. 7 przedstawiono maksymalny błąd między zadaną trajektorią θ_r i wyjściem θ_j robota Puma 560 po 5 iteracjach IUS dla wszystkich przegubów.



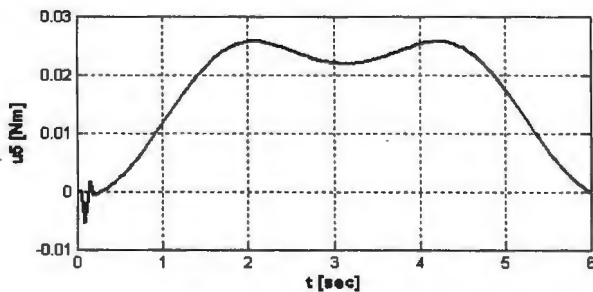
Rys. 13. Różnica pomiędzy zadaną trajektorią θ_r i wyjściem θ_j robota Puma 560 po 5 iteracjach IUS dla piątego przegubu - aproksymacja nieliniowych funkcji modelu robota za pomocą sieci neuronowych.



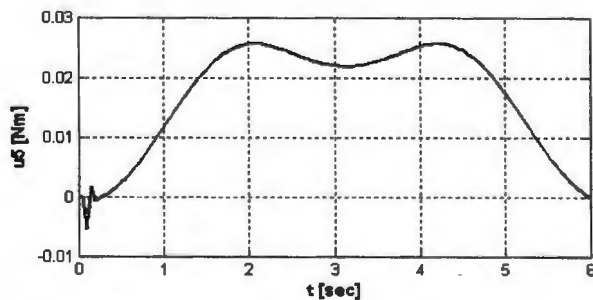
Rys. 14. Różnica pomiędzy zadaną trajektorią θ_r i wyjściem θ_j robota Puma 560 po 5 iteracji IUS dla piątego przegubu - dokładny model robota.

Tab. 7. Maksymalny błąd $|\theta_r - \theta_j|_{\max}$ po 5 iteracji IUS - aproksymacja nieliniowych funkcji modelu robota za pomocą sieci neuronowych.

Przegub	$ \theta_r - \theta_j _{\max}$
1	$7.3 \cdot 10^{-7}$ [°]
2	$4.4 \cdot 10^{-7}$ [°]
3	$2.5 \cdot 10^{-6}$ [°]
4	$4.0 \cdot 10^{-7}$ [°]
5	$4.3 \cdot 10^{-5}$ [°]
6	$2.3 \cdot 10^{-6}$ [°]



Rys. 15. Sygnał sterujący $\tau=u$ robota Puma 560 po 5 iteracji IUS dla piątego przegubu - aproksymacja nieliniowych funkcji modelu robota za pomocą sieci neuronowych.



Rys. 16. Sygnał sterujący $\tau=u$ robota Puma 560 po 5 iteracji IUS dla piątego przegubu - dokładny model robota.

3. UWAGI KOŃCOWE

W pracy do aproksymacji nieliniowych funkcji modelu robota Puma 560 użyto rekurencyjnych sieci neuronowych. Na podstawie rezultatów uzyskanych podczas uczenia sieci neuronowych można stwierdzić, że nieliniowe funkcje modelu robota były aproksymowane z dużą dokładnością. Średnia różnica pomiędzy wyjściem robota i wyjściem modelu neuronowego po 500 iteracjach wynosiła dla danych uczących w przybliżeniu 0.1 [°], tab. 4.

Podczas testu przeprowadzonego dla danych testowych, innych niż dane uczące, średnia różnica między wyjściem robota, a wyjściem neuronowego modelu wynosiła w przybliżeniu 2.2 [°], tab. 5.

Następnie na podstawie wyznaczonego neuronowego modelu dokonano syntezy układu sterowania robota zgodnie z algorytmem IUS. Na podstawie uzyskanych wyników po piątej iteracji IUS widać, że błąd jest mały, rzędu $10^{-5} \div 10^{-7}$ [°], tab. 7. Uzyskane rezultaty wskazują na to, że wyznaczony model mógłby być wykorzystany do syntezy sterowania robota. Zastosowanie sieci neuronowych powinno pozwolić na projektowanie układu sterowania wtedy, gdy nie jest możliwe wyznaczenie parametrów fizycznych robota metodami analitycznymi.

Uzyskane struktury sieci neuronowych wymagają dalszego sprawdzenia przydatności do syntezy układu sterowania robota, a w szczególności czy wyznaczony model generalizuje dane w całej roboczej przestrzeni robota.

Dysponując dobrym neuronowym modelem robota można podjąć próbę implementacji adaptacyjnego algorytmu sterowania robota i odpornego na zmiany parametrów obiektu oraz środowiska. Wyznaczenie modelu robota w taki sposób pozwoli na uniezależnienie tego przedsięwzięcia od producentów sprzętu.

IDENTIFICATION OF ROBOT MODEL USING RECURRENT NEURAL NETS - SYNTHESIS OF CONTROL SYSTEM

Abstract: There is presented method for calculation of parameters of robot model in the form of the Lagrange-Euler equations using recurrent neural nets. Neural model of robot has been built using only information from inputs and outputs of robot (i. e. control signals and joint positions) and knowledge of model structure. There are presented the possible architecture of neural nets and the results obtained during calculation of the model for PUMA 560 robot. Then, there are presented results of synthesis of control system according to iterative learning control of robot based on the obtained neural model of the robot.

Literatura

- [1] Corke P.I. (1999) *Matlab Robotics Toolbox (release 5)*, CSIRO, Australia.
- [2] Fu K.S., Gonzalez R.C., Lee C.S.G. (1987) *Robotics, Control, Sensing, Vision, and Intelligence*. McGraw-Hill Book Company.

- [3] Kurek J.E. (1999) Iteracyjnie uczące się sterowanie robota. *Pomiary Automatyka Kontrola*, 8, 32-35.
- [4] Kurek J.E. (1999) Iteracyjnie uczące się sterowanie dla układu z czasem dyskretnym z zakłóceniami. *XIII Krajowa Konferencja Automatyki, Opole, 1999*.
- [5] Możaryn J., Kurek J.E. (2003) Comparison of the sliding mode control and decoupled sliding mode control of robot Puma 560, *9th IEEE International Conference of Methods and Models in Automation and Robotics, Międzyzdroje 2003*.
- [6] Możaryn J., Wildner C., Kurek J.E. (2002) Wyznaczanie parametrów modelu robota przemysłowego przy pomocy sieci neuronowych, *XIV Krajowa Konferencja Automatyki KKA 2002, Zielona Góra, 2002*.
- [7] Osowski S. (1996) *Sieci neuronowe w ujęciu algorytmicznym*. WNT, Warszawa.
- [8] Wildner C., Kurek J.E. (2002) Calculation of SCARA robot model using neural nets, *Third International Workshop On Robot Motion And Control, ROMOCO, Bukowy Dworek*.
- [9] Wildner C., Kurek J.E. (2003) Calculation of robot model using neural nets and evolutionary algorithm, *Mechatronics, Robotics And Biomechanics, Hrotovice, Czech Republic 2003*.
- [10] Wildner C., Kurek J.E. (2003) Calculation of Puma 560 robot model using neural nets and evolutionary algorithm, *Inter Academia – 2nd International Conference on Global Research and Education, Warszawa*.
- [11] Wildner C., Kurek J. E. (2003) Wyznaczanie parametrów modelu robota przy pomocy radialnych sieci neuronowych, *X Warsztaty Naukowe Polskiego Towarzystwa Symulacji Komputerowej – Symulacja w Badaniach i Rozwoju, Zakopane*.



Instytut Badań Systemowych
Polskiej Akademii Nauk

ISBN 83-89475-01-4