# XV Krajowa Konferencja Automatyki

## Tom III

KKA 2005

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

# XV Krajowa Konferencja Automatyki
## Tom III

Redaktorzy:
Zdzisław BUBNICKI
Roman KULIKOWSKI
Janusz KACPRZYK

**ORGANIZATOR**
Komitet Automatyki i Robotyki Polskiej Akademii Nauk
Instytut Badań Systemowych Polskiej Akademii Nauk
**WSPÓŁORGANIZATORZY**
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

## WSPÓŁORGANIZATORZY

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 naukowi | Jan STUDZIŃSKI |
| | Jan W. OWSIŃSKI |

**ISBN 83-89475-02-2**

Copyright © Instytut Badań Systemowych Polskiej Akademii Nauk
All rights reserved

Druk: ARGRAF, Warszawa

# SZTUCZNA INTELIGENCJA

# ROZMYTE MODELOWANIE TARCIA PRZY POMOCY ALGORYTMU PSEUDO-BAKTERYJNEGO I METODY NAJMNIEJSZYCH KWADRATÓW

Marcin JASTRZĘBSKI

Politechnika Łódzka, Wydział Elektrotechniki i Elektroniki, Instytut Automatyki
ul. Stefanowskiego 18/22, 91-924 Łódź e-mail: marcinjastrzebski@gazeta.pl

**Streszczenie:** Zaproponowano nową metodę strojenia rozmytej bazy reguł typu Sugeno przy wykorzystaniu zmodyfikowanego algorytmu pseudo-bakteryjnego (PBGA) i metody najmniejszych kwadratów (LSQ). Ponieważ podejmowana jest tutaj próba zamodelowania zjawiska o charakterze nieciągłym, zaproponowane zostały nowe rodzaje funkcji następników reguł. W przedstawionym algorytmie (PBGA+LSQ) strojenie parametrów wejściowych funkcji przynależności przeprowadzane jest przy użyciu algorytmu PBGA, natomiast strojnie następników przy pomocy metody LSQ. Efektywność metody zilustrowano trzema przykładami.

**Słowa kluczowe:** Modelowanie tarcia, system rozmyty, algorytm bakteryjny.

## 1. WSTĘP

Wykorzystanie modeli tarcia jest użyteczne między innymi w projektowaniu precyzyjnych serwomechanizmów, systemów sterownia robotów, czy systemów anty-poślizgowych w hamulcach samochodowych. Fakt ten tłumaczy dużą liczbę publikacji, które skupiają się na analizie i modelowaniu tego zjawiska.
W artykule [4] znajomość siły tarcia wykorzystana była do wyznaczenia sygnału sterującego silnikiem liniowym. Wartość tej siły obliczona była przez model rozmyty typu Sugeno. Model nie był dość dokładny. Różnica między siłą uzyskaną z modelu, a siłą prezentowaną podczas uczenia była szczególnie widoczna gdy prędkość silnika była bliska zeru. Powodem tej rozbieżności jest to, że dla $v$=0 tarcie jest nieciągłe, podczas gdy następniki funkcji w standardowym systemie rozmytym typu Sugeno są liniową kombinacją wartości wejść (ciągłe następniki).
Wiele technik jest używanych do projektowania modeli rozmytych, jak np. dobrze znane podejście neuro-rozmyte, klastrowanie, czy algorytmy ewolucyjne.
Na podstawie obserwacji zdolności bakterii do transferu genów, autorzy [6] zaproponowali prosty algorytm pseudo-bakteryjny (PBGA).
Proponowany algorytm wykorzystuje fakt, że parametry funkcji następników rozmytych reguł systemu typu Sugeno mogą być dobierane w trakcie jednej iteracji przy pomocy metody najmniejszych kwadratów (LSQ). Rozszerzenie liniowych następników o zaproponowany nieciągły składnik zawierający informację o miejscu i wielkości nieciągłości nie zmienia tej sytuacji.
W artykule proponowana jest nowa metoda strojenia parametrów systemu rozmytego, wykorzystująca algorytm pseudo-bakteryjny (PBGA) i metodę najmniejszych kwadratów (LSQ).

## 2. SYSTEM ROZMYTY

Rozpatrzmy $n$-wejściowy $[x_1, x_2, ..., x_n]$ model rozmyty, złożony z $N$ reguł postaci:

$$\begin{aligned} \text{Ri:}\quad & \text{IF } (x_1 \text{ is } \mu_{i1}) \text{ and } (x_2 \text{ is } \mu_{i2}) \text{ and } \ldots \text{and } (x_n \text{ is } \mu_{in}) \\ & \text{THEN } y \text{ is } y_i = f_i(x_1, x_2, \ldots, x_n), \end{aligned} \tag{1}$$

gdzie $i$=1..$N$ oznacza numer reguły rozmytej, w których „dzwonowate" funkcje przynależności $\mu_{ij}$ dane są wzorem:

$$\mu_{ij}(x_j) = \frac{1}{1 + \left(\dfrac{x_j - c_{ij}}{a_{ij}}\right)^{2b_{ij}}}, \tag{2}$$

a następniki reguł określone zostały jako:

$$\begin{aligned} f_i(x_1, x_2, \ldots, x_n) = {} & p_{i1} \cdot x_1 + p_{i2} \cdot x_2 + \ldots + p_{in} \cdot x_n + \\ & + q_i + k_i \cdot sign\{G(x_1, x_2, \ldots, x_n)\}, \end{aligned} \tag{3}$$

gdzie $G(x_1, x_2, \ldots, x_n)$ jest wcześniej określoną funkcją opisującą granicę nieciągłości w modelowanym zjawisku. Wyjście $y$ obliczamy jako średnią ważoną wartości następników wszystkich reguł

$$y = \frac{\sum_{i=1}^{N} o_i \cdot y_i}{\sum_{i=1}^{N} o_i}, \tag{4}$$

gdzie

$$o_i = \prod_{j=1}^{n} \mu_{ij}\{x_j\} \tag{5}$$

jest siłą odpalenia *i-tej* reguły.

Rozważmy wsad danych wejściowych zapisany w macierzy $X$ o elementach $x_{hj}$, $h=1...k$, $j=1...n$, gdzie $k$ oznacza liczbę wektorów wierszowych $[x_1, x_2, ..., x_n]$. Oznaczmy też przez $M\{i\}$ macierz wartości funkcji przynależności dla $i$-tej reguły o elementach $m\{i\}_{hj}=\mu_{ij}(x_{hj})$.
Siły odpalenia poszczególnych reguł, dla kolejnych wektorów wejściowych zapiszmy w macierzy $O$ o elementach:

$$o_{hi} = \prod_{j=1}^{n} m\{i\}_{hj}, \qquad (6)$$

$$h=1..k, \quad i=1..N$$

W dalszych rozważaniach przydatny też będzie wektor kolumnowy $s$ zawierający sumę sił odpalenia wszystkich reguł dla każdego wektora wejściowego $[x_1, x_2, ..., x_n]$, zbudowany wg schematu

$$s_h = \sum_{i=1}^{N} o_{hi}, \qquad (7)$$

$$h=1..k,$$

Zbudujmy $N\cdot(n+2)$ kolumnową macierz o $k$ wierszach

$$\mathbf{E} = \left[\mathbf{E}^{<1>} \quad \mathbf{E}^{<2>} \quad ... \quad \mathbf{E}^{<N>}\right] \qquad (8)$$

z podmacierzy

$$E^{<i>} = \left[e^{<i,1>} \quad e^{<i,2>} \quad ... \quad e^{<i,n>} \quad e^{<i,n+1>} \quad e^{<i,n+2>}\right] \qquad (9)$$

$$i=1..N$$

w których wektory $e^{<i,1>}$, $e^{<i,2>}$, ..., $e^{<i,n>}$ zawierają elementy

$$e_h^{<i,j>} = \frac{o_{hi} \cdot x_{hj}}{s_h} \qquad (10)$$

$$h=1..k, \quad i=1..N, \quad j=1..n$$

wektor $e^{<i,n+1>}$ zawiera elementy

$$e^{<i,n+1>}{}_h = \frac{o_{hi}}{s_h} \qquad (11)$$

$$h=1..k, \quad i=1..N$$

natomiast $e^{<i,n+2>}$ jest takie, że

$$e^{<i,n+2>}{}_h = \frac{o_{hi} \cdot sign\{G(x_{h1}, x_{h2}, ..., x_{hn})\}}{s_h} \qquad (12)$$

$$h=1..k, \quad i=1..N$$

Jeżeli teraz parametry następników zapiszemy wg schematu

$$r = [p_{11} \ ... \ p_{1n} \ q_1 \ k_1 \,|\, ... \,|\, p_{N1} \ ... \ p_{Nn} \ q_N \ k_N]^T \qquad (13)$$

to wektor $y$ zawierający odpowiedź systemu na kolejne wektory danych wejściowych ze wsadu $X$ będzie równy:

$$y = E \cdot r \qquad (14)$$

## 3. ALGORYTM PSEUDO-BAKTERYJNY I JEGO MODYFIKACJA

Bakteriofagi mogą przenieść kopię genu z komórki źródłowej i wprowadzić ją do chromosomu komórki zarażonej. Proces ten nazywany jest transdukcją. Poprzez transdukcję możliwe jest rozprzestrzenienie cechy jednej bakterii na resztę populacji. Ten mechanizm rekombinacji genetycznej jest podstawą ewolucji mikrobów. Zainspirowani tym zjawiskiem autorzy [6] zaproponowali mutację bakteryjną, na działaniu której opiera się algorytm pseudo-bakteryjny (PBGA). Algorytm ten wykorzystywany był z powodzeniem do strojenia systemów rozmytych typu Mamdani [2,6,7] jak i typu Sugeno [3].

W oryginalnym podejściu [6] algorytm pseudobakteryjny składał się z mutacji bakteryjnej, jako mechanizmu polepszania niewielkich części chromosomu i krzyżowania. Dobre wyniki przynosi także zastosowanie algorytmu tylko z mutacją bakteryjną [2,3,7], lecz ze zmieniającą się długością mutowanej części i stopnia tej mutacji.

W przytoczonych pracach, mutacji bakteryjnej poddawane były zarówno parametry funkcji przynależności jak i parametry następników, co odpowiada bakterii o długości $(4\times n+2)\cdot N$ parametrów, dla systemu rozmytego postaci zaproponowanego w rozdziale 2.

W poprzednim rozdziale pokazane zostało, że $y$ zależy liniowo od wektora parametrów następników zebranych w wierszu $r$. Zadanie polega na zminimalizowaniu błędu średniokwadratowego

$$RMSE = \sqrt{\frac{\|\mathbf{d}-\mathbf{y}\|^2}{k}} \qquad (15)$$

Dla danego wsadu $X$ i przy znanych parametrach funkcji przynależności, znana jest również macierz $E$. Wektor minimalizujący wskaźnik $RMSE$ obliczyć możemy przy pomocy metody najmniejszych kwadratów (LSQ) w trakcie jednej iteracji wg wzoru

$$\mathbf{r} = \mathbf{E} \backslash \mathbf{d} = \left(\mathbf{E}^T \cdot \mathbf{E}\right)^{-1} \cdot \mathbf{E}^T \cdot \mathbf{d} \qquad (16)$$

Uwzględniając (14), (15) oraz (16), wskaźnik RMSE zapisać możemy jako:

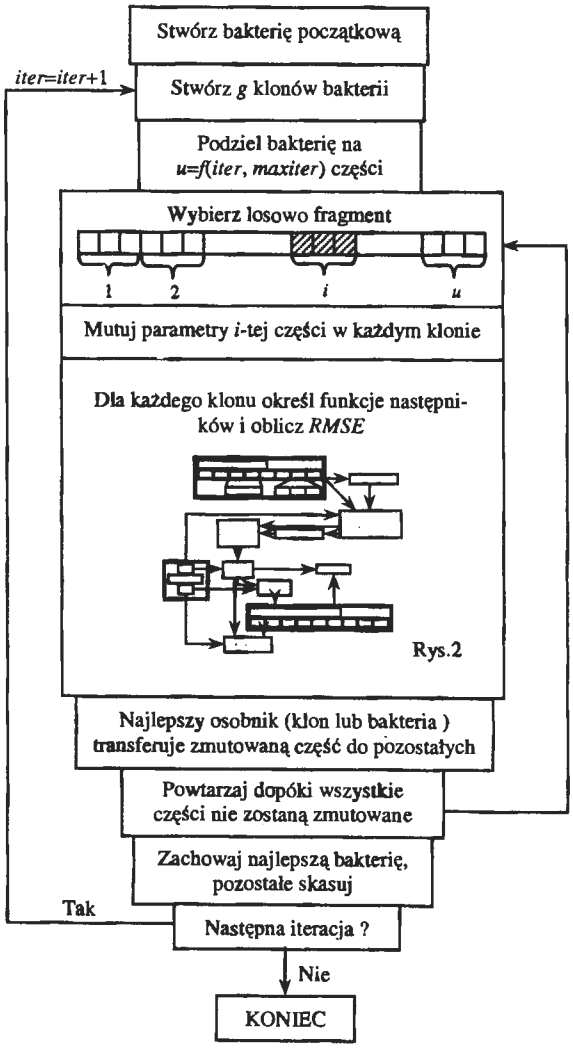
$$RMSE = \sqrt{\frac{\|\mathbf{d}-\mathbf{y}\|^2}{k}} = \sqrt{\frac{\|\mathbf{d}-\mathbf{E}\cdot\mathbf{r}\|^2}{k}} = \sqrt{\frac{\|\mathbf{d}-\mathbf{E}\cdot(\mathbf{E}\backslash\mathbf{d})\|^2}{k}} \qquad (17)$$

Przy zadanej parze uczącej $(X,d)$ macierz $E$, a więc i błąd $RMSE$ będzie zależał tylko od wartości parametrów funkcji przynależności ($3\cdot n\cdot N$ parametrów).

Opisywana modyfikacja algorytmu polega na poddawaniu mutacji bakteryjnej tylko parametrów funkcji przynależności, natomiast parametry funkcji następników dobierane są za pomocą metody najmniejszych kwadratów. Ponadto po każdej mutacji, przeprowadzonej na każdym klonie, przeliczane są tylko wartości funkcji przynależności i siły odpaleń tych reguł, na które muta-

cja miała bezpośredni wpływ. Podejście takie znacznie przyspiesza algorytm.

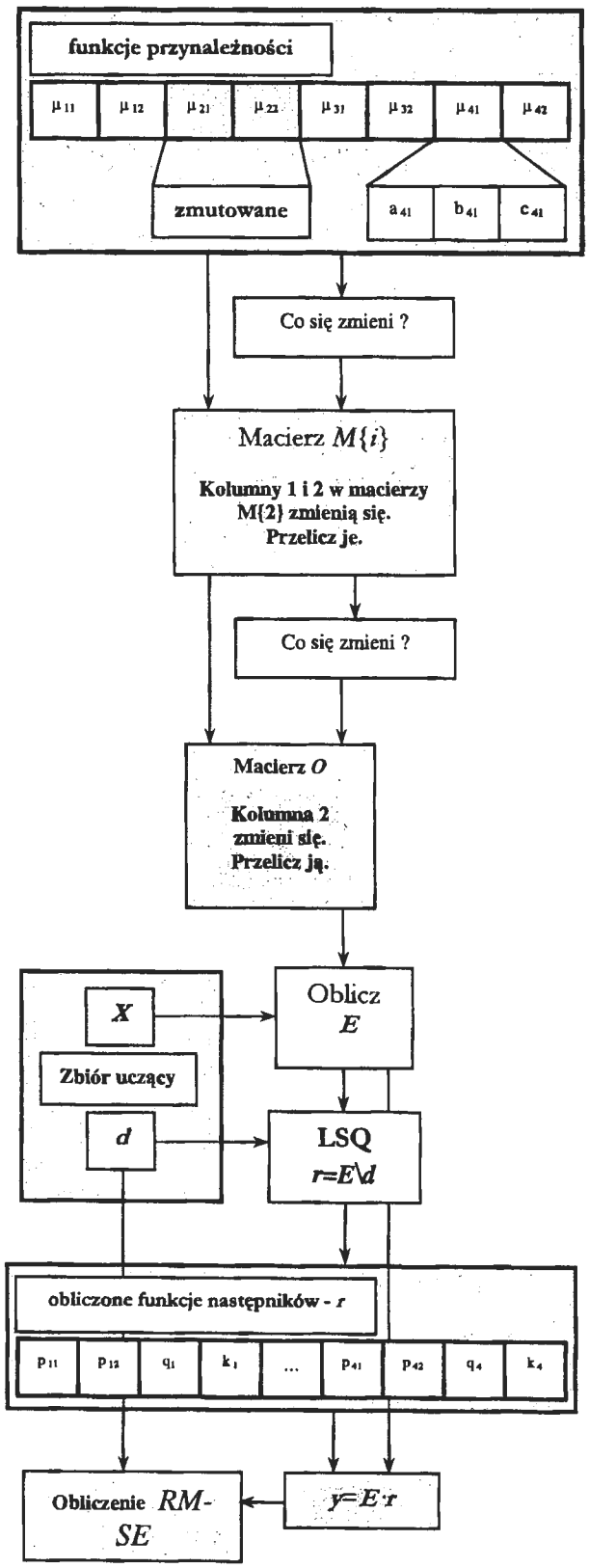
Schemat algorytmu przedstawiony został na rys. 1, którego uzupełnieniem jest rys. 2, pokazujący sposób oceny poszczególnych klonów po mutacji bakteryjnej parametrów funkcji przynależności i po doborze parametrów funkcji następników za pomocą metody najmniejszych kwadratów.




Rys.1. Schemat proponowanego algorytmu.

## Opis algorytmu PBGA+LSQ

*Inicjalizacja* – Po zdefiniowaniu przedziałów dla parametrów funkcji przynależności, należy wygenerować bakterię początkową. W podanym przykładzie bakterię otrzymano z typowej bazy, zawierającej $z^n$ reguł (z-liczba funkcji przynależności równomiernie pokrywających zakres wejściowy). Początkowo te same parametry występują w funkcjach przynależności innych reguł, lecz w trakcie procesu strojenia funkcje te stają się niezależne.




Rys. 2. Sposób obliczania następników i wskaźnika *RMSE* po mutacji wybranej części. Przykład dla 2 wejść i 4 reguł.

*Algorytm* – Diagram proponowanej mutacji bakteryjnej przedstawiony jest na rys. 1. Najpierw tworzonych jest *g* klonów istniejącej bakterii (zakodowanych funkcji przynależności). Następnie bakteria ta dzielona jest na pewną liczbę części, przy czym liczba ta zależy od dopuszczalnej liczby iteracji i numeru aktualnej iteracji.

W proponowanym podejściu długość fragmentów zawierających parametry funkcji przynależności zmienia się tak, że na początku procesu optymalizacji jeden fragment stanowić mogą funkcje przynależności całej reguły bądź kilku reguł, przechodząc stopniowo do fragmentów, na które składają się pojedyncze funkcje przynależności (3 parametry dla ostatnich iteracji). Na tak określonym fragmencie przeprowadza się mutację we wszystkich klonach równocześnie. Kolejnym krokiem jest dobór następników w każdym klonie przy pomocy metody najmniejszych kwadratów. Następnie oceniane są wszystkie klony i bakteria oryginalna, a najlepszy osobnik transferuje zmutowany fragment do pozostałych osobników. Proces wyboru fragmentu, mutacji parametrów funkcji przynależności, dostrajanie następników, ocena i transfer jest powtarzany dopóki wszystkie części nie zostaną zmutowane.

*Zakończenie* – jeśli osiągnięta zostanie zadana liczba iteracji to następuję wyprowadzenie parametrów funkcji przynależności i parametrów funkcji następników. W przeciwnym przypadku następuje kolejna mutacja bakteryjna.

## 4. WYNIKI TESTÓW NUMERYCZNYCH

Przedstawione zostaną teraz trzy eksperymenty mające za zdanie pokazać zdolność modelowania nieliniowości przez system rozmyty z nieciągłymi funkcjami następników, strojony za pomocą zaproponowanego algorytmu PBGA+LSQ.

### 4.1. Przykład 1

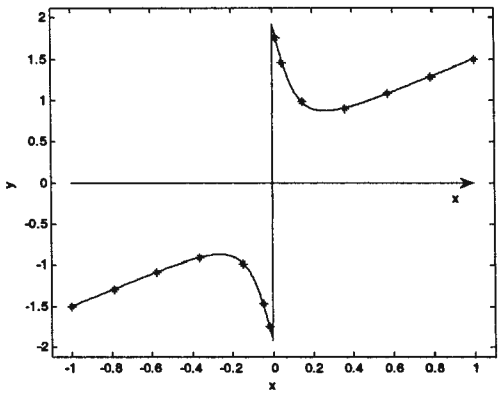
W przykładzie tym zostaną zamodelowane dwie funkcje. Pierwsza z nich opisana zależnością (19), druga w podobny sposób, ale przesunięta o 1 do góry.

Porównując rys. 3 i rys. 4 zauważyć możemy, że asymetria nieciągłości nie ma negatywnego wpływu na jakość modelu, co mogło być błędnie sugerowane przez symetrię funkcji *sign*. Do zamodelowania tych zależności użyty został model $N$=2 regułowy. Wartość uzyskanego błędu RMSE dla obu przypadków była rzędu $10^{-4}$.

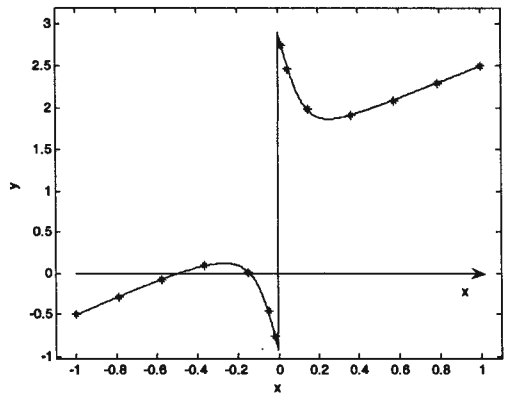
### 4.2. Przykład 2

Kolejny przykład ma na celu pokazanie zdolności modelowania zjawisk ze zmienną wielkością nieciągłości. Modelowana będzie zależność (18)

$$z = f(x,y) = \begin{cases} -x^2 - 2x - 2 & gdzie \quad y < 0 \\ x^2 + 2x + 2 & gdzie \quad y \geq 0 \end{cases} \tag{18}$$




Rys. 3. Uskok symetryczny, *-dane uczące i krzywa generowana przez model.




Rys. 4. Uskok niesymetryczny, *-dane uczące. I krzywa generowana przez model.

Pierwszy model (rys. 5) z ciągłymi funkcjami następników został wygenerowany przy pomocy narzędzia Anfis w Matlabie. Rys. 6. przedstawia powierzchnię wygenerowaną za pomocą algorytmu PBGA+LSQ. Obydwa modele rozmyte zawierały po 4 reguły. Różnica między jakością modelu jest widoczna, co potwierdza wskaźnik *RMSE* umieszczony na górze wykresu.
Ponadto, z rys. 6 wynika zdolność odtwarzania przez model nieciągłości o zmiennej wielkości.

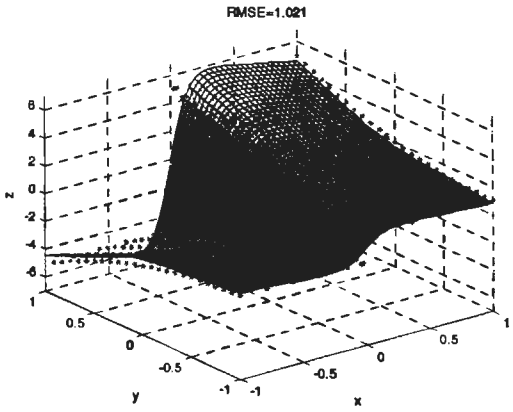
### 4.3. Przykład 3 - model tarcia

Armstrong-Helouvry [1] zaproponowali model tarcia statycznego jako (19)

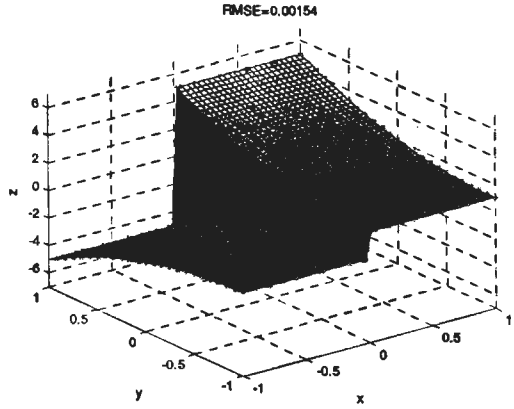
$$F_T = \left( Fc + (Fc - Fs) \cdot e^{-\left(\frac{|v|}{v_s}\right)^{\delta}} + F_v |v| \right) \cdot sign(v) \tag{19}$$

Jeżeli powierzchnia styku jest nieregularna to siła tarcia zależy także od położenia. Dlatego też modelowana siła będzie funkcją zarówno prędkości jak i położenia.

$$F = f(v,x) \tag{20}$$




Rys. 5. Powierzchnia generowana przez model rozmyty z ciągłymi funkcjami następników (ANFIS).




Rys. 6. Powierzchnia generowana przez model z nieciągłymi funkcjami następników, który był strojony przez algorytm PBGA+LSQ.

Zależność ta jest nieciągła dla $v$=0, co pociąga za sobą konieczność stosowania nieciągłych następników w modelu rozmytym, zawierającym reguły:

$$R_i = \text{IF } (v \text{ is } \mu_{i1}) \text{ and } (x \text{ is } \mu_{i2}) \quad \text{THEN } y \text{ is } y_i = f_i(v,x), \tag{21}$$

gdzie funkcje następników

$$f_i(v,x) = p_{i1} \cdot v + p_{i2} \cdot x + q_i + k_i \cdot sign(v) \tag{22}$$

w których składnik $k_i \cdot sign(v)$ jest konsekwencją występowania wspomnianej nieciągłości dla $v$=0.

Zbiór uczący stanowiło 350 trójek postaci $(v,x,F)$, gdzie $F$ dane jest wzorem (23)

$$F = F_T + F_r, \tag{23}$$

w którym $F_T$ jest określone przez (19), a $F_r$ jest siłą reprezentującą nieregularności powierzchni zamodelowana jako (24)

$$F_r = A_r \sin(\omega \cdot x_r - \alpha_o) \tag{24}$$

gdzie: $F_c$=0,5 ; $F_s$=2 ; $F_v$=1 ; $v_s$=0,1 ; $\delta$=1 ; $A_r$=1 ; $\omega$=6 ; $\alpha_0$=0,5$\pi$

Po zakodowaniu parametrów funkcji przynależności $N$=9 reguł rozmytych., otrzymamy bakterię o długości $3 \cdot n \cdot N$=54 parametrów. Liczba iteracji ustawiona została na *maxiter*=150. Podczas iteracji numer *iter*, mutowane były $g$=3 klony, a liczba funkcji przynależności stanowiąca jeden fragment opisana była zależnością (25)

$$l = round\left[\frac{3 \cdot N \cdot n}{3} \cdot \left(1 - \frac{iter}{maxiter}\right)^3 + 1\right] \tag{25}$$

gdzie $round(x)$ – oznacza zaokrąglenie elementu $x$ do najbliższej całkowitej liczby.

Mutacji poddane zostają równocześnie wszystkie parametry wylosowanego fragmentu. Jeśli np. $a$ jest mutowanym parametrem, to w wyniku tej operacji otrzymamy $a'$ [5]

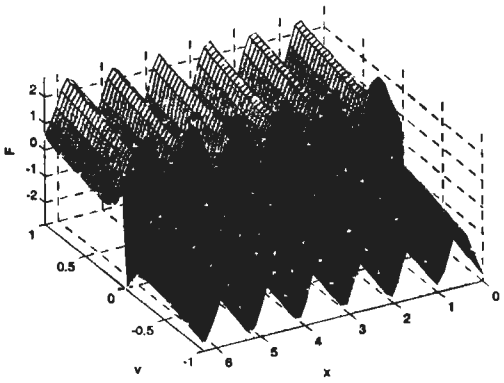
$$a' = \begin{cases} a + \Delta(iter, up(a) - a) & gdy\ wylosowano\ 0 \\ a + \Delta(iter, a - lo(a)) & gdy\ wylosowano\ 1 \end{cases} \tag{26}$$

gdzie $up(a)$ i $lo(a)$ określają granice, w których zmieniać się może mutowany parametr $a$.
Funkcja $\Delta(iter, \alpha)$ jest taka, że operator zaczyna od przeszukiwania całej przestrzeni (gdy *iter* jest małe), a w dalszych etapach przeszukuje tylko otoczenie lokalne. Użyto następującej funkcji:

$$\Delta(iter, \alpha) = \alpha \cdot \beta \cdot \left(1 - \frac{iter}{maxiter}\right)^{\lambda}, \tag{27}$$

gdzie: $\beta$ – liczba wylosowana z przedziału <0,1>, $\lambda$ określa szybkość zawężania przeszukiwanej przestrzeni.

Błąd otrzymany po 150 iteracjach wyniósł 0,049. Proces optymalizacji trwał około 14s. Powierzchnia wygenerowana przez model i zmiana błędu RMSE w trakcie uczenia zamieszczone zostały na rys. 7 i 8.

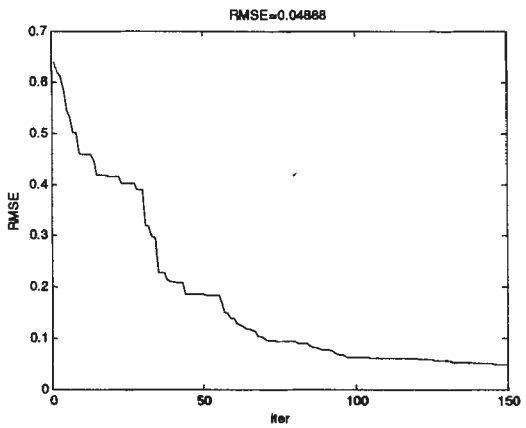



Rys. 7. Dane uczące i powierzchnia wygenerowana przez model.

## 5. WNIOSKI

Otrzymane wyniki dowodzą że:

- zjawiska nieciągłe mogą być modelowane za pomocą systemów rozmytych z nieciągłymi funkcjami następników,
- asymetria i wielkość „uskoku" nie mają negatywnego wpływu na jakość modelu strojonego algorytmem PBGA+LSQ,
- zaproponowany algorytm PBGA+LSQ jest efektywny w strojeniu parametrów modeli rozmytego typu Sugeno.




Rys.8. Wykres błędu uczenia.

### FUZZY MODELING OF FRICTION BY PSEUDO-BACTERIAL AND LEAST SQUARE METHOD

**Abstract:** In this paper a new method of tuning parameters of Sugeno fuzzy rules is presented. Because modeled phenomenon is discontinuous, new type of discontinuous consequent function was introduced. Proposed algorithm (PBGA+LSQ) combines pseudo-bacterial algorithm (PBGA) for tuning parameters of membership functions and least square method (LSQ) for parameters of consequent functions.

### Literatura

[1] Armstrong-Helouvry B., Dupont P., Canudas de Wit C. (1994) A Survey of models, analysis tools and compensation methods for the control of machines with friction. *Automatica*, **30**, 1083-1153.

[2] Botzheim J., Hámori B., Kóczy L.T. (2001) *Applying Bacterial Algorithm to Optimise Trapezoidal Membership Functions in a Fuzzy Rule Base*, Dept. of Telecommunication and Telematics, Budapest University of Technology and Economics.

[3] Jastrzębski M., Kabziński J. (2004) *Optimisation of Takagi-Sugeno Fuzzy Models by Pseudo-Bacterial Genetic Algorithm* IEEE International Conference on Methods and Models in Automation and Robotics (MMAR 2004), 1411-1416.

[4] Kabziński (2004) J. *Adaptive Control of Linear Permanent Magnet Motors with Fuzzy Modeling of Disturbance Forces*, IEEE International Conference on Methods and Models in Automation and Robotics (MMAR 2004), 393-397.

[5] Michalewicz Z. (1999) *Algorytmy genetyczne + struktury danych = programy ewolucyjne*, WNT, Warszawa.

[6] Nawa N. E., Furuhashi T. (1997) A Study on Nonlinear Model Identification using Pseudo-Bacterial Genetic Algorithm, *Proc. of the 1997 International Conference on Neural Information Processing and Intelligent Information System*, 408-411.

[7] Salmeri M., Re M., Petrongari E., Cardarilli G.C. (1999) A Novel Bacterial Algorithm to Extract the Rule Base from a Training Set. *Fuzzy Systems, 2000. FUZZ IEEE 2000. The Ninth IEEE International Conference*, **2**, 759-761.

Instytut Badań Systemowych
Polskiej Akademii Nauk

ISBN 83-89475-00-6