

# Algorytm implementacji regulatora rozmytego o niskim koszcie numerycznym

**Streszczenie.** Niniejszym referacie przedstawiono nowy algorytm implementacji systemu rozmytego. Jego idea polega na zastosowaniu macierzowej formy regulatora i zastosowaniu dodatkowej warstwy identyfikującej aktualny sektor pracy. Dzięki zastosowanemu podejściu udało się istotnie zmniejszyć ilość operacji matematycznych koniecznych dla wyznaczenia wartości wyjściowej algorytmu. W efekcie pozwoliło to na zmniejszenie długości czasu koniecznego na obliczenie wartości sterowania w układzie rzeczywistym. Rozważania teoretyczne potwierdzono badaniami symulacyjnymi w pakiecie Matlab-Simulink, a następnie zweryfikowano eksperymentalnie z wykorzystaniem pakietu DSpace 1103.

**Abstract.** In the paper a novel implementation algorithm for a fuzzy controller is presented. It is based on the matrix form of the controller and application of the additional layer which identify the current state of the object. The proposed algorithm reduces the mathematical operation needed for computation of the controller output significantly. It results on the faster computational time of the implementation of the fuzzy controller. Theoretical considerations were confirmed by simulation study in Matlab-Simulink, and have been verified experimentally using DSpace 1103 package. **A novel implementation algorithm for a fuzzy controller based on the matrix form of the controller**

**Słowa kluczowe:** regulator neuronowo-rozmyty, optymalizacja numeryczna, obniżony koszt obliczeniowy, tranzycja Petriego  
**Keywords:** neuro-fuzzy controller, numerical optimization, low computational effort, Petri transition

doi:10.12915/pe.2014.11.60

## Wstęp

W miarę rozwoju automatyzacji, układom regulacji stawiane są coraz większe wymagania [1] dotyczące ich właściwości statycznych i dynamicznych. Bardzo duża liczba prac skupia się na poprawie jakości regulacji poprzez opracowywanie coraz to nowszych i bardziej zaawansowanych algorytmów sterowania. Jako przykład można wymienić regulatory predykcyjne [2], [3] adaptacyjne [5], [4], [5] czy rozmyte [6], [7], [8]. Zwłaszcza te ostatnie cieszą się stale rosnącym zainteresowaniem. Wiele ośrodków na świecie stara się poprawić działanie klasycznych regulatorów rozmytych poprzez modyfikację ich struktury. Są to między innymi rozwiązania polegające na wprowadzeniu dodatkowych sprzężeń rekurencyjnych [9], [10] czy zastąpienie zbiorów typu I zbiorami typu II [4], [11], [12]. Klasyczną i najprostszą możliwością zwiększenia nieliniowości powierzchni sterowania jest zwiększenie liczby reguł. Podejście takie jednak napotyka na istotny problem. Wzrost liczby reguł przekłada się na zwiększenie złożoności obliczeniowej regulatora rosnącej w tempie wykładniczym. Prowadzi to do zwiększenia kosztów numerycznych takiego rozwiązania. Jedną z metod, umożliwiającą redukcję złożoności obliczeniowej regulatora, jest jego implementacja *off-line* w sposób analogiczny do algorytmów predykcyjnych [13]. Utrudnia to jednak wprowadzenie bezpośredniej adaptacji do struktury regulatora i z tego powodu jest to rzadko stosowane podejście.

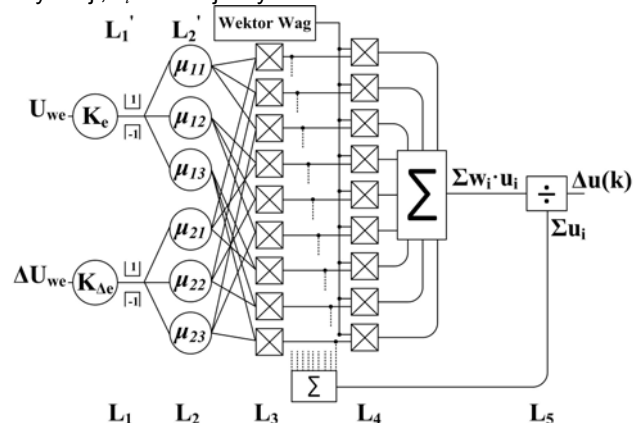
W niniejszej pracy proponuje się rozwiązanie pozwalające na istotne zmniejszenie kosztów numerycznych implementacji regulatora neuronowo-rozmytego. Dzięki temu, możliwe jest znaczne zwiększenie rozmiarów bazy reguł, przy niewielkim wzroście nakładu obliczeniowego. Obniżenie kosztu numerycznego uzyskano poprzez zastosowanie rozwiązania znanego z Sieci Petriego [14], [15]. Dzieląc płaszczyznę sterowania na sektory, przy założeniu odcinkowo-liniowych funkcji wejściowych (w pracy jako przykład przyjęto zbiory trójkątne), możliwym stało się obliczanie wartości sygnału sterującego jedynie na podstawie części bazy reguł odpowiadającej aktualnemu sektorowi pracy. Proponowany algorytm nie wpływa w żadnym stopniu na dokładność sterowania. Zauważyć tu można analogię do wykonania instrukcji danego miejsca – aktywacja tranzycji wskazuje reguły konieczne do wyznaczenia aktualnego sterowania.

## Regulator rozmyty

Na rys. 1. przedstawiono schemat blokowy rozmytego regulatora neuronowo-rozmytego [6]. Baza reguł składa się z reguł typu IF-THEN o następującej postaci:

$$R_j: \text{IF } x_1 \text{ IS } A_1^j \text{ AND } x_2 \text{ IS } A_2^j \text{ THEN } y = u_i,$$

gdzie:  $x_i$  – wejście systemu,  $A_i^j$  – wejściowa funkcja aktywacji,  $u_i$  – funkcja wynikowa.



Rys. 1. Schemat blokowy regulatora neuronowo-rozmytego

W analizowanym regulatorze można wyróżnić 5 warstw. W warstwie  $L_1$ ,  $L_1'$  następuje normalizacja wartości sygnałów wejściowych. Sygnały te są skalowane oraz ograniczane zgodnie z równaniami (1, 2).

$$(1) \quad \bar{L}_1 = \Delta U_{we} \cdot K_{\Delta e}$$

$$(2) \quad \bar{L}_1' = U_{we} \cdot K_e$$

Przy założonych ograniczeniach  $\max\{\bar{L}_1\} = 1$ ,  $\max\{\bar{L}_1'\} = 1$ .

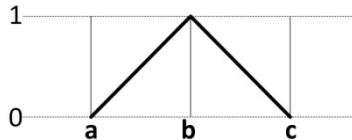
Następnie w warstwach  $L_2$ ,  $L_2'$ , zwanych warstwami rozmywania, budowane są wektory poziomów aktywacji reguł przynależności dla każdego z wejść. Wektory te dają się zdefiniować jako (3, 4).

$$(3, 4) \quad \bar{L}_2 = \begin{bmatrix} \mu_{11} \\ \mu_{12} \\ \mu_{13} \end{bmatrix} \quad \bar{L}_2' = \begin{bmatrix} \mu_{21} \\ \mu_{22} \\ \mu_{23} \end{bmatrix}$$

gdzie  $\mu_{mn} = f_{aktywacji\_n}(L_m)$ .

W niniejszej pracy przyjęto jako funkcje wejściowe zbiory trójkątne. Są one opisane za pomocą równania (5). Kształt takiej funkcji zaprezentowano na rys. 2.

$$(5) \quad \mu_{mn} = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right)$$



Rys. 2. Trójkątna funkcja przynależności

gdzie  $a, b, c$  – parametry trójkątnej funkcji przynależności, odpowiednio, położenie początku, wierzchołka oraz końca funkcji.

W warstwie piątej, zgodnie ze wzorem (6), wyznaczane są poziomy spełnienia przesłanek.

$$(6) \quad [L_3] = \overline{L_2} * \overline{L_2}^T$$

gdzie \* oznacza  $t$ -normę (w artykule przyjęto jej najczęściej stosowaną postać - *prod*)

Warstwa  $L_4$  realizuje implikacje rozmytą (w pracy założono typ Larsena) i może być opisana za pomocą następującego wyrażenia:

$$(7) \quad [L_6] = [L_5] \bullet [W]$$

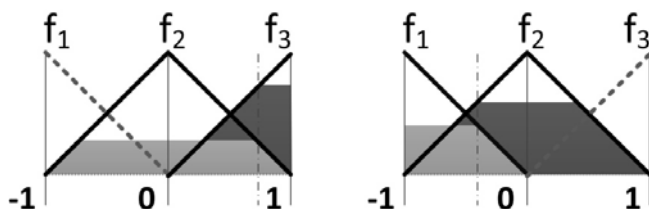
Gdzie:  $\bullet$  – iloczyn Hadamara,  $[W]$  - macierz współczynników wagowych taka że,

$$(8) \quad [W] = \begin{bmatrix} w_{11} & w_{12} & w_{31} \\ w_{21} & w_{22} & w_{32} \\ w_{31} & w_{32} & w_{33} \end{bmatrix} = \begin{bmatrix} w_{NB-NB} & w_{NB-ZE} & w_{NB-PB} \\ w_{ZE-NB} & w_{ZE-ZE} & w_{ZE-PB} \\ w_{PN-NB} & w_{PB-ZE} & w_{PB-PB} \end{bmatrix}$$

Warstwa  $L_7$  odpowiada za proces defuzyfikacji. Odbyna się ona według algorytmu defuzyfikacji opisanego poniższym równaniem.

$$(9) \quad L_5 = \frac{\sum L_4}{\sum L_3} = \Delta u(k)$$

Należy zauważyć, że dla dowolnych wartości wejściowych nie więcej niż dwie funkcje przynależności, dla każdego z wejść, są jednocześnie aktywne. Fakt ten zobrazowano na rys. 3. Linia przerywaną zaznaczono funkcję o zerowym poziomie aktywacji w danym przypadku.



Rys. 3. Przykładowe poziomy aktywacji funkcji przynależności w zależności od wartości sygnału wejściowego

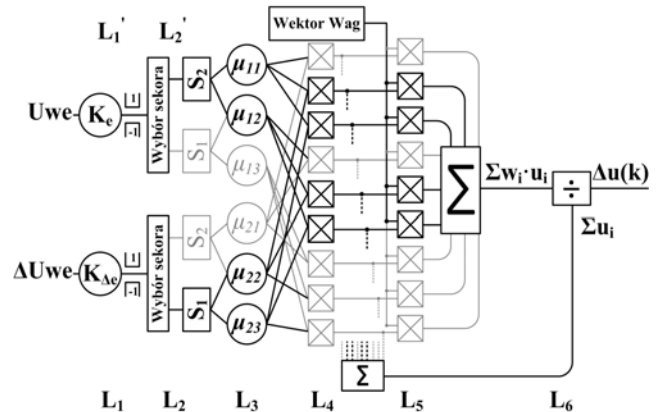
W klasycznej implementacji algorytmu systemu rozmytego przeprowadza się wszystkie operacje matematyczne, również w przypadku mnożenia elementów zerowych. W niniejszej pracy proponuje się algorytm, który wykrywa

aktualny sektor pracy systemu rozmytego i pomija (nie przeprowadza obliczeń) obszary które nie są aktywne. Pozwala to na zaoszczędzenie mocy obliczeniowej, a tym samym implementację regulatorów rozmytych o bardziej rozbudowanych bazach reguł w układach o większych wymaganiach czasowych, lub z użyciem słabszych a tym samym tańszych układów mikrokontrolerów.

### Idea regulatora

W odróżnieniu od klasycznego regulatora neuronowo-rozmytego (rys. 1), w proponowanym układzie wprowadzono dwie dodatkowe warstwy. W warstwach tych, na podstawie wartości sygnałów wejściowych, odbywa się identyfikacja sektora pracy, w którym aktualnie znajduje się układ. Następnie aktywowane są jedynie funkcje przynależności odpowiadające rozpoznanemu sektorowi. Widoczna jest tu analogia do Sieci Petriego gdzie dana tranzycja aktywna jest jedynie przy spełnionych warunkach wejściowych. Podobnie w zaproponowanym rozwiązaniu, wyznaczenie wartości poszczególnych funkcji przynależności jest uwarunkowane obecnością systemu w danym sektorze. Znamiennym dla zaproponowanego rozwiązania jest fakt znacznie mniejszego obciążenia numerycznego. W przypadku regulatora o trójkątnych funkcjach przynależności jednocześnie nie więcej niż dwie funkcje wejściowe są aktywne. Pozostałe funkcje przyjmują wartości zerowe. Wartości te dalej oddziałują na kolejne warstwy systemu neuronowo-rozmytego. W przypadku regulatora o 9 regułach co najwyżej cztery reguły posiadają niezerowe konkluzje. Oznacza to, że 5 reguł ma zerowe poziomy aktywacji. W klasycznym podejściu przeprowadza się obliczenia wszystkich reguł zarówno zerowych jak i niezerowych. Dzięki zastosowaniu identyfikacji sektora w jakim obecnie znajduje się układ można wyznaczać wartość sterowania poprzez obliczenie jedynie niezerowych reguł.

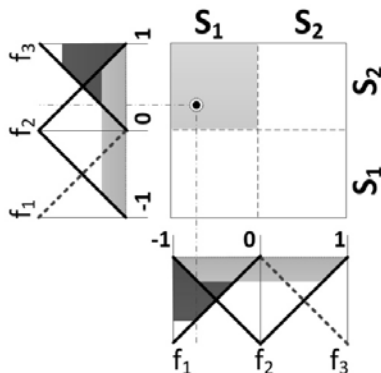
Na rys. 4 przedstawiono schemat blokowy proponowanego regulatora. Dla wartości sygnałów wejściowych  $K_e=0.2$  oraz  $K_{de}=-0.7$ , w przypadku regulatora o 9 regułach, jedynie jeden z 4 sektorów będzie aktywny. Zaznaczono to na rys. 5.



Rys. 4. Schemat blokowy regulatora z dodatkowymi warstwami

Można zauważyć, że dla przypadku wartości wejściowej  $K_e$  dodatniej, nie ma potrzeby wyznaczanie wartości funkcji przynależności obejmujących dolną (ujemną) połowę płaszczyzny sterowania. Podobnie w przypadku wejścia  $K_{de}$  o wartości ujemnej nie uzasadnione jest obliczanie wartości funkcji z prawej (dodatniej) połowy półpłaszczyzny, funkcje te będą z definicji równe zero. Zauważając tę prawidłowość, łatwo stwierdzić, iż sensownym staje się analizowanie jedynie części, w tym przypadku 1/4 powierzchni sterowania. Co istotne, w przypadku regulatora o 25 czy 49 regułach analizie należy poddać odpowiednio 1/16 i 1/36 sektorów całej płaszczyzny sterowania.

W ogólności będzie to  $1/(n-1)^2$  sektorów, gdzie  $n$  to ilość funkcji przynależności przydzielonych do każdego z dwóch wejść regulatora (przy założeniu jednakowej ilości reguł na każdym z wejść). Na schemacie blokowym (rys. 4.) na szaro zaznaczono elementy systemu neuronowo-rozmytego niewymagające obliczeń ze względu na zerowe poziomy przynależności sygnałów wejściowych do zbiorów wejściowych.



Rys. 5 Sektorowy podział powierzchni sterowania analizowanego systemu

Poniżej przedstawiono procedurę przeprowadzania obliczeń w proponowanym regulatorze rozmytym.

Warstwa  $L_1$  jest warstwą wejściową. Sygnały są w niej skalowane zgodnie z wartościami współczynników  $K_e$  oraz  $K_{de}$ .

Warstwa  $L_2$  to warstwa wyboru sektora w którym znajduje się w danej chwili czasowej sygnał wejściowy. Daje się wyróżnić  $n-1$  sektorów, gdzie  $n$  to liczba funkcji przynależności zdefiniowana dla każdego wejścia.

Warstwa  $L_3$  to warstwa rozmywania. Działa ona analogicznie do odpowiadającej warstwy z regulatora klasycznego ( $L_2$  rys.1) jednak wyznaczane są wartości jedynie w czterech, a nie wszystkich 6 funkcji przynależności. Założone postacie funkcji są identyczne co oznacza, że dają się opisać tymi samymi wzorami oraz parametrami. W przypadku spełnienia warunku uzupełnienia do jedności funkcji przynależności można dalej uprościć algorytm obliczeniowy. W takim przypadku należy wyznaczyć jedynie jeden stopień przynależności do zbioru wejściowego. Stopień przynależności drugiego zbioru rozmytego wyznaczyć jako uzupełnianie do jedności.

Warstwa  $L_4$  to warstwa przesłanek. Wyliczane są w niej poziomy aktywacji poprzedników poszczególnych reguł. Podobnie jak i w przypadku warstwy  $L_3$ , wyznaczenie poziomów aktywacji należy przeprowadzić jedynie dla 4 zamiast 9 reguł. Poziom aktywacji jest wyznaczany z następującej zależności (przy operatorze  $t$ -normy typu  $prod$ ).

$$(10) \quad u_i = \mu_{1n_1} \cdot \mu_{2n_2}$$

gdzie:  $n_1, n_2 \in [1, 2]$

Warstwa  $L_5$  to warstwa inferencji, poziomy aktywacji aktywnych reguł są mnożone przez odpowiadające im współczynniki wagowe.

Warstwa  $L_6$  jest warstwą defuzyfikacji. Zgodnie ze wzorem (11), wyznaczana jest tu wartość sygnału sterującego.

$$(11) \quad \Delta u_k = \frac{\sum_{i=1}^4 w_i \cdot u_i}{\sum_{i=1}^4 u_i}$$

Dla zachowania optymalności objętości kodu zdecydowano się na zachowanie 4 funkcji przynależności oraz 4 reguł w przypadku położenia skrajnych sygnałów wejściowych. Możliwym było by zastosowanie jedynie dwóch lub nawet jednej reguły dla położenia charakterystycznych ( $u_{we} = \{-1, 0, 1\}$ ) odpowiednio jednego lub obu sygnałów wejściowych, jednak algorytm wymagał by dodatkowego rozpoznawania takich sytuacji. Ponieważ taka sytuacja, szczególnie w przypadku prawidłowo sterowanych układów rzeczywistych występuje sporadycznie, zdecydowano się nie wyróżniać jej w proponowanym algorytmie.

### Analiza złożoności obliczeniowej klasycznego oraz proponowanego rozwiązania

W Tab. 1 zestawiono porównanie złożoności obliczeniowej klasycznego regulatora neuronowo-rozmytego oraz systemu wylazanego zgodnie z proponowanym algorytmem. Koszty numeryczne wyznaczono na podstawie zależności zdefiniowanych w poprzednich sekcjach pracy.

Analizując dane zawarte w Tab. 1 można zauważyć, że w przypadku systemu o 9 regułach ilość operacji matematycznych klasycznego algorytmu daje się opisać jako (12) a proponowanego systemu (13). Zmniejsza się koszt wynikający z konieczności obliczania mniejszej ilości reguł przynależności oraz poziomów aktywacji reguł, pojawia się jednak koszt wynikający z konieczności identyfikacji sektora pracy. Zaleta proponowanego rozwiązania uwidacznia się w przypadku systemu o większej liczbie reguł. Tak więc dla układu o 49 regułach liczba operacji w rozwiązaniu klasycznym wynosi (14) a proponowanym rozwiązaniu nadal jedynie (12), przy nieznacznie zwiększonym koszcie identyfikacji sektora. Koszt identyfikacji sektora odpowiada kosztowi funkcji warunkowych pozwalających na zdefiniowanie aktualnego sektora na podstawie wartości sygnałów wejściowych i wyznaczenia parametrów funkcji przynależności mających zostać użytymi w procesie wyznaczenia wartości wyjściowej w danej iteracji algorytmu. Oznacza to znaczną redukcję złożoności obliczeniowej algorytmu.

$$(12) \quad K_{reg\_3x3\_CLASSIC} = \sum_{n=1}^{n=5} K_{Ln} = \underbrace{2 \cdot (K_m + K_{sat})}_{L_1+L_1} + \underbrace{6 \cdot K_{\mu\Delta}}_{L_2+L_2} + \underbrace{2 \cdot (9 \cdot K_m)}_{L_3+L_6} + \underbrace{2 \cdot (8 \cdot K_s)}_{L_5} + K_d$$

$$(13) \quad K_{reg\_3x3\_NEW} = \sum_{n=1}^{n=6} K_{Ln} = \underbrace{2 \cdot (K_m + K_{sat})}_{L_1+L_1} + \underbrace{2 \cdot K_{id} + 4 \cdot K_{\mu\Delta}}_{L_2+L_2} + \underbrace{2 \cdot (4 \cdot K_m)}_{L_4+L_5} + \underbrace{2 \cdot (3 \cdot K_s)}_{L_6} + K_d$$

$$(14) \quad K_{reg\_7x7\_CLASSIC} = \sum_{n=1}^{n=5} K_{Ln} = \underbrace{2 \cdot (K_m + K_{sat})}_{L_1+L_1} + \underbrace{14 \cdot K_{\mu\Delta}}_{L_2+L_2} + \underbrace{2 \cdot (49 \cdot K_m)}_{L_3+L_4} + \underbrace{2 \cdot (48 \cdot K_s)}_{L_5} + K_d$$

gdzie:  $K_m$  – numeryczny koszt mnożenia,  $K_{sat}$  – koszt bloku nasycenia,  $K_{id}$  – koszt identyfikacji sektora,  $K_{\mu\Delta}$  – koszt trójkątnej funkcji aktywacji,  $K_s$  – koszt sumowania,  $K_d$  – koszt dzielenia

W celu weryfikacji praktycznej przedstawionych powyżej wylczeń, przeprowadzono badania symulacyjne i eksperymentalne analizowanych rozwiązań. Jako pierwsze wykonano badania symulacyjne klasycznych oraz proponowanych algorytmów regulatorów rozmytych. Czas realizacji algorytmów w badaniach symulacyjnych i eksperymentalnych (w badaniach eksperymentalnych wykorzystano kartę z procesorem sygnałowym DS. 1103)

zamieszczono w Tab. 2. Dodatkowo na rys. 6. zamieszczono czas wykonywania operacji badanych algorytmów w przypadku realizacji rzeczywistej.

Tabela 1. Porównanie złożoności numerycznej klasycznego regulatora neuronowo-rozmytego oraz zaproponowanej modyfikacji

| Regulator Klasyczny                                       |          | Proponowany algorytm |  |
|---|----------|----------------------|--|
| Liczba zmiennych  |          |                      |  |
| Mniej zmiennych   |          | Więcej zmiennych     |  |
| Wyznaczanie aktualnego sektora                            |          |                      |  |
| Nie   |          | Tak                  |  |
| Obliczanie wartości funkcji przynależności                |          |                      |  |
|   |          | Liczba reguł         |  |
| 6 funkcji   | 9 reguł  | 4 funkcje            |  |
| 10 funkcji  | 25 reguł | 4 funkcje            |  |
| 14 funkcji  | 49 reguł | 4 funkcje            |  |
| Obliczanie poziomów aktywacji reguł                       |          |                      |  |
| 9 mnożeń  | 9 reguł  | 4 mnożenia           |  |
| 25 mnożeń   | 25 reguł | 4 mnożenia           |  |
| 49 mnożeń   | 49 reguł | 4 mnożenia           |  |
| Sumowanie sygnałów do defuzyfikacji (dwukrotnie) $\sum_n$ |          |                      |  |
| n=9   | 9 reguł  | n=4                  |  |
| n=25  | 25 reguł | n=4                  |  |
| n=49  | 49 reguł | n=4                  |  |
| Mnożenie przez współczynniki wagowe                       |          |                      |  |
| 9 mnożeń  | 9 reguł  | 4 mnożenia           |  |
| 25 mnożeń   | 25 reguł | 4 mnożenia           |  |
| 49 mnożeń   | 49 reguł | 4 mnożenia           |  |

Tabela 2. Czasy obliczeń algorytmów

| Średni czas potrzebny na 100-krotne policzenie algorytmu w Simulinku przy jednakowych parametrach symulacji [s] |              |               |
|---|--------------|---------------|
| Klasyczny   | Liczba reguł | Zmodyfikowany |
| 43.44   | 9 reguł      | 22.82         |
| 74.92   | 25 reguł     | 22.80         |
| 117.05  | 49 reguł     | 23.23         |
| Czas wykonywania algorytmów na karcie procesorowej Dspace 1103 [s]  |              |               |
| 1.8032e-006   | 9 reguł      | 2.0781e-006   |
| 2.7844e-006   | 25 reguł     | 2.0784e-006   |
| 3.6578e-006   | 49 reguł     | 2.4594e-006   |
| 4.7256e-006   | 81 reguł     | 2.0869e-006   |
| 2.3432e-005   | 961 reguł    | 2.5404e-006   |

Jak wynika z danych zamieszczonych w Tab. 2 oraz na rys. 6, czas potrzebny na realizację tradycyjnego układu systemu rozmytego zwiększa się znacząco wraz ze wzrostem liczby reguł. Przykładowo dla systemu z 3 funkcjami przynależności dla każdego wejścia wynosi on 1.8032e-006s a dla układu o 31 funkcjach 2.3432e-005s. Oznacza to w przybliżeniu liniowy wzrost nakładu obliczeniowego potrzebnego na wyznaczenie wyjścia, co zostało pokazane za pomocą prostej aproksymacyjnej pierwszego rzędu. W proponowanym algorytmie o zmniejszonej złożoności obliczeniowej wzrost ilości funkcji wejściowych w praktycznie nie przekłada się na wzrost złożoności obliczeniowej algorytmu.

### Podsumowanie

Artykule przedstawiono zagadnienia związane z analizą kosztów numerycznych regulatorów rozmytych. Na podstawie przeprowadzonych badań można wyciągnąć następujące wnioski podsumowujące:

- Zastosowanie dodatkowej warstwy identyfikującej aktualny sektor pracy układu, a następnie wyznaczanie sterowania na podstawie jedynie części płaszczyzny sterowania pozwoliło na zmniejszenie złożoności numerycznej algorytmu regulatora neuronowo-rozmytego.
- W przypadku rozwiązania klasycznego, czas konieczny na wyznaczenie wartości sterowania na wyjściu regulatora wzrasta wraz z rozmiarem bazy reguł.

- Istotną zaletą zaproponowanego algorytmu jest względna stałość, a tym samym przewidywalność czasu koniecznego na obliczenia, niezależnie od rozmiaru bazy reguł.

- Redukcja złożoności obliczeniowej regulatora rozmytego o trójkątnych funkcjach przynależności nie wpływa na jakość sterowania systemu rozmytego.

Praca finansowana przez Narodowe Centrum Nauki w ramach projektu Adaptacyjne sterowanie rozmyte złożonego układu napędowego o zmiennych parametrach, 2012-2015, UMO-2011/03/B/ST7/02517

### LITERATURA

- [1] Brock S., Zawirski K., New approaches to selected problems of precise speed and position control of drives, *IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society*, (2012), 6291 - 6296
- [2] Cychowski M., Serkies P., Nalepa R., Szabat K., Model predictive speed and vibration control of dual-inertia PMSM Drives, *IEEE International Symposium on Industrial Electronics*, (2011), 1919 - 1924
- [3] Thomsen S., Hoffmann N., Fuchs F.W., PI Control, PI-Based State Space Control, and Model-Based Predictive Control for Drive Systems With Elastically Coupled Loads—A Comparative Study, *IEEE transactions on industrial electronics*, 58, (2011), nr 8, 3647 - 3657
- [4] Knychas S., Szabat K., Zastosowanie adaptacyjnego regulatora opartego na zbiorach rozmytych typu II do sterowania prędkością układu napędowego, *Przegląd Elektrotechniczny*, 87, (2011), nr 4, 160 - 163
- [5] Orłowska-Kowalska T., Dybkowski M., Szabat K., Adaptive Sliding-Mode Neuro-Fuzzy Control of the Two-Mass Induction Motor Drive Without Mechanical Sensors, *IEEE Trans. on Industrial Electronics*, 57, (2010), nr 2, 553 - 564
- [6] Brock S., User-friendly design method for fuzzy speed controller for servo drives, *Proceedings of the First Workshop on Robot Motion and Control*, (1999), 17 - 21
- [7] Zawirski K., Urbanski K., Application of fuzzy control technique to thyristor DC drive, *Proceedings of the IEEE International Symposium on Industrial Electronics*, 3, (1997), 1173 - 1178
- [8] Orłowska-Kowalska T., Szabat K., Jaszczak K., The influence of parameters and structure of PI-type fuzzy-logic controller on DC drive system dynamics, *Fuzzy Sets and Systems*, 131 (2002), 251 - 264
- [9] Knychas S., Szabat K., Adaptacyjny układ sterowania z rekurencyjnymi regulatorami rozmytymi dla układu napędowego o zmiennych parametrach, *Przegląd Elektrotechniczny*, 89, (2013), nr 6, 155 - 158
- [10] Wai R.-J., Liu C., Experimental Verification of Dynamic Petri Recurrent-Fuzzy-Neural-Network Path Tracking Control for Mobile Robot, *IEEE International Conf. on Control and Automation*, (2009), 1359 - 1364
- [11] Liao Q., Li N., Li S., Type-II T-S Fuzzy Model-based Predictive Control, *Joint 48<sup>th</sup> IEEE Conference on Decision and Control and 28<sup>th</sup> Chinese Control Conference Shanghai*, (2009), 16 - 18
- [12] Linda O., Manic M., Uncertainty-Robust Design of Interval Type-2 Fuzzy Logic Controller for Delta Parallel Robot, *IEEE transactions on industrial informatics*, 7, (2011), nr. 4, 661 - 669
- [13] Kamide Y., Zanma T., Ishida M., Position control using off-line model predictive control in piecewise affine system experimental verification, *10<sup>th</sup> IEEE International Workshop on Advanced Motion Control*, (2008), 170 - 175
- [14] Wai R.-J., Chu C.-C., Motion Control of Linear Induction Motor via Petri Fuzzy Neural Network, *IEEE Transactions on industrial electronics*, 54, (2007), nr 1, 281 - 295
- [15] Derugo P., Dybkowski M., Szabat K., Zastosowanie adaptacyjnego neuronowo-rozmytego regulatora prędkości z konkurencyjnymi warstwami Petriego do sterowania silnika elektrycznego, *Przegląd Elektrotechniczny*, 89, (2013), nr 12, 64 - 67

**Autorzy:** mgr. inż. Piotr Derugo, dr hab. inż. Krzysztof Szabat, Politechnika Wroclawska, Instytut Maszyn Napędów i Pomiarów Elektrycznych, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, E-mail: piotr.derugo@pwr.wroc.pl, krzysztof.szabat@pwr.wroc.pl