

**Tomasz BURATOWSKI**

AKADEMIA GÓRNICZO-HUTNICZA, WYDZIAŁ INŻYNIERII MECHANICZNEJ I ROBOTYKI, KATEDRA ROBOTYKI I DYNAMIKI MASZYN

**Zastosowanie logiki rozmytej w modelowaniu dynamiki układów mechanicznych****Dr inż. Tomasz Buratowski**

Pracuje w Katedrze Robotyki i Dynamiki Maszyn AGH. Jego zainteresowania skupiają się na mechatronice oraz wykorzystaniu sztucznej inteligencji. Jest autorem prac nad zastosowaniem powyższych technik do zadań związanych z modelowaniem, identyfikacją i sterowaniem mobilnymi robotami.

tburatow@agh.edu.pl

**Streszczenie**

W artykule przedstawiono sposób wykorzystania algorytmów obliczeniowych opartych na logice rozmytej do modelowania układów dynamicznych. Do prowadzenia obliczeń symbolicznych przyjęto programy Maple™. Natomiast obliczenia numeryczne wykonano w środowisku programu Matlab™-Simulink. Szczegółowo przedstawiono sposób budowania modeli dynamicznych z logiką rozmytą na przykładzie mobilnego robota.

**Abstract**

This paper discusses the mode in which algorithms based on fuzzy logic have been used to model dynamic systems. For symbolic computations Maple™ programme has been applied. Numerical calculations have been done in the Matlab™-Simulink programme environment. The mode of constructing dynamic systems with the use of fuzzy logic has been presented in a detailed way with a mobile robot as an example.

**1 Wstęp**

W niniejszej pracy wykorzystano zbiory rozmyte oraz algorytmy genetyczne w procesie identyfikacji. Teoria Zbiorów rozmytych służy do opisu zjawisk i pojęć, które mają charakter wieloznaczny i nieprecyzyjny. Znanie wcześniej metody matematyczne, wykorzystujące klasyczną teorię zbiorów i logikę dwuwartościową nie były w stanie rozwiązać tego typu problemów [1,2,3]. Jak wcześniej przedstawiono za pomocą zbiorów rozmytych można formalnie określić pojęcia nieprecyzyjne i wieloznaczne takie jak „duża prędkość“, „mała prędkość“. W projektowaniu zbiorów rozmytych najważniejsze jest określenie tzw. zbioru rozważań. W przypadku pojęcia wieloznacznego „duża prędkość“ inna wartość będzie uważana za dużą, jeżeli ograniczymy się do obszaru rozważań [0, 2 km/h], a inna jeżeli przyjmujemy przedział [0, 10000 km/h]. Obszar rozważań, nazywany przestrzenią lub zbiorem, będzie najczęściej oznaczony literą  $X$ . Należy pamiętać, iż  $X$  jest zbiorem nierozmytym. W tak przedstawionym zagadnieniu definicja zbioru rozmytego [1,2,3] została sformułowana następująco:

Zbiorem rozmytym  $A$  w pewnej niepustej przestrzeni  $X$ , co zapisano jako,  $A \subseteq X$  nazwano zbiór par:

$$A = \{(x, \mu_A(x)); x \in X\} \quad (1)$$

gdzie

$$\mu_A : X \rightarrow [0,1] \quad (2)$$

jest funkcją przynależności zbioru rozmytego  $A$ . Funkcja ta każdemu elementowi  $x \in X$  przypisuje jego stopień przynależności do zbioru rozmytego  $A$ , przy czym można wyróżnić trzy przypadki: pełną przynależność elementu  $x$  do zbioru rozmytego  $A$ , gdy  $\mu_A(x) = 1$ , brak przynależności elementu  $x$  do zbioru rozmytego  $A$ , gdy  $\mu_A(x) = 0$ , oraz częściową przynależność elementu  $x$  do zbioru rozmytego  $A$ , gdy  $0 < \mu_A(x) < 1$ . Podstawowe pojęcia logiki rozmytej wprowadził w 1964 roku Lotfi Zadeh. Pierwsze praktyczne zastosowanie tej

teorii należy do E. H. Mamdaniego, który w 1975 roku opisał i zbudował prosty układ sterowania.

**2 Metody tworzenia układów rozmytych**

Główną zaletą modeli rozmytych względem konwencjonalnych modeli matematycznych jest możliwość ich opracowania na bazie znacznie mniejszej ilości informacji o systemie. Informacja ta, jak wcześniej wspomniano, może mieć charakter nieprecyzyjny, rozmyty. Dotychczasowe prace nad tą dziedziną nauki wypracowały trzy zasadnicze metody modelowania rozmytego [4,5,6], czyli tworzenia modeli rozmytych systemów rzeczywistych.

Pierwszą metodą modelowania, która jako pierwsza powstała i została zastosowana w praktyce jest metoda modelowania rozmytego na bazie wiedzy eksperta systemu. Bazuje ona na wiedzy i doświadczeniu człowieka dobrze znającego modelowany system. Wiedza człowieka może mieć charakter jawny, świadomy lub podświadomy. Suma świadomej i podświadomej wiedzy o systemie rzeczywistym zawarta w umyśle eksperta określana jest jako model mentalny lub umysłowy. Zbiór słownie sformułowanych reguł określających zależności wejścia/wyjścia oraz zbiór słownych informacji o wartościach lingwistycznych stosowanych przez eksperta nazywano modelem werbalnym.

Szereg niedoskonałości metody wiedzy eksperta skłoniło do opracowania rozmytych modeli samonastrajających się. Rozumie się przez to model o stałej bazie reguł i ilości zbiorów rozmytych. Strojanie polega głównie na określeniu takich parametrów funkcji przynależności wejść i wyjść, które minimalizują błąd modelu względem modelowanego systemu w sensie przyjętego sposobu oceny tego błędu. Do strojenia parametrów modelu najczęściej stosuje się następujące metody:

- I. metody rozmytych sieci neuronowych
- II. metody poszukiwań
- III. metody oparte na klasteryzacji

Największe znaczenie mają pierwsze trzy metody. Metoda I polega na przekształceniu modelu rozmytego w rozmytą sieć neuronową i metod uczenia tych sieci na bazie danych pomiarowych wejścia/wyjścia systemu w celu nastrojenia parametrów modelu. Metoda II polega na poszukiwaniu optymalnych parametrów modelu rozmytego. Najczęściej stosowaną metodą zorganizowanego poszukiwania jest metoda algorytmów genetycznych. Metody oparte na klasteryzacji mogą łączyć strojenie parametrów modelu z organizacją jego struktury.

Samoorganizujące się modele to trzecia metoda tworzenia układów rozmytych. Są to modele, które same określają optymalną ilość i postać reguł oraz zbiorów rozmytych charakteryzujących każdą z jego zmiennych (wejścia, wyjścia). Również model ten sam określa optymalne parametry funkcji przynależności, współczynniki ufności reguł. Jednoczesne stosowanie samoorganizacji i strojenia daje zwykle wyższą dokładność. Jednak trudności strojenia parametrów rosną tak szybko, że po przekroczeniu pewnego poziomu komplikacji modelu strojenie staje się praktycznie niemożliwe. Dlatego najlepszym podejściem jest dążenie do uproszczenia struktur modeli.

**3 Modelowanie dynamiki układów mechanicznych z wykorzystaniem rozmytych układów.**

W celu zilustrowania możliwości układów rozmytych w poniższym artykule przedstawiono modelowanie dynamiki na przykła-

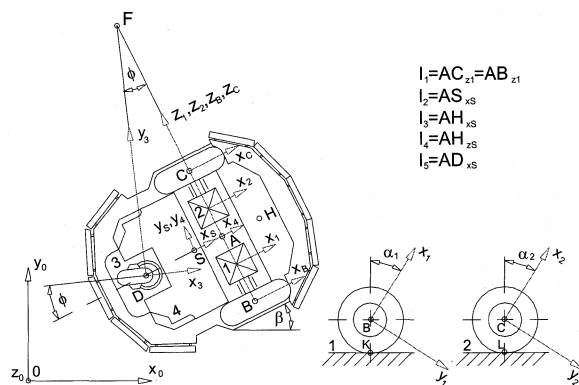
dzie mobilnego robota kołowego Pioneer 2DX. Ponieważ analizowany robot jest układem anholonomicznym silnie nieliniowym, po przeprowadzeniu wstępnych obliczeń symulacyjnych z wykorzystaniem różnego rodzaju algorytmów opartych na sztucznej inteligencji, zdecydowano się zastosować układy rozmyte.

### 3.1 Opis matematyczny układu i stanowiska badawczego.

Modelowanie dynamiki układów mechanicznych sprowadza się do podania dynamicznych równań ruchu oraz ich identyfikacji modelu matematycznego układu mechanicznego. Poprawnie zidentyfikowany i zweryfikowany model matematyczny w eksperymencie może posłużyć do stworzenia odpowiedniego układu sterowania. W dalszej części tego punktu przedstawiono przykładowy sposób tworzenia opisu matematycznego na przykładzie mobilnego robota Pioneer 2DX.

#### 3.1.1 Model matematyczny mobilnego robota Pioneer 2DX

Do obliczeń związanych w opisie dynamiki tego układu przyjęto model robota przedstawiony na rys.1. Podstawowymi elementami tego modelu są: rama 4 pojazdu, zespół napędzający 1 i 2 oraz samonastawne koło podpierające 3.



Rys.1 Model obliczeniowy robota Pioneer 2DX

Wykorzystując program do obliczeń symbolicznych MapleTM zapisano dynamiczne równania ruchu robota nie uwzględniając samonastawnego koła podpierającego 3 w postaci:

$$\begin{aligned} a_2 \ddot{\alpha}_2 + a_1 \dot{\alpha}_1 + a_3 \dot{\alpha}_2^2 - a_3 \dot{\alpha}_1 \dot{\alpha}_2 &= M_1 - a_4 \operatorname{sgn}(\dot{\alpha}_1) \\ a_2 \dot{\alpha}_1 + a_1 \ddot{\alpha}_2 + a_3 \dot{\alpha}_1^2 - a_3 \dot{\alpha}_1 \dot{\alpha}_2 &= M_2 - a_5 \operatorname{sgn}(\dot{\alpha}_2) \end{aligned} \quad (3)$$

gdzie

$$\begin{aligned} a_1 &= \frac{1}{4} \frac{r^2 m_4 l_1^2 + r^2 m_4 l_2^2 + r^2 I_{z_4} + 2 r^2 I_{x_1} + 4 r^2 m_1 l_1^2 + 4 I_{z_1} l_1^2}{l_1^2} \\ a_2 &= \frac{1}{4} \frac{-r^2 m_4 l_2^2 + r^2 m_4 l_1^2 - r^2 I_{z_4} - 2 r^2 I_{x_1}}{l_1^2} \\ a_3 &= \frac{1}{4} \frac{r^3 m_4 l_2}{l_1^2} \quad a_4 = N_1 f_1 \quad a_5 = N_2 f_2 \end{aligned}$$

natomiast:

$m_2 = m_1, m_4$  - masa poszczególnych elementów.  
 $I_{x_2} = I_{x_1}, I_{z_2} = I_{z_1}, I_{z_4}$  - momenty bezwładności względem odpowiednich osi współrzędnych (rys.1).  
 $M_1, M_2$  - momenty napędowe.  
 $N_1, N_2$  - siły nacisku na poszczególne koła.  
 $f_1, f_2$  - współczynnik tarcia toczonego dla poszczególnych kół.  
 $a_1, a_2$  - kąty obrotu własnego kół 1 i 2  
 Analizowany robot posiada 2 stopnie swobody, tak więc biorąc pod uwagę zmienne stanu w postaci:

$$\alpha_1 = x_1, \dot{\alpha}_1 = \dot{x}_1 = x_2, \alpha_2 = x_3, \dot{\alpha}_2 = \dot{x}_3 = x_4 \quad (4)$$

Dynamiczne równania ruchu (1) zostały zapisane jako:

$$\dot{x} = Ax + B[f(x, a) + G(x, a)u] \quad (5)$$

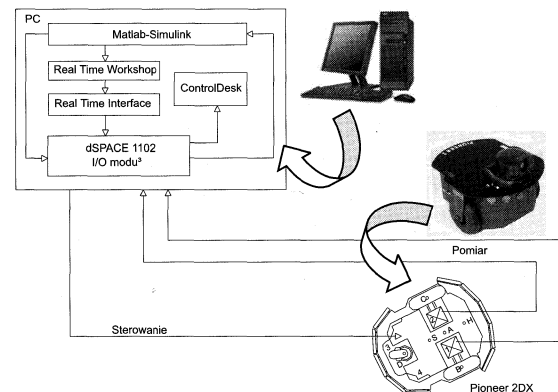
gdzie

$$\begin{aligned} A &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & B &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ f(x, a) &= \begin{bmatrix} 0 \\ 0 \\ \frac{a_2 a_3 \operatorname{sgn}(x_2) + a_2 a_3 x_2^2 - a_2 a_3 x_2 x_4 + a_1 a_3 x_2 x_1 - a_1 a_3 x_4^2 - a_1 a_4 \operatorname{sgn}(x_2)}{a_1^2 - a_2^2} \\ \frac{-a_1 a_3 \operatorname{sgn}(x_2) - a_1 a_3 x_2^2 + a_1 a_3 x_2 x_4 - a_2 a_3 x_2 x_4 + a_2 a_3 x_4^2 - a_2 a_4 \operatorname{sgn}(x_2)}{a_1^2 - a_2^2} \end{bmatrix} \\ G(x, a) &= \begin{bmatrix} 0 & 0 \\ \frac{a_1}{a_1^2 - a_2^2} & -\frac{a_2}{a_1^2 - a_2^2} \\ \frac{a_2}{a_1^2 - a_2^2} & \frac{a_1}{a_1^2 - a_2^2} \end{bmatrix} \end{aligned} \quad (6)$$

Wektor  $u$  w równaniu (5) reprezentuje momenty napędowe  $M_1$  i  $M_2$ , które mogą być pomierzone na rzeczywistym obiekcie lub wygenerowane w oparciu o dynamiczne równania ruchu (3).

#### 3.1.2 Opis stanowiska badawczego

W eksperymentach z wykorzystaniem obiektu rzeczywistego wykorzystano stanowisko badawcze do szybkiego prototypowania mechatronicznego z procesorem sygnałowym TMS320C31, umieszczonym na karcie DS1102 firmy dSPACE oraz bibliotekami oprogramowania karty. Schemat stanowiska przedstawiono na rys.2. Wspomniana karta umożliwia generowanie sygnałów sterujących oraz rejestrację pomiarów.



Rys.2 Schemat stanowiska badawczego

Komputer z programem MatlabTM, który jest skonfigurowany z kartą dSPACE umożliwia tworzenie modeli w środowisku MatlabTM-Simulink, a następnie dzięki oprogramowaniu (ControlDesk) można zarządzać eksperymentami oraz dokonywać rejestracji danych.

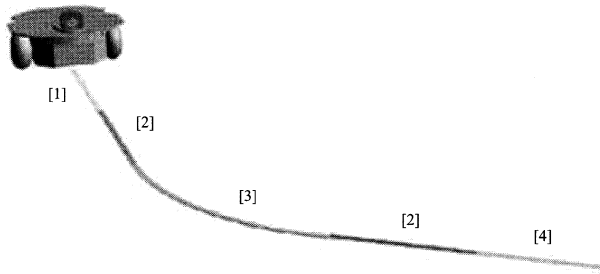
#### 3.1.2.1 Pomiary wykonane na stanowisku badawczym

W celu wykonania pomiaru parametrów kinematycznych i dynamicznych mobilnego robota Pioneer 2DX przyjęto następujący model ruchu, który przedstawiono na rys.3: start [1], jazda po prostej [2], jazda po łuku o kącie skreślenia ramy  $\beta$  i promieniu kół  $r$  [3], jazda po prostej [2], hamowanie [4]. Robot porusza się ze stałą, założoną prędkością punktu A (rys.1)  $v_A$ .

Podstawowe dane techniczne związane z robotem Pioneer 2DX zostały przedstawione w tabeli 1.

Przebiegi czasowe momentów napędowych dla założonego modelu ruchu i danych konstrukcyjnych z tabeli.1 zostały zilustrowa-

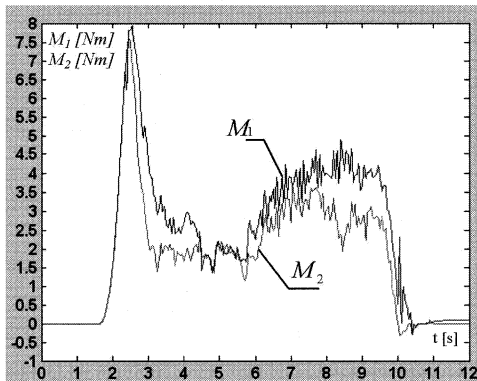
ne na rys. 4. Na rys. 5 przedstawiono parametry kinematyczne robota reprezentujące przebiegi czasowe kątów obrotu własnego kół 1 i 2. Natomiast na rys. 6 przedstawiono przebiegi czasowe prędko-



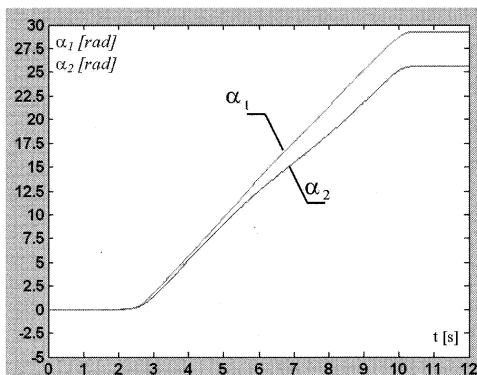
Rys.3 Model ruchu robota

$v_A$ [m/s]	$I_{z1}$ [kgm <sup>2</sup> ]	$m_1$ [kg]	$l_1$ [m]	$N_1$ [N]	$f_1$ [m]	$r$ [m]
0.4	0.051	1.5	0.163	31.25	0.01	0.082
						5
$I_{x1}$ [kgm <sup>2</sup> ]	$I_{z4}$ [kgm <sup>2</sup> ]	$m_4$ [kg]	$l_2$ [m]	$N_2$ [N]	$f_2$ [m]	$\beta$
0.02	0.154	5.67	0.07	31.25	0.01	0-75°

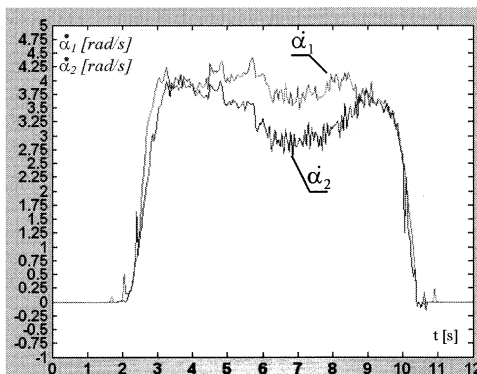
Tabela.1 Dane techniczne mobilnego robota Pioneer 2DX



Rys.4 Przebiegi czasowe momentów napędowych kół 1 i 2.



Rys.5 Przebiegi czasowe kątów obrotu własnego kół 1 i 2.



Rys.6 Przebiegi czasowe prędkości kątowych poszczególnych kół.

ści kątowych kół 1 i 2 zarejestrowane na stanowisku badawczym opisanym w punkcie 3.1.2.

Pozyskane przebiegi czasowe parametrów kinematycznych i dynamicznych zostaną wykorzystane w dalszej części do obliczeń związanych z modelowaniem dynamiki przykładowego modelu mechanicznego.

### 3.2 Zastosowanie układów rozmytych w modelowaniu dynamiki

W celu przeprowadzenia poprawnej analizy dynamiki mobilnych robotów kołowych należy dokonać identyfikacji dynamicznych równań ruchu. Przez zadanie identyfikacji rozumie się wybór najlepszego modelu w klasie modeli sformułowanych na podstawie matematycznego opisu zjawisk fizycznych [1,7]. Ponieważ zależność  $f(x,a)$  nie ma formy liniowej ze względu na parametry  $a$ , w zależnościach tych występują niedokładności modelowania, wektor stanu i momenty napędzające są dostępne pomiarowo, do analizy przyjęto opis emulatora stanu mobilnego robota w postaci:

$$\dot{\hat{x}} = A\hat{x} + B[\hat{f}(\hat{x}, \hat{a}) + G(a)u] + K\tilde{x} \quad (7)$$

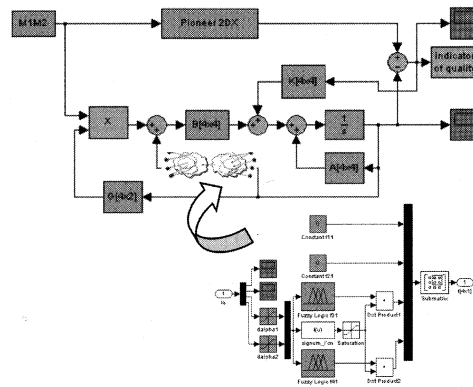
gdzie wektor  $\hat{x}$  jest estymatą wektora stanu  $x$ ,  $\hat{f}(\hat{x}, \hat{a})$  to estymata nieliniowej funkcji występującej w równaniach (5). Przyjmując błąd estymacji wektora stanu w postaci  $\tilde{x} = x - \hat{x}$  oraz odejmując równanie (7) od równania (5) otrzymano opis struktury równoległej emulatora stanu w przestrzeni błędów:

$$\dot{\tilde{x}} = A_H \tilde{x} + B[\tilde{f}(x, \hat{x}, a, \hat{a}) + G(a)u]$$

gdzie:

$A_H = A - K$ , a macierz  $K$  jest tak dobrana aby równanie charakterystyczne macierzy  $A_H$  było ściśle stabilne.

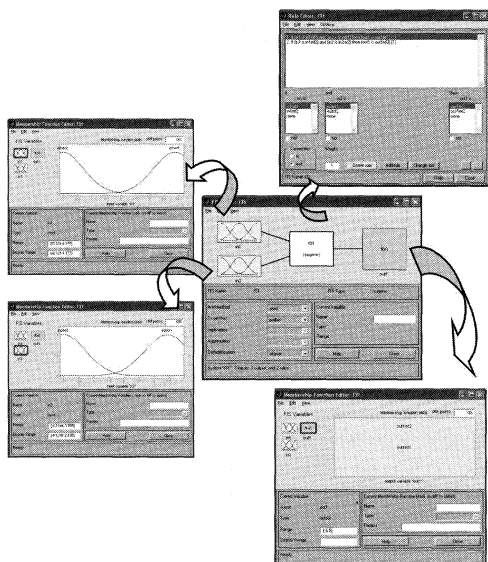
Do wyznaczenia funkcji  $\tilde{f}(x, \hat{x}, a, \hat{a})$  zastosowano układ z logiką rozmytą w strukturze równoległej emulatora stanu robota wykonany w programie Matlab™-Simulink. Na rys.7 przedstawiono model dynamiki robota wykonany w tym programie.



Rys.7 Struktura równoległa emulatora stanu robota Pioneer 2DX

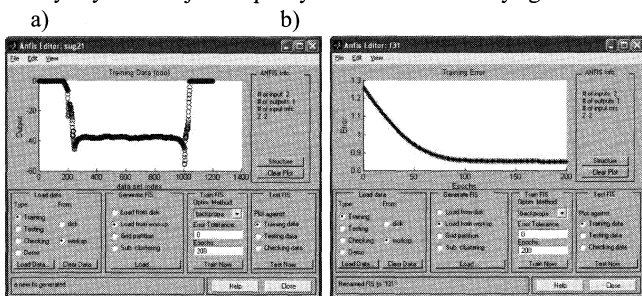
Wygenerowano dwa identyczne pod względem struktury układy rozmyte, zdecydowano się na modele Takagi-Sugeno. Schemat jednego z układów rozmytych przy użyciu edytora dodatkowego zestawu opcji programu Matlab™ (fuzzy logic toolbox) zilustrowano na rys.8.

Przedstawiony sposób modelowania dynamiki i identyfikacji dynamicznych równań ruchu może posłużyć do dalszych obliczeń z wykorzystaniem logiki rozmytej. Sposób prowadzenia obliczeń z wykorzystaniem samonastrajających się układów rozmytych został bardzo dokładnie opisany przez autora pracy [7]. Taki sposób modelowania umożliwił pozyskanie informacji na temat danych wejściowych i wyjściowych operujących na zastosowanych układach rozmytych w oparciu o rzeczywiste dane pomiarowe. Jednak sposób ten umożliwił prowadzenie identyfikacji dynamicznych równań ruchu off-line. Tak więc zdecydowano się wykorzystać samonastrajające się adaptacyjno neuronowe-rozmyte układy wnioskowania.



Rys.8 Struktura układu rozmytego

Program Matlab™ posiada wiele interesujących funkcji związanych z tworzeniem układów rozmytych. Jedną z nich jest funkcja *genfis2* [8], która na podstawie danych wejściowych i wyjściowych potrafi zbudować układ rozmyty. Dla analizowanego zagadnienia funkcja ta wygenerowała dwa identyczne pod względem struktury układy rozmyte. Strukturę jednego z układów przedstawiono na rys.8. Układ posiada dwa wejścia jedno wyjście, każde wejście posiada dwie funkcje przynależności, natomiast baza reguł ma 2 reguły. Danymi wejściowymi podanymi na układ rozmyty są prędkości kątowe kół. Posiadając wygenerowane układy rozmyte można przystąpić do wyuczenia ich za pomocą funkcji *anfis* [8]. Podejście do modelowania wykorzystywane przez funkcję *anfis* jest podobne do wielu technik identyfikacji układów. Najpierw stawiana jest hipoteza na temat struktury modelu sparametryzowanego (przypisując: wejścia do funkcji przynależności, funkcje przynależności do reguł, reguły do wyjść). Następnie zbierane są dane wejścia/wyjścia w formie która będzie możliwa do wykorzystania przez *anfis* w procesie nauczania modeli z układami wnioskowania (FIS) do naśladowania danych uczących przedstawionych poprzez modyfikację parametrów funkcji przynależności według wybranego kryterium błędu. Wykorzystując edytor funkcji *anfis* wczytano dane uczące oraz wygenerowany przy pomocy funkcji *genfis2* układ rozmyty, na rys.9a przedstawiono edytor z wczytanymi danymi. Następnie w procesie uczenia wykorzystano sieć neuronową, która wykorzystuje algorytm wstecznej propagacji błędów. W celu przeprowadzenia obliczeń związanych z adaptacją układu rozmytego przyjęto tolerancję błędów równą 0, natomiast liczbę epok równą 200. Na rys.9b został przedstawiony edytor funkcji *anfis* po wyuczeniu układu rozmytego.

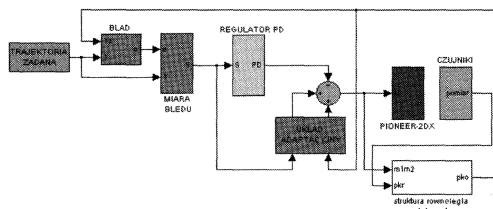
Rys.9 Edytor funkcji *anfis*

Taką samą operację wykonano w przypadku drugiego układu rozmytego (rys.7). Tak wyuczone dwa modele rozmyte zweryfikowano w eksperymencie polegającym na podłączeniu analizowanej struktury do układu sterowania adaptacyjnego przedstawionego na rys.10.

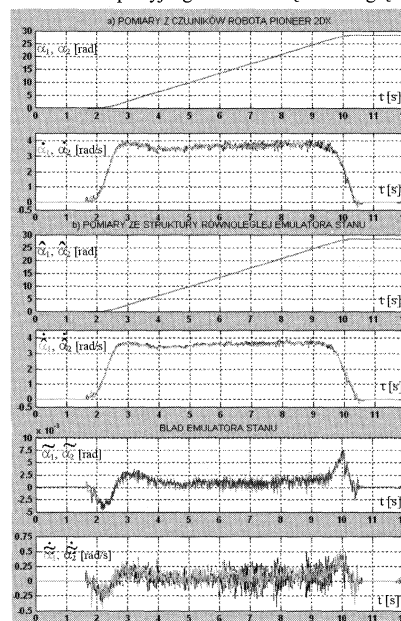
Dokonano wielu eksperymentów zmieniając trajektorię i prędkość robota w zakresie danych, które zostały wykorzystane do na-

uczania. Przykładowe przebiegi otrzymane z eksperymentu w czasie rzeczywistym z wykorzystaniem układu sterowania (rys.10) dla zmienionego modelu ruchu (modelu ruchu po prostej) przyjętego w celu weryfikacji modeli rozmytych, przedstawiono na rys.11.

Z przeprowadzonych badań wynika, iż zastosowane układy rozmyte sprawdzone on-line na rzeczywistym obiekcie zadowalająco spełniły swoje zadanie. Największe błędy układy rozmyte wygenerowały podczas startu i hamowania, co wydaje się być naturalnym zjawiskiem i potwierdzeniem badań eksploatacyjnych.



Rys.10 Układ sterowania adaptacyjnego ze strukturą równoległą emulatora stanu



Rys.11 Przebiegi czasowe z eksperymentu

## 4 Podsumowanie

Przedstawiony sposób modelowania wykorzystujący samoorganizujące się układy rozmyte może być bardzo korzystny do zastosowania, gdy dysponuje się znikomą wiedzą o obiekcie, a jedynie posiada się dane wejściowe i wyjściowe układu. W podobny sposób można wykorzystywać logikę rozmytą do modelowania dynamiki innych układów mechanicznych.

## Literatura

- [1] GIERGIEL J., HENDZEL Z., ŻYLSKI W.: Modelowanie i sterowanie mobilnych robotów kołowych w ujęciu mechatronicznym, PWN, Warszawa, 2002.
- [2] RUTKOWSKA D., PILIŃSKI M., RUTKOWSKI L.: Sieci neuronowe, algorytmy genetyczne i systemy rozmyte, PWN, Warszawa 1999.
- [3] DRIANKOV D., HELLENDORF H., REINFRANK M.: Wprowadzenie do sterowania rozmytego, WNT, Warszawa 1996.
- [4] PIEGAT A.: Modelowanie i sterowanie rozmyte, ELIT, Warszawa 1999
- [5] BUTKIEWICZ B.S.: Układy rozmyte, część 1, Elektronika Praktyczna, Zeszyt 3, 2000.
- [6] BUTKIEWICZ B.S.: Układy rozmyte, część II, Elektronika Praktyczna, Zeszyt 4, 2000.
- [7] BURATOWSKI T., UHL T., ŻYLSKI W.: The comparison of parallel and serial-parallel structures of mobile robot Pioneer 2DX state emulator, Proceedings of the Seventh Symposium on Robot Control SYROCO'03, Wrocław, 2003.
- [8] The MathWorks, Inc., Fuzzy Logic Toolbox.

**Title:** Fuzzy logic application in the modelling of the mechanical systems dynamics

Artykuł recenzowany