

METODY SZTUCZNEJ INTELIGENCJI

Laboratorium nr 12_4F

Temat: Fuzzy Logic Toolbox - Pakiet z interfejsem graficznym do budowy modeli rozmytych w Matlab'ie, przykład modelowania rozmytego

Fuzzy Logic Toolbox - Pakiet z interfejsem graficznym do budowy modeli rozmytych w Matlab'ie, przykład modelowania rozmytego

1. Cel ćwiczenia

Celem ćwiczenia jest zapoznanie się z możliwościami aproksymacji funkcji nieliniowych dwóch zmiennych przy pomocy układów z logiką rozmytą, z zastosowaniem przybornika pakietu Matlab - Fuzzy Logic Toolbox.

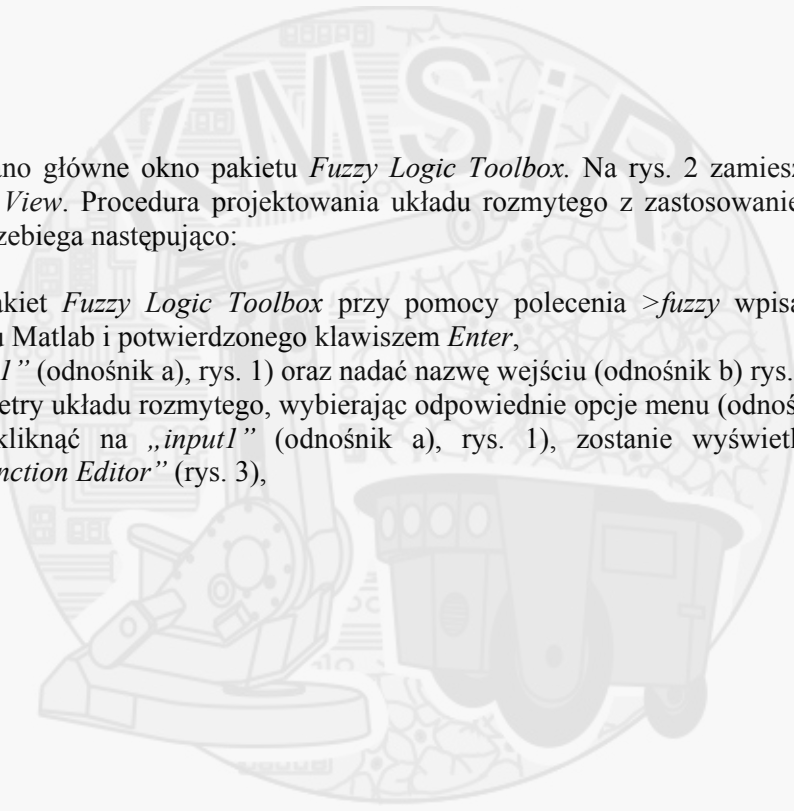
2. Projektowanie układów rozmytych - Fuzzy Logic Toolbox

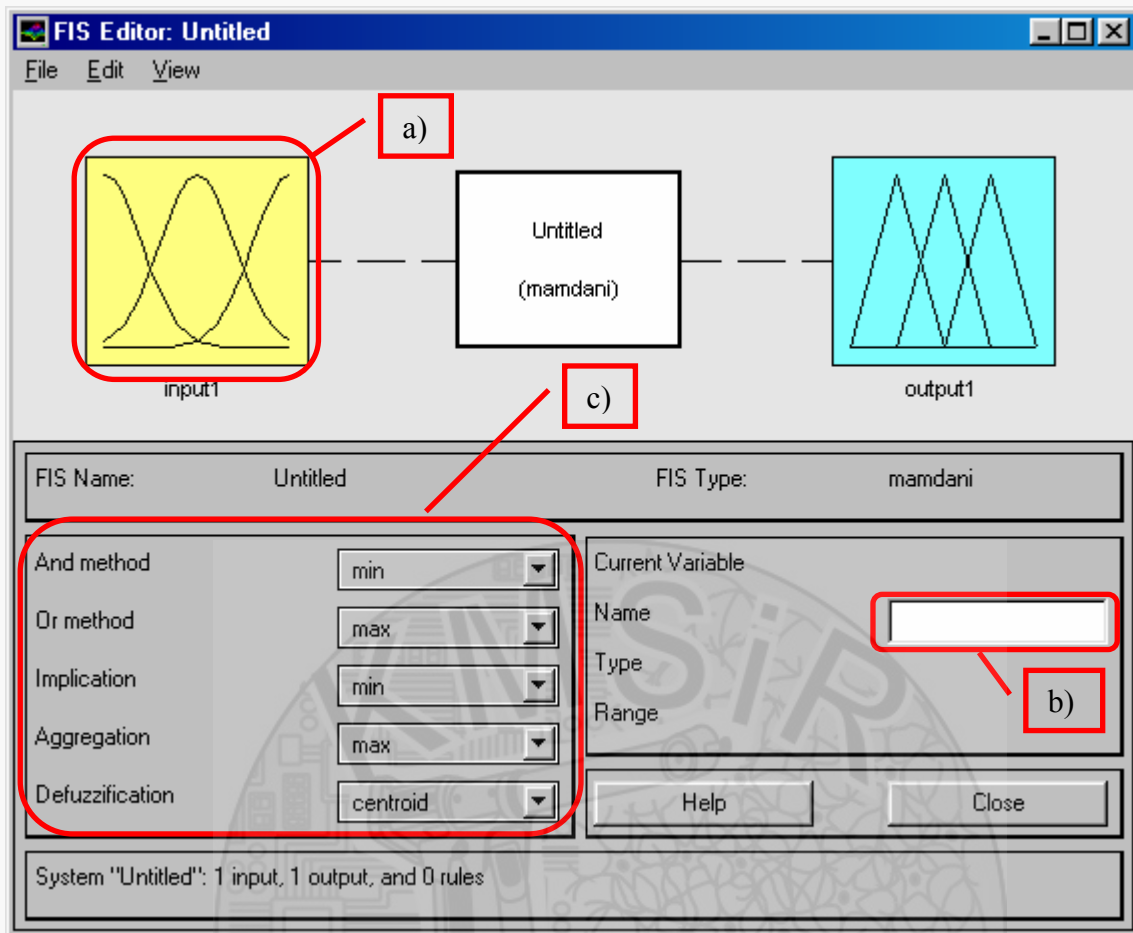
Fuzzy Logic Toolbox – to pakiet z interfejsem graficznym do budowy modeli rozmytych w Matlab-ie. Pakiet *Fuzzy Logic Toolbox* uruchamiamy z poziomu linii poleceń programu Matlab przy pomocy komendy:

`>fuzzy ↵` (enter)

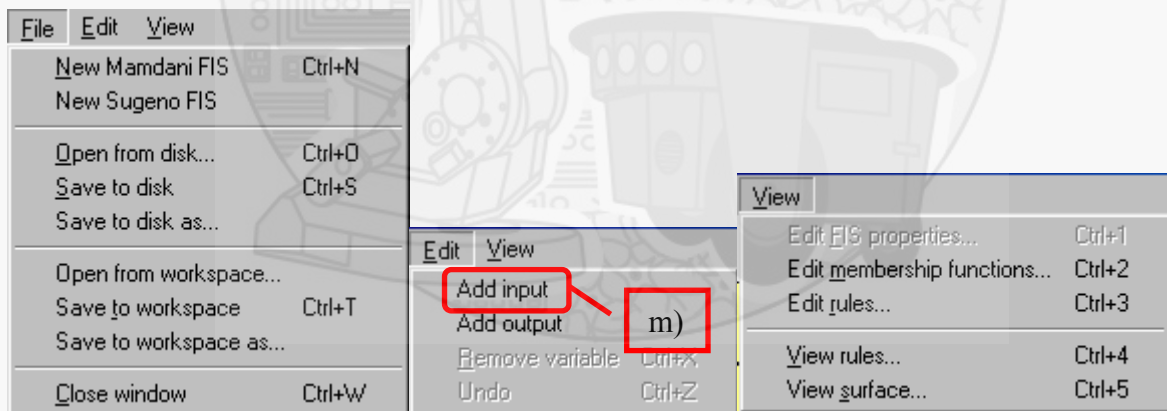
Na rys. 1 pokazano główne okno pakietu *Fuzzy Logic Toolbox*. Na rys. 2 zamieszczono rozwijane menu *File, Edit, View*. Procedura projektowania układu rozmytego z zastosowaniem pakietu *Fuzzy Logic Toolbox* przebiega następująco:

1. Uruchomić pakiet *Fuzzy Logic Toolbox* przy pomocy polecenia `>fuzzy` wpisanego w wierszu poleceń programu Matlab i potwierdzonego klawiszem *Enter*,
2. Kliknąć „*input1*” (odnośnik a), rys. 1) oraz nadać nazwę wejściu (odnośnik b) rys. 1),
3. Ustawić parametry układu rozmytego, wybierając odpowiednie opcje menu (odnośnik c) rys. 1),
4. Dwukrotnie kliknąć na „*input1*” (odnośnik a), rys. 1), zostanie wyświetlone nowe okno „*Membership Function Editor*” (rys. 3),



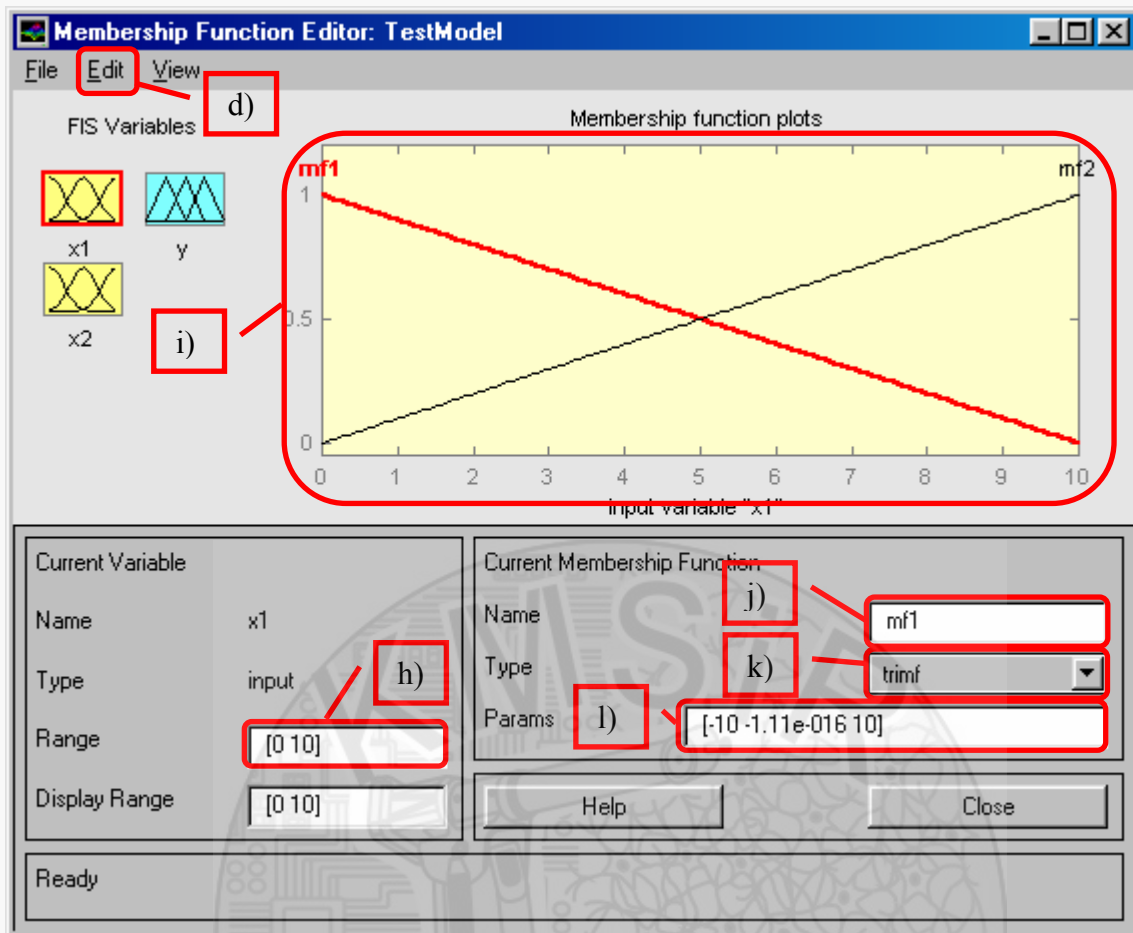


Rys. 1. Główne okno pakietu *Fuzzy Logic Toolbox*

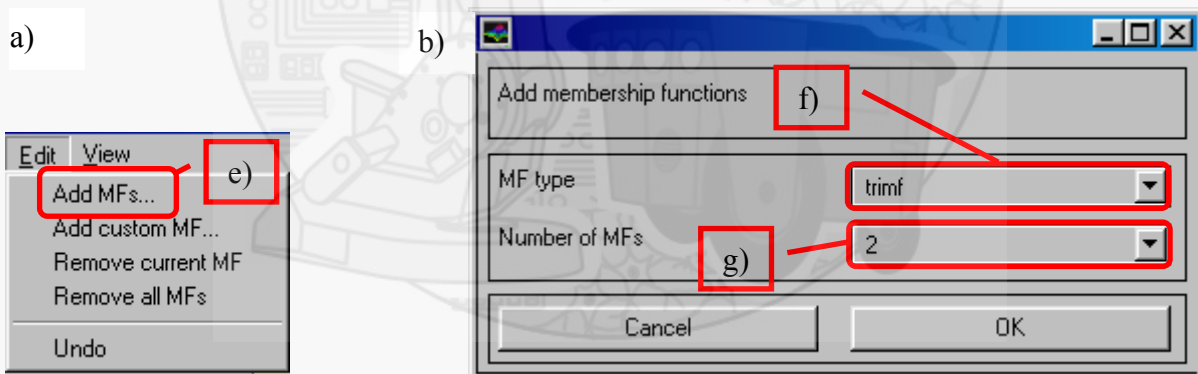


Rys. 2. Rozwijane menu pakietu *Fuzzy Logic Toolbox*, *File*, *Edit*, *View*

5. W oknie edytora „*Membershi Function Editor*” wybrać menu *Edit* (odnośnik d) rys. 3),
6. Z menu *Edit* (rys. 4.a)) wybrać *Add MFs...* (*Membership Functions*) (odnośnik e) rys. 4.a)),
7. W oknie wyboru funkcji przynależności (rys. 4.b)) określić liczbę (odnośnik g) rys. 4.b)) oraz kształt funkcji przynależności (odnośnik f) rys. 4.b)), zaakceptować wybór klikając przycisk „*OK*”,
8. Wprowadzić przewidywany przedział wartości wejścia 1, $x_1 \subseteq X_1$ (odnośnik h) rys. 3), dla każdej z utworzonych funkcji przynależności, której parametry można edytować po kliknięciu na jej wykres (odnośnik i) rys. 3), podać nazwę funkcji przynależności (odnośnik j) rys. 3, np. „*S*”, „*M*”, „*L*”), ustawić odpowiedni typ funkcji przynależności (odnośnik k) rys. 3), podać parametry funkcji przynależności (odnośnik l) rys. 3) - położenie charakterystycznych punktów wykresu funkcji przynależności w przestrzeni wejścia,

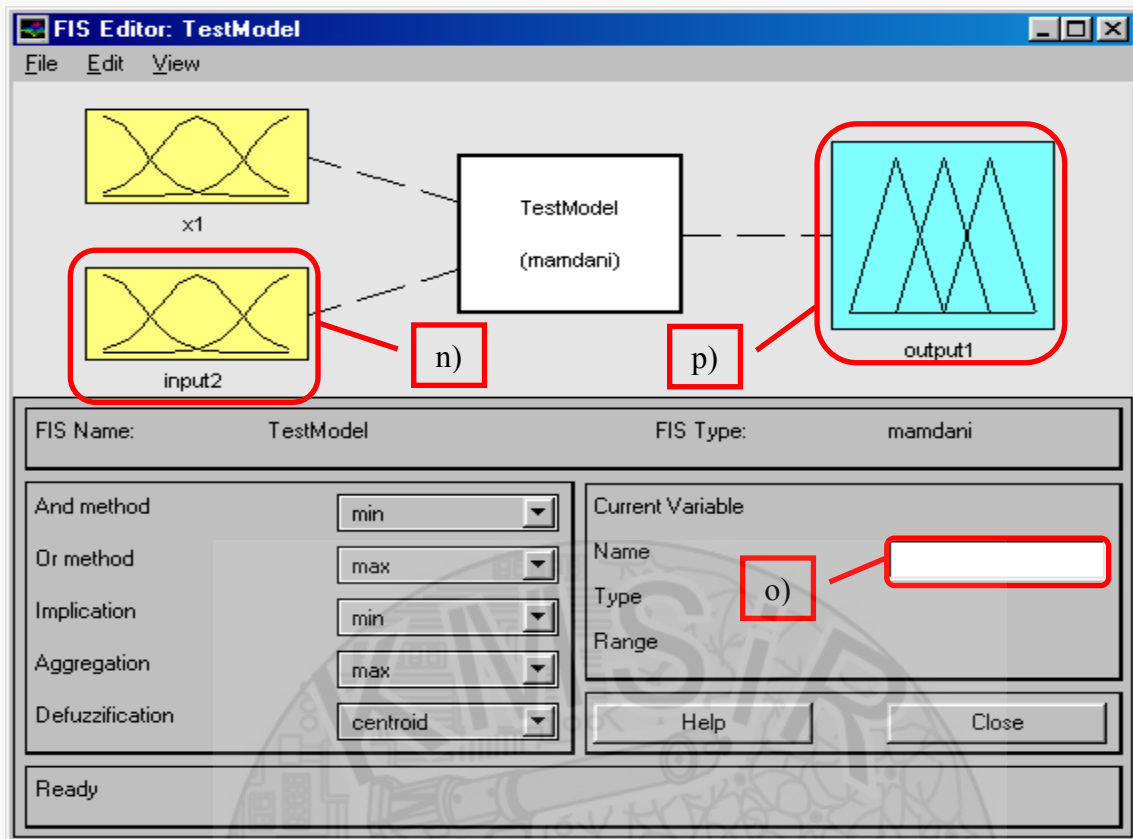


Rys. 3. Okno edytora „Membership Function Editor”

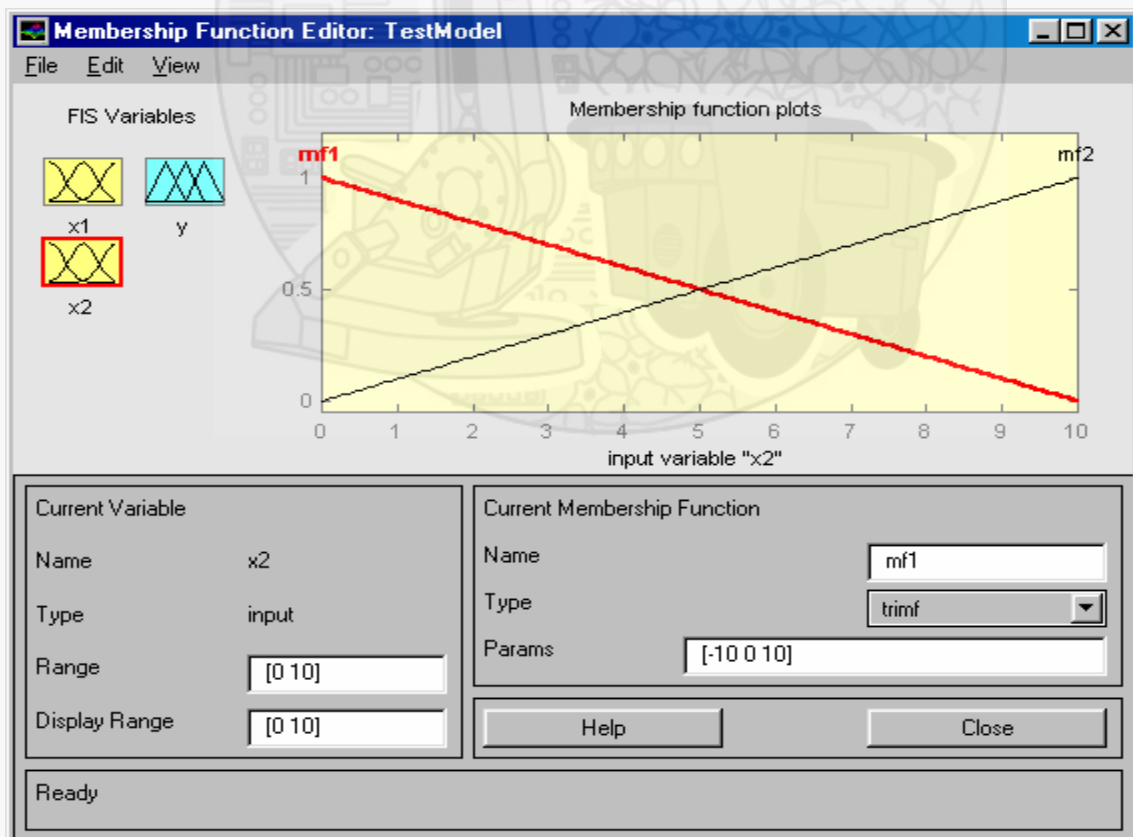


Rys. 4.a) Menu *Edit*, b) okno wyboru funkcji przynależności (MFs - Membership Functions)

9. Wrócić do głównego okna pakietu *Fuzzy Logic Toolbox* (rys. 1), z menu *Edit* wybrać *Add input* (odnośnik m) rys. 2), dodać funkcje przynależności dla drugiego wejścia x_2 (*input2*),
10. Nadać nazwę utworzonemu zbiorowi funkcji przynależności dla drugiego wejścia (*input2*) (odnośnik n) rys. 5), wpisując w polu *Name*: „ x_2 ” (odnośnik o) rys. 5),
11. Powtórzyć procedurę wyboru funkcji przynależności dla drugiego wejścia x_2 wg. punktów 2-8 instrukcji (rys. 6),



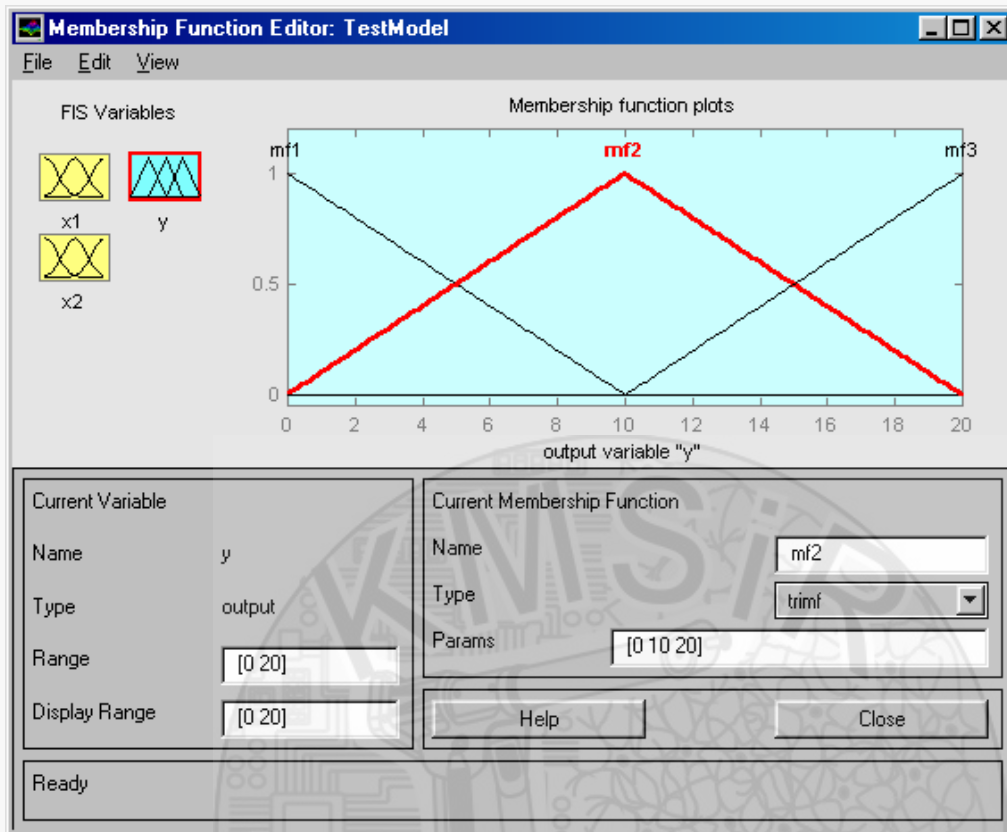
Rys. 5. Dodanie funkcji przynależności dla drugiego wejścia x_2



Rys. 6. Ustawianie parametrów funkcji przynależności zbiorów rozmytych wejścia x_2

12. Nadać nazwę zmiennej wyjściowej (*output*) (odnośnik p) rys. 5), wpisując w polu *Name*: „y” (odnośnik o) rys. 5),

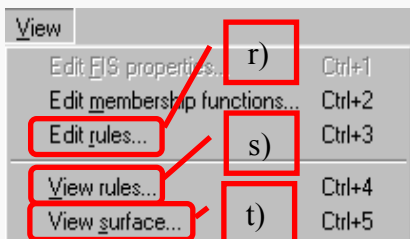
13. Powtórzyć procedurę wyboru funkcji przynależności dla wyjścia y wg. punktów 2-8 instrukcji (rys. 7),



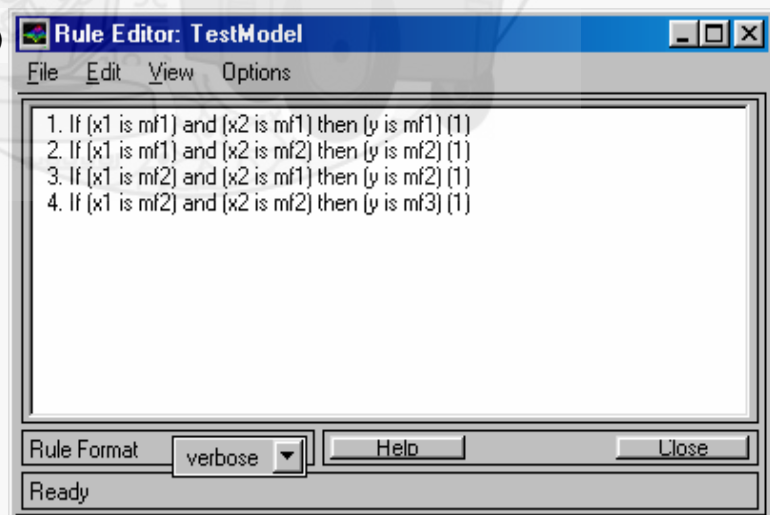
Rys. 7. Ustawianie parametrów funkcji przynależności zbiorów rozmytych wyjścia y

14. Uruchomić edytor bazy reguł (*Rule Editor*) (odnośnik r) rys. 8.a)), następnie wprowadzić bazę reguł zgodnie z zależnością (7), rys. 8.b). Interpretację graficzną wprowadzonej bazy reguł (rys. 9) można podglądać wybierając opcję *View Rules...* (odnośnik s) rys. 8.a)),

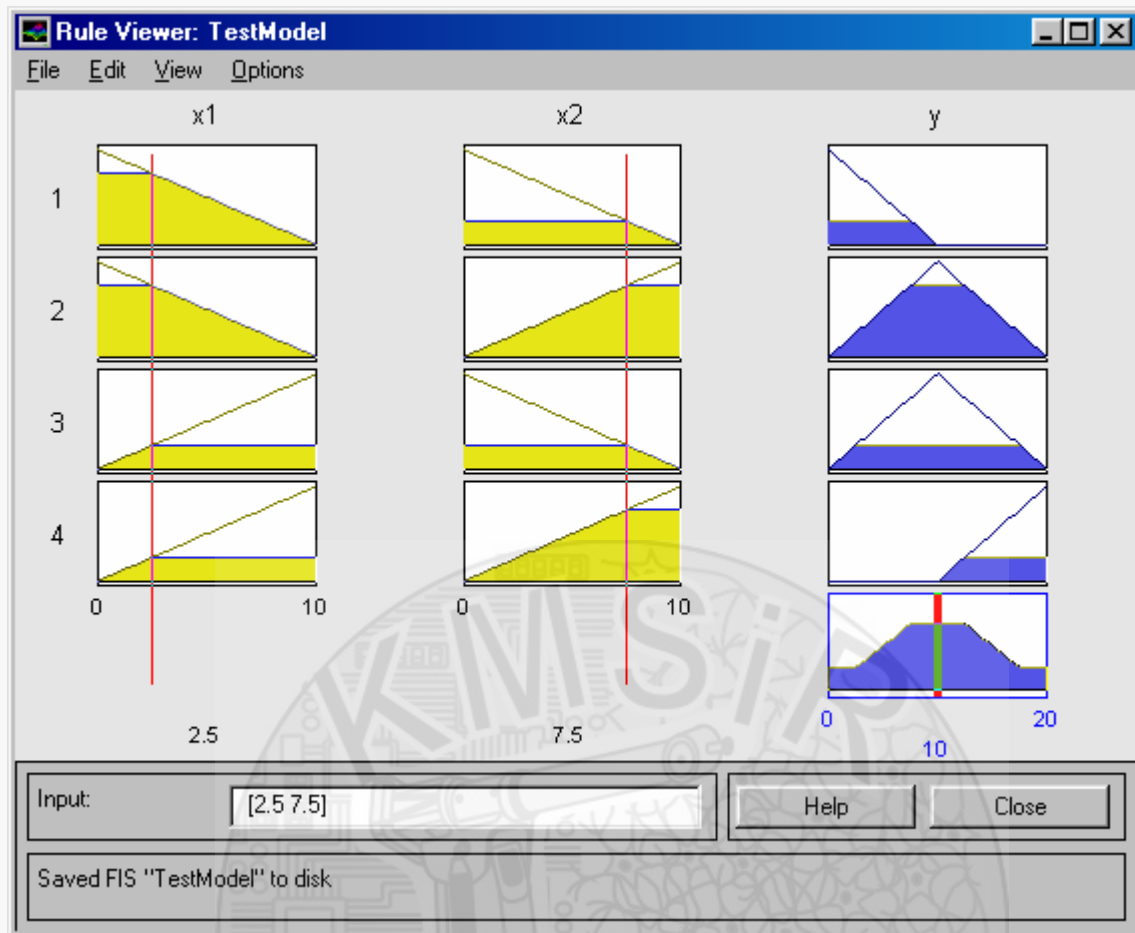
a)



b)

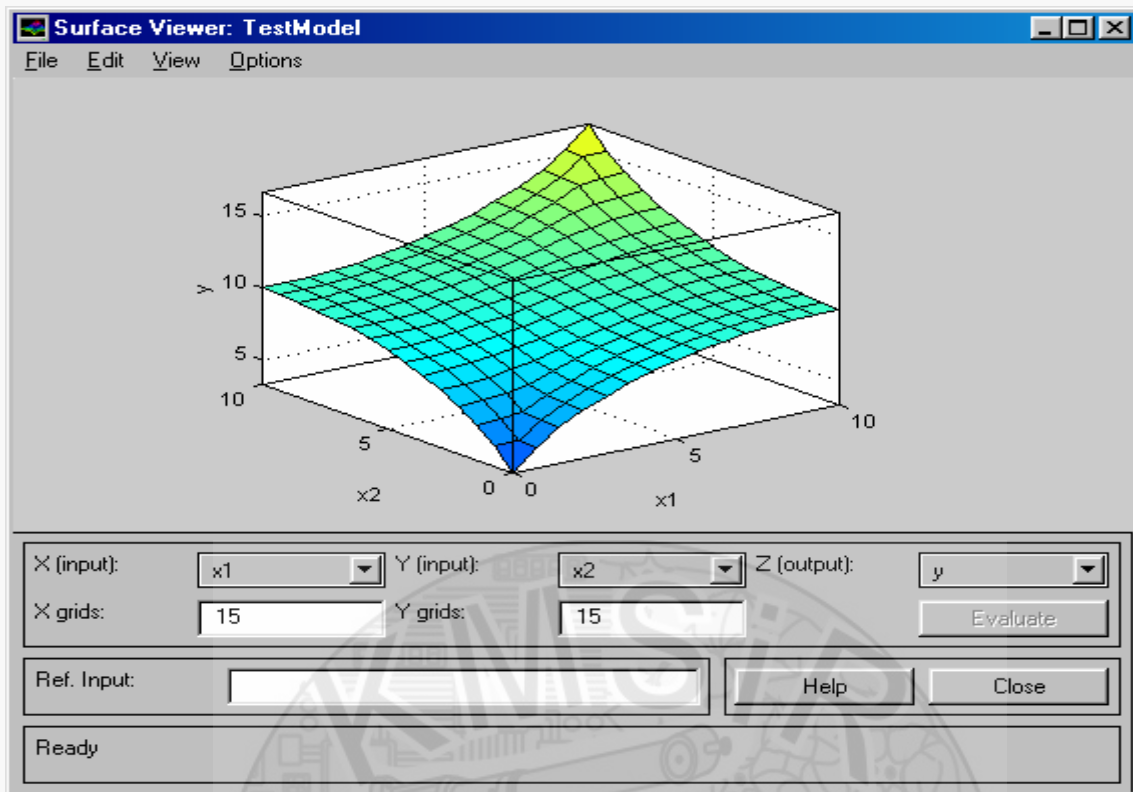


Rys. 8.a) Uruchamianie edytora bazy reguł, b) edycja bazy reguł



Rys. 9. Okno *Rule Viewer* umożliwiające graficzny podgląd wprowadzonej bazy reguł

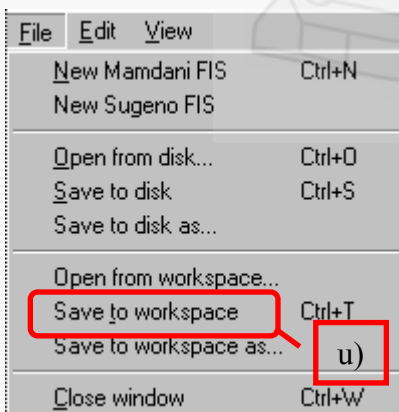
15. Wybierając z menu *View* opcję *View Surface...* (odnośnik t) rys. 8) wyświetlone zostanie okno *Surface Viewer* (rys. 10), umożliwiające podgląd wartości wyjścia modelu rozmytego y w funkcji wejść x_1 , x_2 w postaci wykresu przestrzennego,



Rys. 10. Okno *Surface Viewer* zawierające graficzną interpretację powierzchni modelu rozmytego $y(x_1, x_2)$

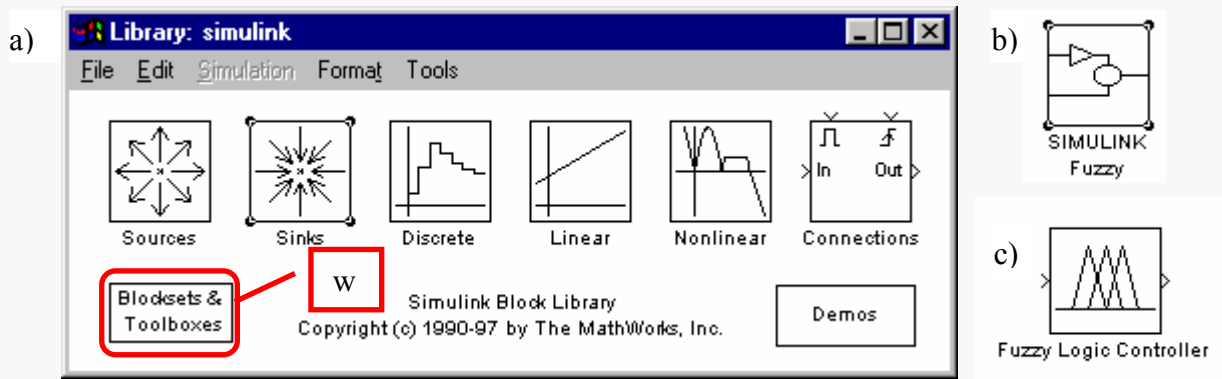
Zastosowanie modelu rozmytego lub rozmytego sterownika zbudowanego w pakiecie *Fuzzy Logic Toolbox* wymaga przeprowadzenia następującej procedury:

16. W pakiecie *Fuzzy Logic Toolbox* należy z menu *File* (rys. 11) wybrać opcję *Save to Workspace* (odnośnik u) rys. 11) i podać nazwę dla utworzonego wcześniej modelu rozmytego, np. *TestModel*. Model zostanie zapisany do przestrzeni roboczej (*Workspace*) Matlab-a, co umożliwi jego zastosowanie w pakiecie Simulink (model rozmyty stworzony w pakiecie *Fuzzy Logic Controller* można zachować w pamięci komputera w postaci pliku z rozszerzeniem *.fis* wybierając z menu *File* opcję *Save to disk* (rys. 11)),



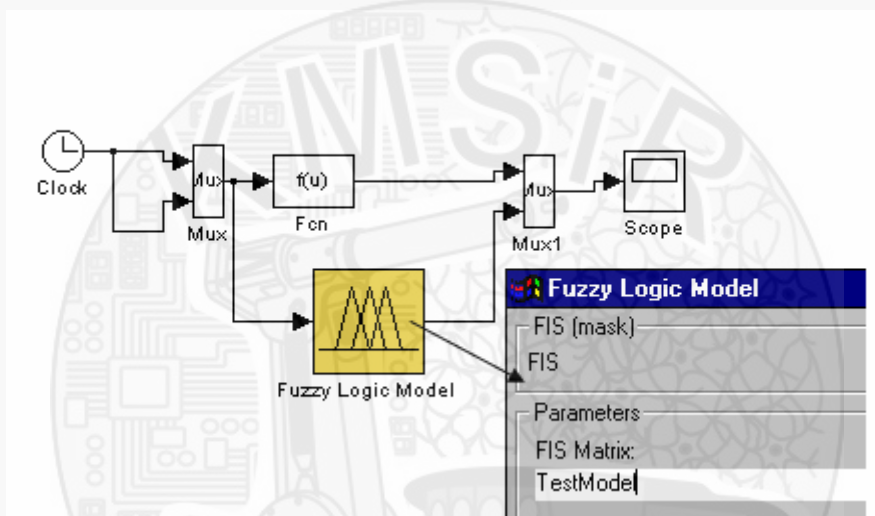
Rys. 11. Menu *File* pakietu *Fuzzy Logic Toolbox*

17. Z biblioteki pakietu Simulink (rys. 12.a)) należy wybrać *Blocksets & Toolboxes* (odnośnik w) rys. 12) -> *SIMULINK Fuzzy* (rys. 12.b)), następnie blok *Fuzzy Logic Controller* (rys. 12.c)) należy przenieść do edytowanego modelu w Simulink-u (rys. 13),



Rys. 12. Wybór bloczka *Fuzzy Logic Controller* z bibliotek Simulink-a

18. Dwukrotnie kliknąć na bloczek *Fuzzy Logic Controller* i wpisać nazwę modelu rozmytego nadaną modelowi stworzonemu w *Fuzzy Logic Toolbox* w pkt. 16.



Rys. 13. Model Simulink-a z zastosowaniem modelu rozmytego stworzonego w pakiecie *Fuzzy Logic Toolbox*

Przykład 1

Zaprojektujmy model rozmyty aproksymujący ciągłą funkcję $g(x)=\sin(x)$, zdefiniowaną na zbiorze $U=[0,3]$ z dokładnością $\varepsilon = 0.5$, to znaczy $\sup_{x \in U} (|g(x) - f(x)|) < \varepsilon$.

Ponieważ $\left\| \frac{\partial g}{\partial x} \right\| = \|\cos(x)\| = 1$, to na podstawie zależności

$$\|g(x_1, x_2) - f(x_1, x_2)\|_{\infty} \leq \left\| \frac{\partial g(x_1, x_2)}{\partial x_1} \right\|_{\infty} h_1 + \left\| \frac{\partial g(x_1, x_2)}{\partial x_2} \right\|_{\infty} h_2 \quad (1)$$

przyjmując $h=0.5$, uzyskujemy żadaną dokładność aproksymacji. W tym celu definiujemy 7 zbiorów rozmytych A_j z np. trójkątnymi funkcjami przynależności o postaci

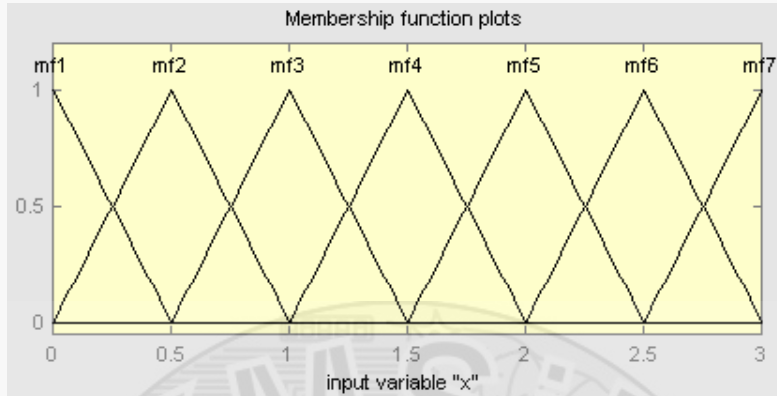
$$\begin{aligned} &\mu_{A_1}(0, 0, 0.5), \\ &\vdots \\ &\mu_{A_7}(2.5, 3, 3), \end{aligned} \quad (2)$$

oraz

$$\mu_{A_i}(x) = \mu_{A_i}(x; e_{j-1}, e_j, e_{j+1}), \quad (3)$$

gdzie $j=2,3,\dots,7$, oraz $e_j = 0 + h \cdot (j-1)$.

Stosując narzędzie projektowe *Fuzzy Logic Toolbox* zbudowano model rozmyty ze zbiorem 7 trójkątnych funkcji przynależności na wejściu, których rozkład przedstawiono na rys. 14.



Rys. 14. Rozkład funkcji przynależności zbiorów rozmytych wejścia x

Wartość wyjścia z projektowanego modelu rozmytego opiszemy zależnością

$$f(x^*) = y^* = \frac{\sum_{j=1}^7 \mu_{A_j}(x^*) \sin(e_j)}{\sum_{j=1}^7 \mu_{A_j}(x^*)}, \quad (4)$$

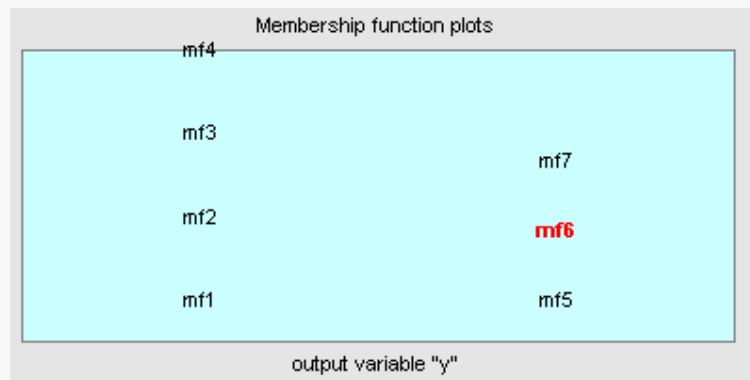
gdzie y^* to wartość wyjścia z modelu rozmytego, $\sin(e_j)$ to wartość zadanej nieliniowej funkcji dla wejścia do modelu rozmytego $x^* = e_j$.

Przyjęto bazę reguł

1. If [x is mf1] then [y is mf1] (1)
2. If [x is mf2] then [y is mf2] (1)
3. If [x is mf3] then [y is mf3] (1)
4. If [x is mf4] then [y is mf4] (1)
5. If [x is mf5] then [y is mf5] (1)
6. If [x is mf6] then [y is mf6] (1)
7. If [x is mf7] then [y is mf7] (1)

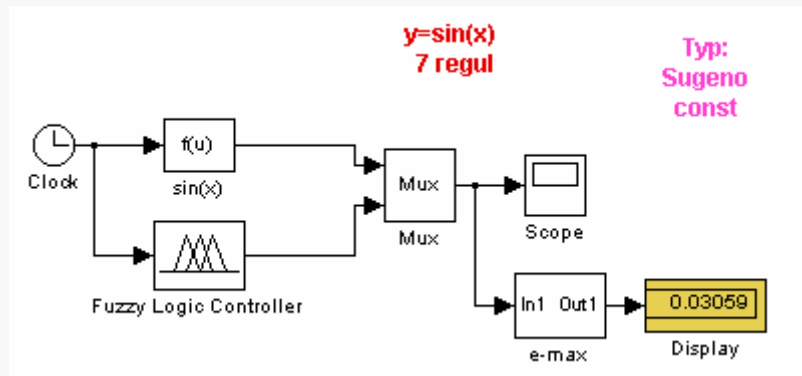
(9.12)

oraz konkluzje bazy reguł w postaci wartości ostrych (singletonów) przyjmujących wartości funkcji aproksymowanej $g(x) = \sin(e_j)$, $j=1,\dots,7$.



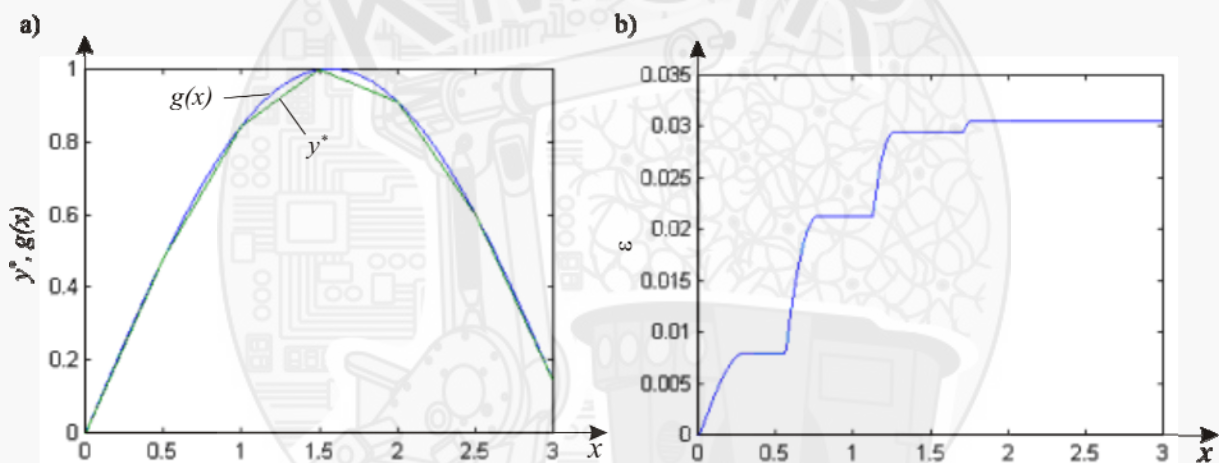
Rys. 15. Wartości konkluzji bazy reguł

W celu weryfikacji zaprojektowanego modelu rozmytego w modelu obliczeniowym wygenerowanym w Simulink-u (rys. 16.) porównano uzyskane przebiegi funkcji zadanej i jej oceny oraz obliczono maksymalny błąd interpolacji.



Rys. 16. Schemat układu w Simulink-u

Przebieg funkcji $y^*=f(x)$ wygenerowanej przez model rozmyty w porównaniu z przebiegiem funkcji $g(x)=\sin(x)$ przedstawiono na rys. 17.a), na rys. 17.b) przedstawiono błąd odwzorowania funkcji zadanej obliczony jako różnica $f(x)-g(x)$.



Rys. 17.a) Przebieg funkcji y^* wygenerowanej przez model rozmyty oraz funkcji $g(x)=\sin(x)$, b) różnica $\varepsilon = f(x)-g(x)$

Z przeprowadzonych symulacji wynika, że jest spełniony warunek dokładności modelu rozmytego $\sup_{x \in U} \|g(x) - f(x)\| < \varepsilon$.

Jeżeli projektowany jest system rozmyty aproksymujący funkcję $g(x_1, x_2) = 0.52 + 0.1x_1 + 0.28x_2 - 0.06x_1x_2$ zdefiniowaną na $U = [-1,1] \times [-1,1] \subset \mathbf{R}^2$ zadaną dokładnością $\varepsilon = 0.1$, to wówczas na podstawie (1) otrzymamy

$$\left\| \frac{\partial g}{\partial x_1} \right\|_{\infty} = \sup_{x \in U} |0.1 - 0.06x_2| = 0.16 \quad \text{oraz} \quad \left\| \frac{\partial g}{\partial x_2} \right\|_{\infty} = \sup_{x \in U} |0.28 - 0.06x_1| = 0.34.$$

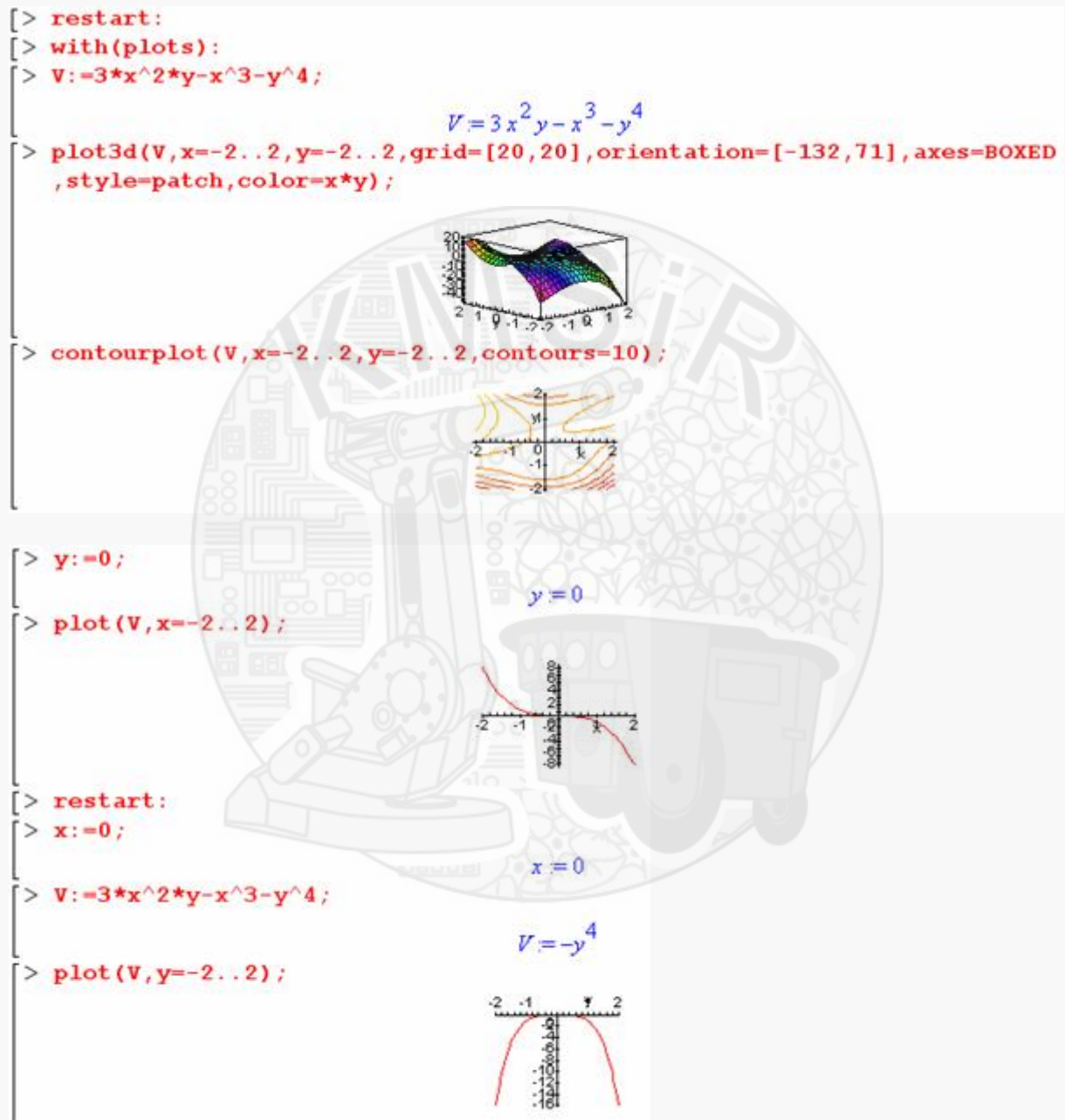
Przyjęcie $h_1 = h_2 = 0.2$ spełnia nierówność $\|g(x_1, x_2) - f(x_1, x_2)\|_{\infty} \leq 0.16 \cdot 0.2 + 0.34 \cdot 0.2 = 0.1$.

Należy wówczas zdefiniować dla każdej zmiennej 11 zbiorów rozmytych co daje $11 \times 11 = 121$ reguł rozmytych. Jednakże nie zawsze konieczne jest zastosowanie tak dużej liczby reguł dla osiągnięcia zakładanej dokładności odwzorowania.

3. Zadania do wykonania

Zaprojektować układ rozmyty aproksymujący ciągłą funkcję $g(x_1, x_2)$ zdefiniowaną na zbiorze $U = [-2, 2] \times [-2, 2] \subset \mathbf{R}^2$ z dokładnością ε ($\sup_{x \in U} (|g(x) - f(x)|) < \varepsilon$), stosując system wnioskujący Mamdaniego lub Takagi-Sugeno dla dowolnie wybranych funkcji przynależności.

3.1. Wyznaczyć graficzny przebieg zadanej nieliniowej funkcji



Nieliniową funkcję należy dobrać zgodnie z nr zespołu:

- $g(x_1, x_2) = x_1^2 - x_1x_2 + x_2^2$,
- $g(x_1, x_2) = x_1^2 - 2x_1x_2 + 2x_2^2 + 2x_1$,

- c) $g(x_1, x_2) = x_1 x_2 + \frac{1}{2}(x_1 + x_2)$,
- d) $g(x_1, x_2) = x_1^3 + x_2^3 - x_1^2 - 2x_1 x_2 - x_2^2$,
- e) $g(x_1, x_2) = x_1^3 - 2x_2^3 - 3x_1 + 6x_2$,
- f) $g(x_1, x_2) = x_1^3 - 2x_1^2 + 2x_2^2 + x_2^4$,
- g) $g(x_1, x_2) = x_1^2 - x_1 x_2 - x_2^2$,
- h) $g(x_1, x_2) = x_1^2 + x_1 x_2 + x_2^2 - 4 \ln(x_1 + 1)$,

3.2. Wyznaczyć model rozmyty zadanej nieliniowej funkcji

Tab. 1. Parametry dokładności odwzorowania zadanej nieliniowej funkcji

nr zespołu	1.	2.	3.	4.	5.	6.	7.	8.
ε	0.2	0.1	0.15	0.35	0.25	0.4	0.5	0.3

Należy przeprowadzić badania symulacyjne dla układu z logiką rozmytą, sprawdzając wpływ typu zastosowanych funkcji przynależności do zbiorów rozmytych oraz ich rozłożenia w przestrzeni wejść, na dokładność aproksymacji zadanej nieliniowej funkcji.

Sprawozdanie powinno zawierać:

1. Wstęp teoretyczny
 - podstawowe wiadomości na temat układów z logiką rozmytą,
 - podstawowe wiadomości na temat pakietu Fuzzy Logic Toolbox.
2. Przebieg ćwiczenia
 - obliczenia związane z doбором wartości h_1 oraz h_2 ,
 - obrazy modelu Simulink'a modelu rozmytego,
 - wykresy funkcji przynależności do zbiorów rozmytych dla wejść x_1 , x_2 oraz wyjścia modelu rozmytego y (patrz rys. 14),
 - bazę reguł modelu rozmytego,
 - graficzną interpretację powierzchni modelu rozmytego (patrz rys. 10).
3. Wyniki symulacji (analogicznie jak w przykładzie 1).
 - graficzny przebieg zadanej nieliniowej funkcji $g(x_1, x_2)$ (jak w 3.1),
 - przebieg funkcji zadanej $g(x_1, x_2)$ i wyjścia z modelu rozmytego y^* , oraz błędu odwzorowania e (analogicznie jak na rys. 17), dla $x_1 = -2, x_1 = -1, x_1 = 0, x_1 = 1, x_1 = 2, x_2 \in \langle -2, 2 \rangle$,
4. Wnioski

Uwaga. Każdy realizowany podpunkt sprawozdania powinien być odpowiednio skomentowany.