

IMPLEMENTAREA ÎN MEDIUL MATLAB-SIMULINK A UNOR STRATEGII DE MODELARE BAZATE PE LOGICA FUZZY (2)

SISTEME DE INFERENȚĂ FUZZY. CONTROL FUZZY

3. Breviar teoretic

3.3. Sisteme de inferență

Un sistem de **inferență fuzzy** constă dintr-o bază de reguli fuzzy (motiv pentru care mai este denumit sistem bazat pe reguli fuzzy), un dicționar care definește funcțiile de apartenență utilizate de către regulile fuzzy și un mecanism de inferență, care, pe baza unei ipoteze produce o concluzie aplicând anumite reguli impuse.

O regulă din baza de reguli are informații de tip fuzzy în antecedent. Consecința unei reguli este de asemenea de tip fuzzy în sistemele de inferență de tip de tip Mamdani sau Larsen și de tip crisp în sistemele de inferență de tip Takagi-Sugeno.

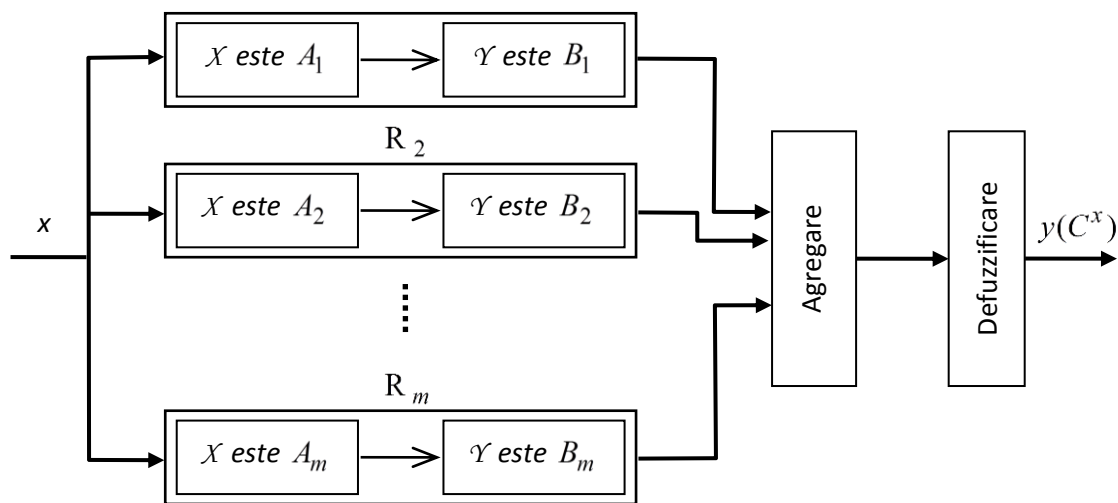


Figura 1. Schema bloc a unui sistem de inferență de tip Mamdani sau Larsen.

Figura 1 prezintă schema bloc a unui sistem de inferență de tip Mamdani sau Larsen. Diferența dintre cele două tipuri de sisteme este dată de operatorul utilizat pentru implicație.

O bază de reguli fuzzy

R_1 : IF X este A_1 THEN Y este B_1

R_2 : IF X este A_2 THEN Y este B_2

...

R_m : IF X este A_m THEN Y este B_m

poate fi privită ca reuniunea acestora, $R = \bigcup_{i=1}^m R_i$. Considerând ipoteza $X = x$, pentru obținerea concluziei regulile $R_1, R_2, \dots, R_m$ sunt *agregate* utilizând un *operator de agregare* de tip T-

conormă, $A : [0,1]^m \rightarrow [0,1]$. Astfel, considerând baza de reguli $R_1 - R_m$ de mai sus cu operatorul de implicație $I(a,b)$, pentru ipoteza $X = x$, concluzia este reprezentată de mulțimea fuzzy C^x având funcția de apartenență $\mu_{C^x} : Y \rightarrow [0,1]$ precizată prin

$$\mu_{C^x}(y) = A \left[I(\mu_{A_1}(x), \mu_{B_1}(y)), I(\mu_{A_2}(x), \mu_{B_2}(y)), \dots, I(\mu_{A_m}(x), \mu_{B_m}(y)) \right], y \in Y.$$

De exemplu, într-un sistem de inferență de tip Mamdani operatorul de implicație utilizat este $I_M(a,b) = \min\{a,b\}$, iar operatorul de agregare este T-conorma maxim, $A_M(a_1, a_2, \dots, a_m) = \max\{a_1, a_2, \dots, a_m\}$. Astfel, considerând ipoteza $X = x$ pentru baza de reguli $R_1 - R_m$ de mai sus, concluzia este reprezentată de mulțimea fuzzy C^x având funcția de apartenență

$$\mu_{C^x} : Y \rightarrow [0,1], \mu_{C^x}(y) = \max_{i=1,m} \left\{ \min\{\mu_{A_i}(x), \mu_{B_i}(y)\} \right\}.$$

Aceeași conormă este utilizată și în sistemul de inferență de tip Larsen, însă împreună cu operatorul de implicație $I_L(a,b) = a \cdot b$. În acest sistem de inferență, considerând ipoteza $X = x$ pentru aceeași bază de reguli $R_1 - R_m$, concluzia este reprezentată de mulțimea fuzzy C^x având funcția de apartenență

$$\mu_{C^x} : Y \rightarrow [0,1], \mu_{C^x}(y) = \max_{i=1,m} \left\{ \mu_{A_i}(x) \cdot \mu_{B_i}(y) \right\}.$$

Într-un mod intuitiv, cele prezentate mai sus pot fi generalizate la cazul sistemelor de inferență de tip Mamdani sau Larsen cu n intrări și o ieșire, pentru care baza de reguli are forma:

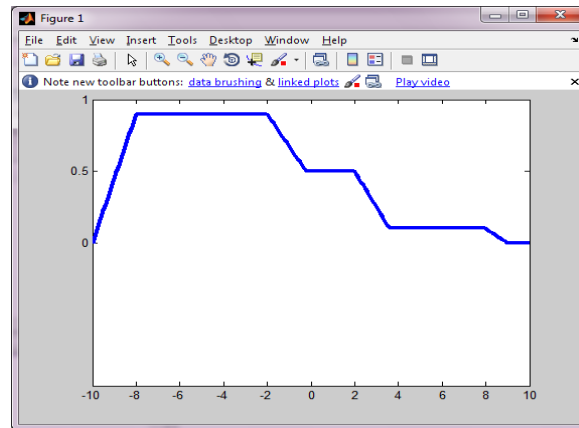
$$R_j : \text{IF } (X_1 \text{ este } A_{j,1} \text{ AND } X_2 \text{ este } A_{j,2} \text{ AND } \dots \text{ AND } X_n \text{ este } A_{j,n}) \\ \text{THEN } Y \text{ este } B_j, j = \overline{1, m}.$$

În urma aplicării regulilor din baza de reguli a sistemului de inferență se obține întotdeauna o mulțime fuzzy. În cazul în care este necesar ca ieșirea sistemului de inferență să fie de tip crisp se utilizează **o metodă de defuzzificare** pentru a extrage o valoare care reprezintă mulțimea fuzzy de ieșire.

Pentru prezentarea unor metode de defuzzificare notăm cu C mulțimea fuzzy definită pe spațiul Y (mulțime continuă sau discretă de numere reale) prin funcția de apartenență μ_C obținută prin agregarea regulilor de inferență.

Să presupunem că avem de defuzzificat următoarea regiune:

```
% Defuzzificare
x = -10:0.1:10;
mf1 = trapmf(x, [-10 -8 -2 2]);
mf2 = trapmf(x, [-5 -3 2 4]);
mf3 = trapmf(x, [2 3 8 9]);
mf1 = max(0.5*mf2, max(0.9*mf1, 0.1*mf3));
figure(1);
plot(x, mf1, 'LineWidth', 3);
set(gca, 'YLim', [-1 1], 'YTick', [0 .5 1]);
```



Metoda centroidului (eng. *center of area – COA*) returnează abscisa centrului de greutate al suprafeței situate între axa absciselor și graficul funcției de apartenență μ_C .

Dacă Y este mulțime continuă, rezultatul defuzzificării este dat de

$$y_{COA}(C) = \frac{\int_{c_1}^{c_2} y \mu_C(y) dy}{\int_{c_1}^{c_2} \mu_C(y) dy},$$

unde $[c_1, c_2]$ este un interval ce conține suportul lui μ_C sau, dacă acesta este nemărginit, $[c_1, c_2]$ este intervalul de interes pentru aplicația considerată.

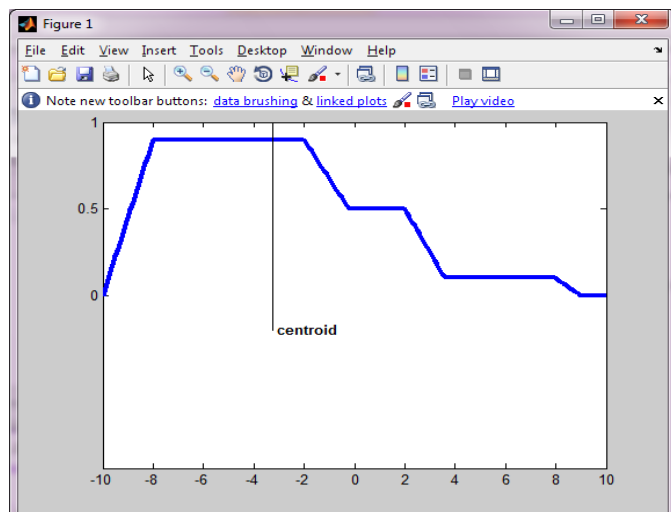
Dacă Y este mulțime discretă, $Y = \{y_1, y_2, \dots, y_k\}$, abscisa centroidului este dată de

$$y_{COA}(C) = \frac{\sum_{i=1}^k y_i \mu_C(y_i)}{\sum_{i=1}^k \mu_C(y_i)}.$$

Dacă $y_{COA}(C)$ nu aparține mulțimii Y , se alege elementul cel mai apropiat de această valoare.

```
% Metoda centroidului
x1 = defuzz(x,mf1,'centroid')
h1 = line([x1 x1],[-0.2 1.2],'Color','k');
t1 = text(x1,-0.2,' centroid','FontWeight','bold');
```

x1 = -3.2281



Metoda bisectoarei ariei (eng. *bisector of area – BOA*) returnează abscisa $y_{BOA}(C)$ corespunzătoare dreptei de ecuație $y = y_{BOA}(C)$ ce împarte suprafața situată între axa absciselor și graficul funcției de apartenență μ_C în două arii egale.

Dacă Y este mulțime continuă, $y_{BOA}(C)$ se alege astfel încât

$$\int_{c_1}^{y_{BOA}(C)} \mu_C(y) dy = \int_{y_{BOA}(C)}^{c_2} \mu_C(y) dy.$$

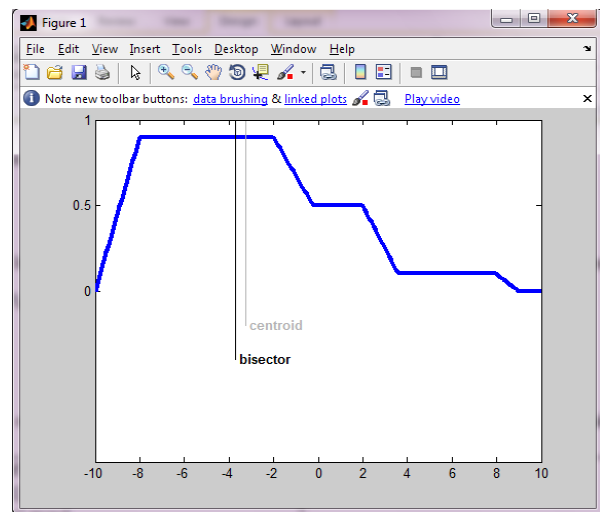
În cazul în care $Y = \{y_1, y_2, \dots, y_m\}$, considerând punctele ordonate în ordine crescătoare, $y_1 < y_2 < \dots < y_k$, $y_{BOA}(C)$ este dat de

$$\sum_{y=y_1}^{y_{BOA}(C)} \mu_C(y) = \sum_{y=y_{BOA}(C)}^{y_k} \mu_C(y).$$

Dacă $y_{BOA}(C)$ nu aparține mulțimii Y , se alege elementul cel mai apropiat de această valoare.

```
% Metoda bisectoarei ariei
x2 = defuzz(x,mf1,'bisector')
gray = 0.7*[1 1 1];
set([h1 t1], 'Color', gray)
h2 = line([x2 x2], [-0.4 1.2], 'Color', 'k');
t2 = text(x2, -0.4, ' bisector', 'FontWeight', 'bold');
```

x2 = -3.7000



Metoda centrului maximelor (eng. *center of maxima – COM*) consideră mulțimea punctelor în care funcția de apartenență μ_C își atinge valoarea maximă (care reprezintă înălțimea mulțimii fuzzy C , $h(C)$), $M = \{y \in [c_1, c_2] \mid \mu_C(y) = h(C)\}$, și returnează media aritmetică a extremelor sale

$$y_{COM}(C) = \frac{1}{2} [\inf(M) + \sup(M)].$$

În cazul discret, pentru $Y = \{y_1, y_2, \dots, y_k\}$, rezultatul defuzzificării este

$$y_{COM}(C) = \frac{1}{2} [\min\{y_j \mid y_j \in M\} + \max\{y_j \mid y_j \in M\}]$$

unde

$$M = \{y_j \in Y \mid \mu_C(y_j) = h(C)\}.$$

Metoda mediei maximelor (eng. *mean of maxima – MOM*) este definită numai în cazul discret și furnizează media aritmetică a tuturor punctelor în care funcția de apartenență μ_C își atinge valoarea maximă $h(C)$:

$$y_{MOM}(C) = \frac{1}{|M|} \sum_{y_k \in M} y_k$$

unde $M = \{y_j \in Y \mid \mu_C(y_j) = h(C)\}$ și $|M|$ notează cardinalul (numărul de elemente al) mulțimii M .

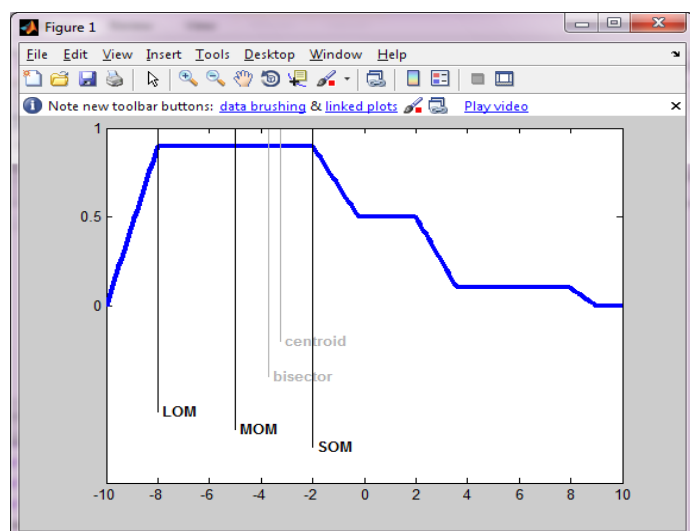
Metodele de defuzzificare **minimul maximelor** (eng. *smallest of maxima – SOM*) și **maximul maximelor** (eng. *largest of maxima – LOM*) returnează marginea inferioară, respectiv cea superioară, a mulțimii M a punctelor în care funcția de apartenență μ_C își atinge valoarea maximă

$$y_{SOM}(C) = \inf(M), \quad y_{LOM}(C) = \sup(M).$$

În cazul discret, dacă $y_{COM}(C)$, $y_{SOM}(C)$ sau $y_{LOM}(C)$ nu aparține mulțimii Y , se alege elementul cel mai apropiat de această valoare.

```
% Metoda mediei maximelor
% Metoda minimului maximelor
% Metoda maximului maximelor
x3 = defuzz(x,mf1,'mom')
x4 = defuzz(x,mf1,'som')
x5 = defuzz(x,mf1,'lom')
set([h2 t2], 'Color', gray)
h3 = line([x3 x3], [-0.7 1.2], 'Color', 'k');
t3 = text(x3, -0.7, ' MOM', 'FontWeight', 'bold');
h4 = line([x4 x4], [-0.8 1.2], 'Color', 'k');
t4 = text(x4, -0.8, ' SOM', 'FontWeight', 'bold');
h5 = line([x5 x5], [-0.6 1.2], 'Color', 'k');
t5 = text(x5, -0.6, ' LOM', 'FontWeight', 'bold');
```

```
x3 = -5
x4 = -2
x5 = -8
```



Alegerea metodei

```
% Alegerea metodei
set([h3 t3 h4 t4 h5 t5], 'Color', gray)
set([h1 t1], 'Color', 'red')
```

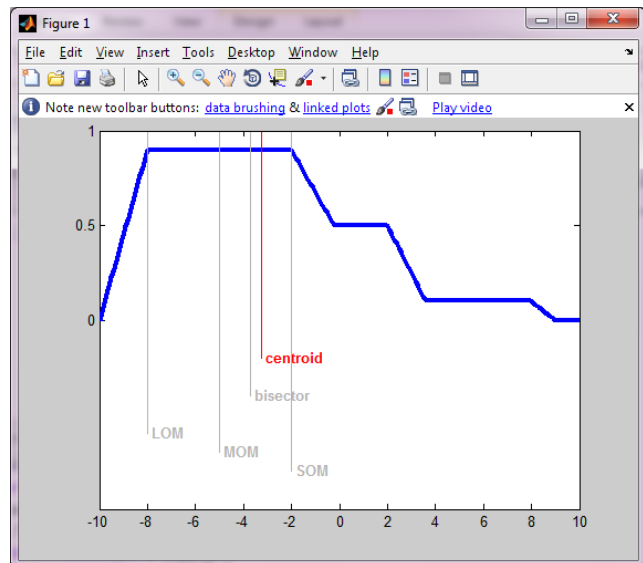


Figura 2 prezintă schema bloc a unui sistem de inferență de tip Takagi-Sugeno-Kang (TSK).

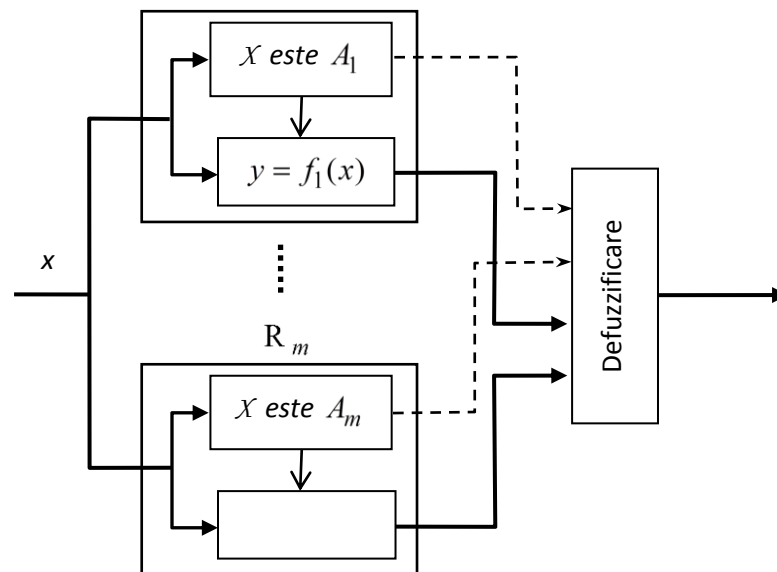


Figura 2. Schema bloc a unui sistem de inferență de tip Takagi-Sugeno-Kang.

Baza de reguli a unui asemenea sistem are forma

$$R_1: \text{IF } X \text{ este } A_1 \text{ THEN } y = f_1(x)$$

$$R_2: \text{IF } X \text{ este } A_2 \text{ THEN } y = f_2(x)$$

... ..

$$R_m: \text{IF } X \text{ este } A_m \text{ THEN } y = f_m(x)$$

unde $f_1, f_2, \dots, f_m: X \rightarrow Y$. În practică, X și Y sunt intervale sau mulțimi discrete de numere reale.

În ipoteza $X = x$, valoarea de adevăr a propoziției din antecedentul regulii R_j , $w_j = \mu_{A_j}(x)$,

reprezintă puterea regulii R_j în ansamblul sistemului de reguli considerat. Această valoare caracterizează contribuția pe care, în ipoteza considerată, ieșirea regulii R_j $y_j = f_j(x)$, o are la formarea ieșirii y a sistemului de inferență TSK.

Pentru a obține ieșirea y se utilizează una din următoarele metode:

- **suma ponderată** (eng. *weighted sum* – WS) a ieșirilor individuale ale regulilor:

$$y_{WS} = \sum_{j=1}^m \mu_{A_j}(x) \cdot y_j = \sum_{j=1}^m \mu_{A_j}(x) \cdot f_j(x);$$

- **media ponderată** (eng. *weighted average* – WA) a ieșirilor regulilor:

$$y_{WA} = \frac{1}{\sum_{j=1}^m \mu_{A_j}(x)} \sum_{j=1}^m \mu_{A_j}(x) \cdot y_j = \frac{1}{\sum_{j=1}^m \mu_{A_j}(x)} \sum_{j=1}^m \mu_{A_j}(x) \cdot f_j(x).$$

Modul de funcționare a unui sistem de inferență de tip TSK prezentat anterior poate fi generalizat cu ușurință în cazul unui sistem cu n intrări și o ieșire, când baza de reguli are forma:

$$R_j: \text{IF } (X_1 \text{ este } A_{j,1} \text{ AND } X_2 \text{ este } A_{j,2} \text{ AND } \dots \text{ AND } X_n \text{ este } A_{j,n}) \\ \text{THEN } y = f_j(x_1, x_2, \dots, x_n), \quad j = \overline{1, m},$$

cu $f_1, f_2, \dots, f_m: X_1 \times X_2 \times \dots \times X_n \rightarrow Y$, unde $X_i, i = \overline{1, n}$, și Y reprezintă intervale sau mulțimi discrete de numere reale.

Dacă funcțiile f_j sunt constante, se spune că sistemul TSK are ordinul zero, iar dacă aceste funcții sunt liniare, sistemul TSK are ordinul 1.

Următorul exemplu ilustrează modul de funcționare a unui sistem de inferență de tip TSK. Ieșirea sistemului depinde atât de operatorii logici utilizați în sistemul respectiv cât și de metoda de defuzificare aplicată.

3.4. Aplicații ale sistemelor fuzzy. Control fuzzy.

Modul în care un sistem de inferență fuzzy poate fi utilizat în construirea variantei fuzzy a unui regulator PID pentru controlul unui proces în timp continuu este prezentat în figura 3. Mărimile de intrare în regulatorul fuzzy sunt eroarea e , derivata erorii, $\dot{e}$ (variația erorii Δe în cazul sistemelor în timp discret), și integrala acesteia, $\int e$ (respectiv suma discretă). La ieșire, regulatorul acesta furnizează derivata mărimii de comandă, $\dot{u}$, astfel că este necesară inserarea unui integrator între sistemul fuzzy și procesul controlat.

Blocurile proporționale K_p , K_i , K_d sunt necesare, pe de o parte pentru a permite „acordarea” regulatorului fuzzy; pe de altă parte, aceste blocuri permit scalarea mărimilor de intrare în regulator, operație necesară deoarece regulatoarele fuzzy standard au domenii fixe de variație a intrărilor (de exemplu, intervalul $[-1, 1]$).

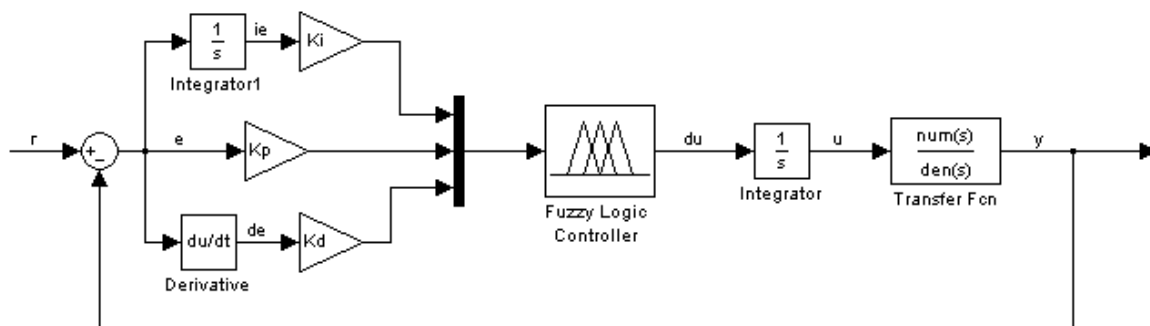


Figura 3. Schema bloc a unui sistem cu control fuzzy în timp continuu.

În mod tradițional, în **regulatele fuzzy** se utilizează sisteme de inferență de tip Mamdani/Larsen, pentru care baza de reguli conține reguli de forma

IF e este ... AND $\dot{e}$ este ... AND $\ddot{e}$ este ... THEN $\dot{u}$ este ...

sau sisteme de inferență de tip Takagi-Sugeno-Kang, pentru care se consideră reguli de forma

IF e este ... AND $\dot{e}$ este ... AND $\ddot{e}$ este ... THEN $\dot{u} = f(e, \dot{e}, \ddot{e})$

În **proiectarea unui regulator fuzzy**, utilizatorul este liber să aleagă:

- numărul de termeni lingvistici care caracterizează fiecare mărime de intrare: de regulă acești termeni descriu în limbaj natural modul de variație a mărimii pe care o caracterizează (se utilizează valori de genul negativ mare, negativ mic, zero, pozitiv mic, pozitiv mare, la care se adaugă calificatori);
- tipul și parametrii funcțiilor de apartenență corespunzătoare acestora;
- operatorii logici fuzzy (disjuncție, conjuncție, implicație);
- regulile bazei de reguli a sistemului de inferență;
- procedura de defuzzificare.

Pentru implementarea sistemului de inferență fuzzy se utilizează interfața grafică cu utilizatorul **fuzzy**, ce face parte din pachetul de programe **Fuzzy Toolbox** pentru MATLAB. Editarea unui *sistem de inferență fuzzy* (FIS) cu ajutorul acestei interfețe se realizează prin:

- precizarea tipului de FIS implementat: Mamdani/Larsen sau Takagi-Sugeno-Kang;
- selectarea operatorilor utilizați în FIS (conjuncție, disjuncție, implicație, agregare, defuzzificare). Pentru negație se utilizează operatorul fuzzy standard, $N(a) = 1 - a$, $a \in [0, 1]$.
- setarea numărului de variabile de intrare/ieșire din FIS, a numelor (variabile lingvistice) și domeniilor de variație ale acestora (vezi figura 8);
- modificarea numărului de termeni lingvistici care caracterizează fiecare variabilă în parte și precizarea funcțiilor de apartenență corespunzătoare;
- precizarea regulilor bazei de reguli;
- vizualizarea reprezentării grafice a dependenței ieșirii de intrări.

4. Probleme propuse

Problema 15.

Se consideră un sistem simplu de inferență cu două intrări și o ieșire, având o singură regulă:

$R : \text{IF } (X_1 \text{ este } A_1 \text{ AND } X_2 \text{ este } A_2) \text{ THEN } Y \text{ este } B$

unde A_1 , A_2 și B sunt mulțimi fuzzy triunghiulare date de $\mu_{A_1}(x) = f_{\text{tri}}(x; 0, 2, 4)$, $\mu_{A_2}(x) = f_{\text{tri}}(x; 3, 4, 5)$, respectiv $\mu_B(x) = f_{\text{tri}}(x; 0, 2, 6)$, definite pe intervalul $[0, 6]$. În acest sistem de inferență se consideră operatorul standard pentru conjuncție (min).

La intrarea sistemului de inferență se aplică $X_1 = 3$ și $X_2 = 4$.

Indicații:

Se observă că $\mu_{A_1}(3) = 0.5$ și $\mu_{A_2}(4) = 1$, astfel că valoarea de adevăr a propoziției „3 este A_1 AND 4 este A_2 ” este $i = \min\{\mu_{A_1}(3), \mu_{A_2}(4)\} = 0.5$.

Aplicând **implicația Mamdani** $I_M(i, c) = \min\{i, c\}$ pentru $i = 0.5$, concluzia corespunzătoare regulii R se obține ca fiind mulțimea fuzzy C_M cu funcția de apartenență:

$$\mu_{C_M} : [0, 6] \rightarrow [0, 1], \quad \mu_{C_M}(z) = \min\left\{\min\left\{\mu_{A_1}(3), \mu_{A_2}(4)\right\}, \mu_B(z)\right\}.$$

Aplicând **implicația Larsen** $I_L(i, c) = i \cdot c$ pentru $i = 0.5$ se obține concluzia corespunzătoare regulii R ca fiind mulțimea fuzzy C_L cu funcția de apartenență:

$$\mu_{C_L} : [0, 6] \rightarrow [0, 1], \quad \mu_{C_L}(z) = \min\left\{\mu_{A_1}(3), \mu_{A_2}(4)\right\} \cdot \mu_B(z).$$

Prin defuzzificarea mulțimilor C_M și C_L prin cinci metode prezentate mai sus se obțin valorile următoare:

- metoda centrului de greutate al ariei: $z_{COA}(C_M) = 2.778$, $z_{COA}(C_L) = 2.667$;
- metoda bisectoarei ariei: $z_{BOA}(C_M) = 2.75$, $z_{BOA}(C_L) = 2.55$;
- metoda mediei maximelor: $z_{MOM}(C_M) = 2.50$, $z_{MOM}(C_L) = 2$;
- metoda minimului maximelor: $z_{SOM}(C_M) = 1$, $z_{SOM}(C_L) = 2$;
- metoda maximului maximelor: $z_{LOM}(C_M) = 4$, $z_{LOM}(C_L) = 2$.

Figura 4 ilustrează grafic aplicarea operatorilor fuzzy considerați.

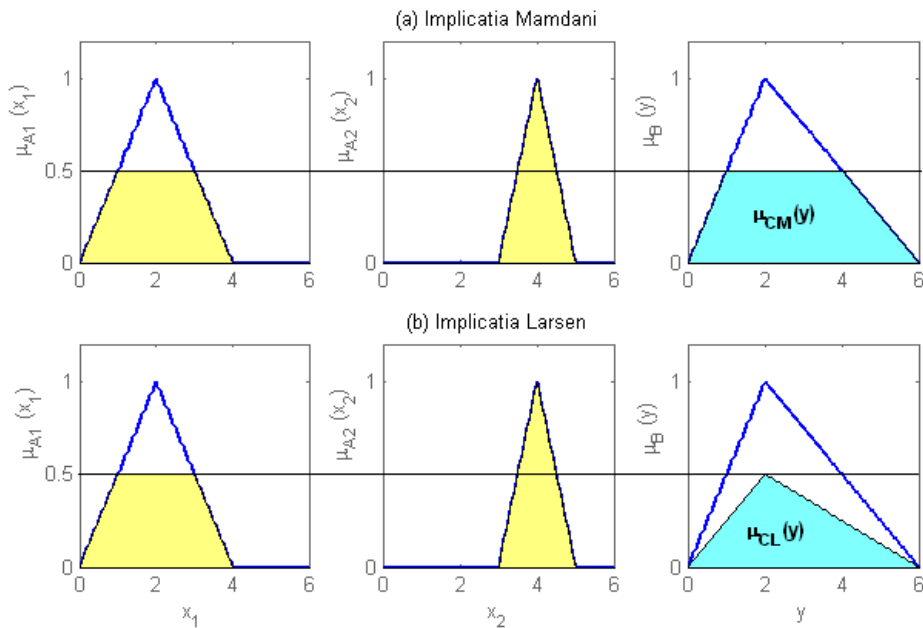


Figura 4. Reprezentarea grafică a aplicării implicației (a) Mamdani și (b) Larsen pentru regula R dacă $X_1 = 3$ și $X_2 = 4$.

Figura 5 prezintă graficele funcțiilor de apartenență ale mulțimilor C_M (sus) și C_L (jos) precum și rezultatele defuzzificării acestora prin cele cinci metode.

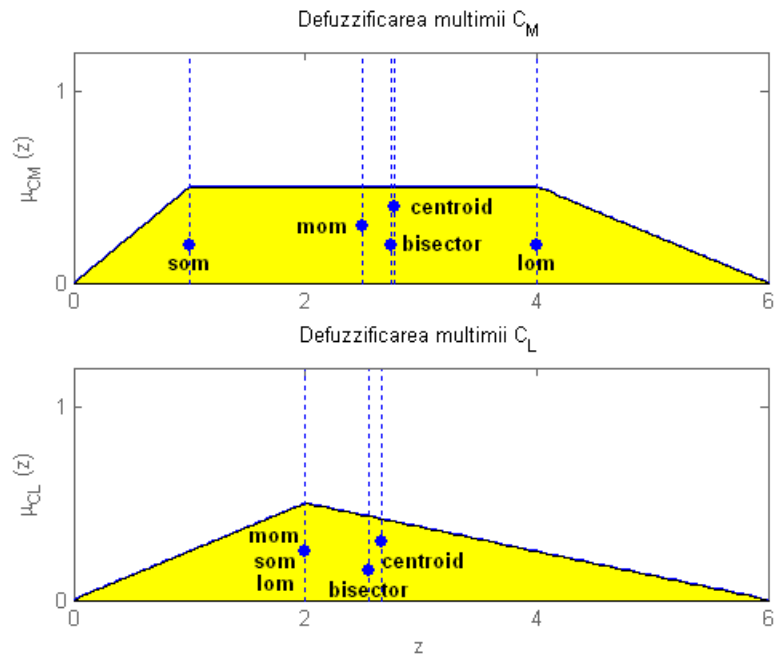


Figura 5. Defuzzificarea mulțimilor C_M și C_L .

Scrieți codul matlab pentru implementarea celor cinci metode de defuzzyficare.

Problema 16.

Se consideră un sistem de inferență TSK de ordinul zero cu două intrări și o ieșire, având două reguli:

$$R_1: \text{IF } (X_1 \text{ este } A_{1,1} \text{ AND } X_2 \text{ este } A_{1,2}) \text{ THEN } y = -1$$

$$R_2: \text{IF } (X_1 \text{ este } A_{2,1} \text{ AND } X_2 \text{ este } A_{2,2}) \text{ THEN } y = 1$$

unde $A_{1,1}$, $A_{1,2}$, $A_{2,1}$ și $A_{2,2}$ sunt mulțimi fuzzy trapezoidale definite pe intervalul $[0, 6]$ prin $\mu_{A_{1,1}}(x) = \mu_{A_{2,2}}(x) = f_{\text{trap}}(x; 0, 0, 1, 6)$, $\mu_{A_{1,2}}(x) = \mu_{A_{2,1}}(x) = f_{\text{trap}}(x; 0, 5, 6, 6)$.

Indicații:

La intrarea sistemului de inferență se aplică $X_1 = 2$ și $X_2 = 3$. Utilizând reprezentarea grafică din figura 6, se observă că $\mu_{A_{1,1}}(2) = 0.8$, $\mu_{A_{1,2}}(3) = 0.6$, $\mu_{A_{2,1}}(2) = 0.4$ și $\mu_{A_{2,2}}(3) = 0.6$.

Ieșirile celor două reguli sunt $y_1 = -1$ și $y_2 = 1$

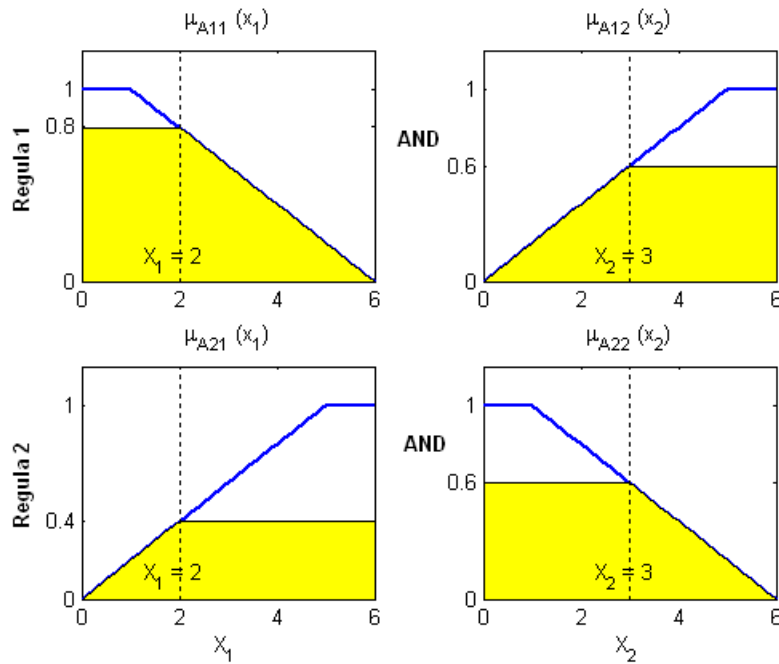


Figura 6. Reprezentarea grafică a funcțiilor de apartenență ce intervin în regulile sistemului de inferență de tip TSK.

Considerând operatorul standard pentru conjuncție (min), valoarea de adevăr a propoziției „2 este $A_{1,1}$ AND 4 este $A_{1,2}$ ” este $w_1 = \min\{\mu_{A_{1,1}}(2), \mu_{A_{1,2}}(3)\} = 0.6$, iar cea a propoziției „2 este $A_{2,1}$ AND 4 este $A_{2,2}$ ” este $w_2 = \min\{\mu_{A_{2,1}}(2), \mu_{A_{2,2}}(3)\} = 0.4$. Ieșirile sistemului TSK corespunzătoare celor două metode de ponderare a ieșirilor individuale ale regulilor sunt:

$$y_{WS} = w_1 y_1 + w_2 y_2 = 0.6 \cdot (-1) + 0.4 \cdot 1 = -0.2,$$

$$y_{WA} = \frac{w_1 y_1 + w_2 y_2}{w_1 + w_2} = \frac{0.6 \cdot (-1) + 0.4 \cdot 1}{0.6 + 0.4} = -0.2.$$

Dacă în sistemul de inferență considerat se utilizează produsul algebric ca operator logic de conjuncție, ponderile regulilor R_1 și R_2 sunt $\tilde{w}_1 = \mu_{A_{1,1}}(2) \cdot \mu_{A_{1,2}}(3) = 0.48$, respectiv $\tilde{w}_2 = \mu_{A_{2,1}}(2) \cdot \mu_{A_{2,2}}(3) = 0.24$. În această situație ieșirile sistemului TSK corespunzătoare celor două metode de ponderare a ieșirilor regulilor sunt:

$$\tilde{y}_{WS} = \tilde{w}_1 y_1 + \tilde{w}_2 y_2 = 0.48 \cdot (-1) + 0.24 \cdot 1 = -0.24,$$

$$\tilde{y}_{WA} = \frac{\tilde{w}_1 y_1 + \tilde{w}_2 y_2}{\tilde{w}_1 + \tilde{w}_2} = \frac{0.48 \cdot (-1) + 0.24 \cdot 1}{0.48 + 0.24} = -\frac{1}{3}.$$

Problema 17.

Se consideră sistemul având funcția de transfer $G(s) = e^{-s} \frac{1.2}{10s+1}$ (element de ordinul I cu timp mort). Pentru controlul acestui proces cu un regulator fuzzy de tip PD se consideră modelul Simulink prezentat în figura 7, realizat după modelul general din figura 3. Regulatorul fuzzy are două intrări: eroarea e și derivata erorii, $\dot{e}$, scalate cu factorii $K_p = 0.25$ și, respectiv, $K_d = 1.50$, furnizând la ieșire derivata comenzii, $\dot{u}$.

Indicații:

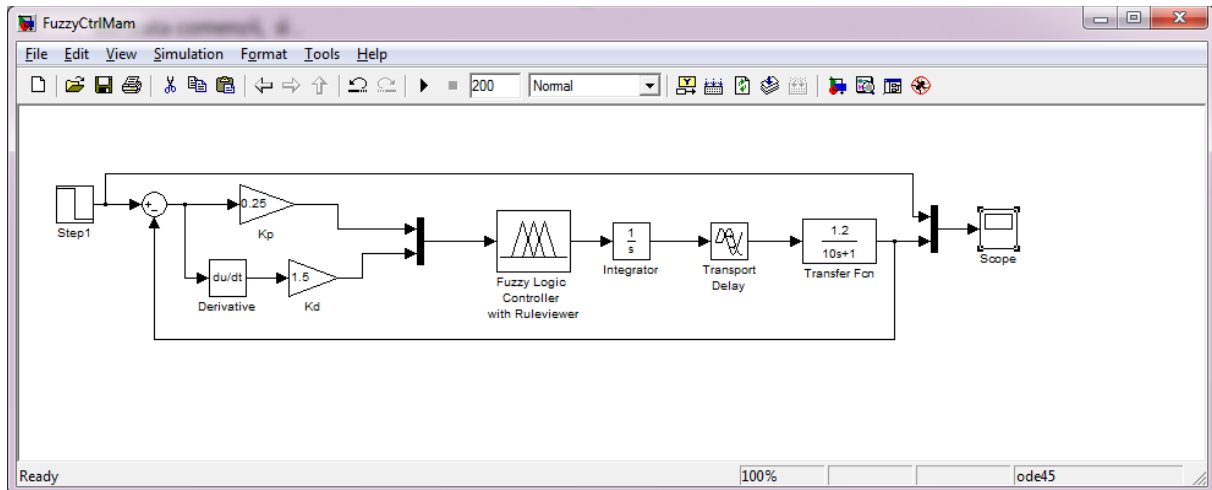


Figura 7. Modelul Simulink al sistemului studiat.

Pentru caracterizarea mărimilor de intrare (eroare scalată E , derivata erorii scalată DE) se utilizează câte doi termeni lingvistici: *negativ* (Neg) și *pozitiv* (Poz), considerați mulțimi fuzzy trapezoidale având funcțiile de apartenență (figura H.2.4):

$$\mu_{neg}(x) = f_{trap}(x; -2.8, -2, -0.8, 0.8), \quad \mu_{poz}(x) = f_{trap}(x; -0.8, 0.8, 2, 2.8).$$

Pentru caracterizarea mărimii de ieșire (derivata comenzii du) se utilizează trei termenii lingvistici: *negativ* (Neg), *pozitiv* (Poz) și *zero* ($Zero$), considerați mulțimi fuzzy triunghiulare având funcțiile de apartenență (figura H.2.5):

$$\mu_{neg}(x) = f_{tri}(x; -2, -1, 0), \quad \mu_{poz}(x) = f_{tri}(x; 0, 1, 2), \quad \mu_{zero}(x) = f_{tri}(x; -1, 0, 1).$$

Intervalul de interes pentru cele trei mărimi utilizate în această aplicație este $[-1, 1]$.

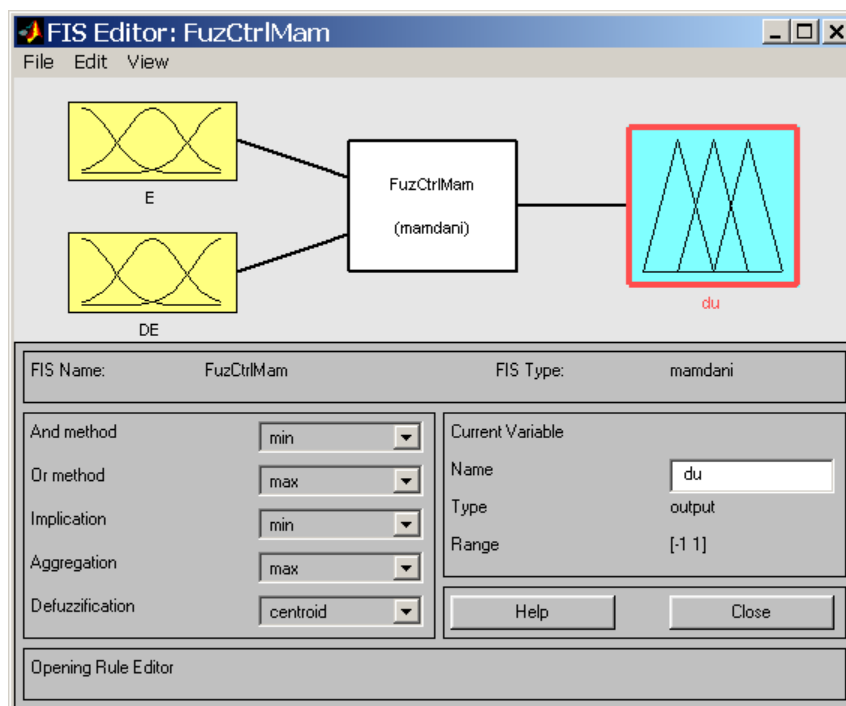


Figura 8. Sistemul de inferență de tip Mamdani.

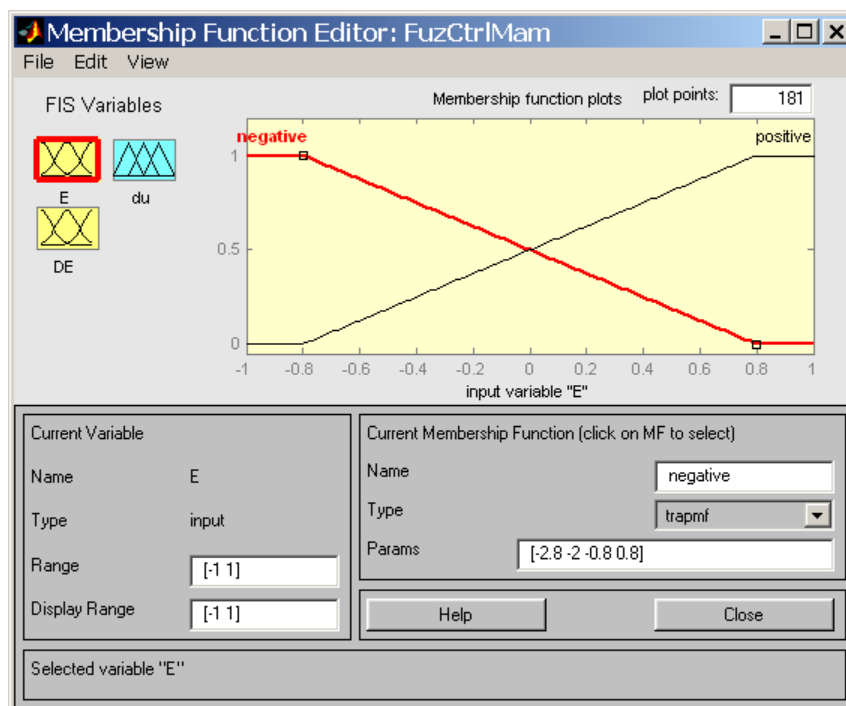


Figura 9. Termenii lingvistici corespunzători mărimilor de intrare.

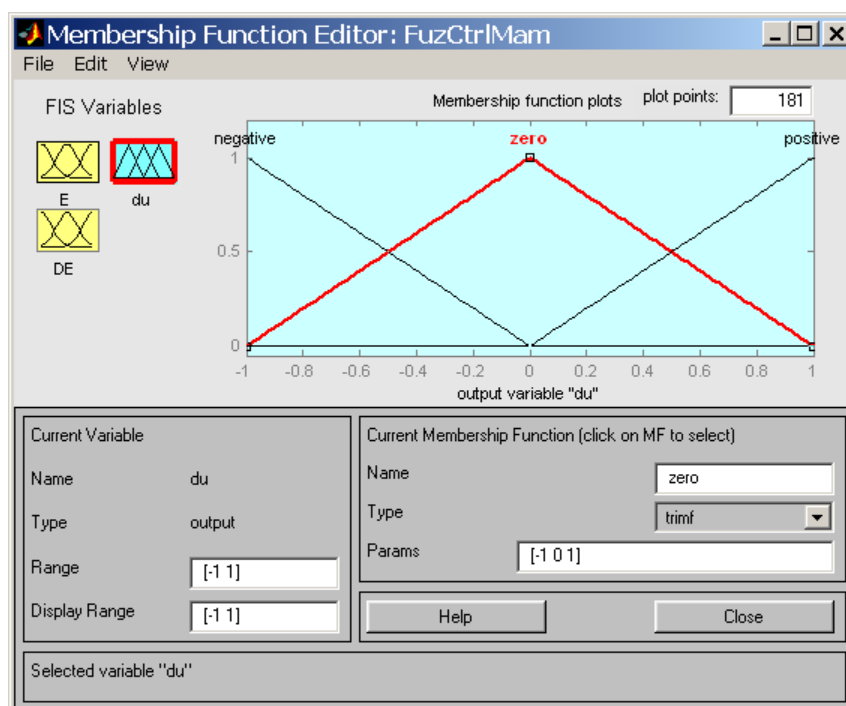


Figura 10. Termenii lingvistici corespunzători mărimii de ieșire.

Pentru regulatorul de tip Mamdani se consideră cele patru reguli reprezentate în tabelul H.2.1 și figura H.2.6, și anume:

R_1 : IF E este Poz AND DE este Poz THEN du este Poz

R_2 : IF E este poz AND DE este Neg THEN du este Zero

R_3 : IF E este Neg AND DE este Poz THEN du este Zero

R_4 : IF E este Neg AND DE este Neg THEN du este Neg

Tabelul 1. Tabelul de inferență pentru regulatorul fuzzy de tip Mamdani.

E \ DE	N	P
N	N	Z
P	Z	P

Așa cum se poate observa în figura 7, în FIS de tip Mamdani implementat se utilizează operatorii logici fuzzy standard și defuzzificarea rezultatului se realizează prin metoda centrului de greutate. Reprezentarea grafică a acestor reguli este prezentată în figura 12.

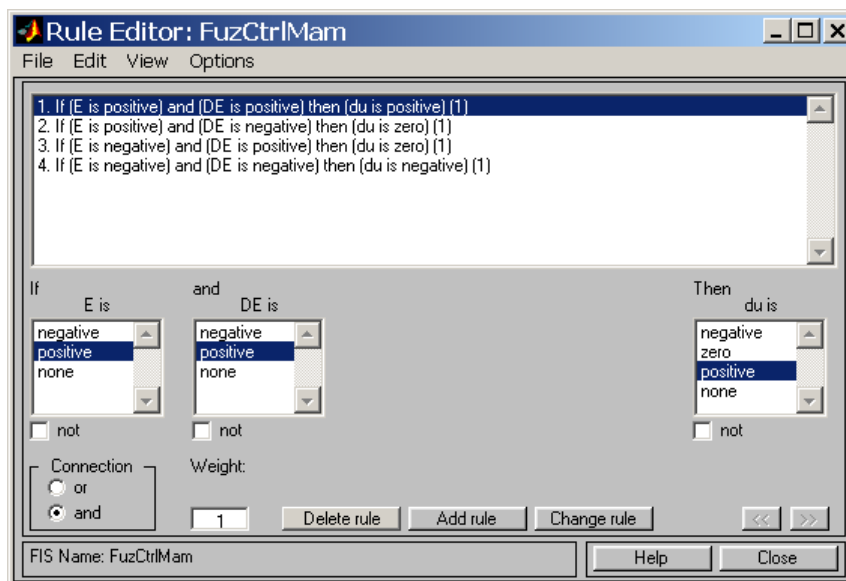


Figura 11. Baza de reguli a regulatorului fuzzy de tip Mamdani.

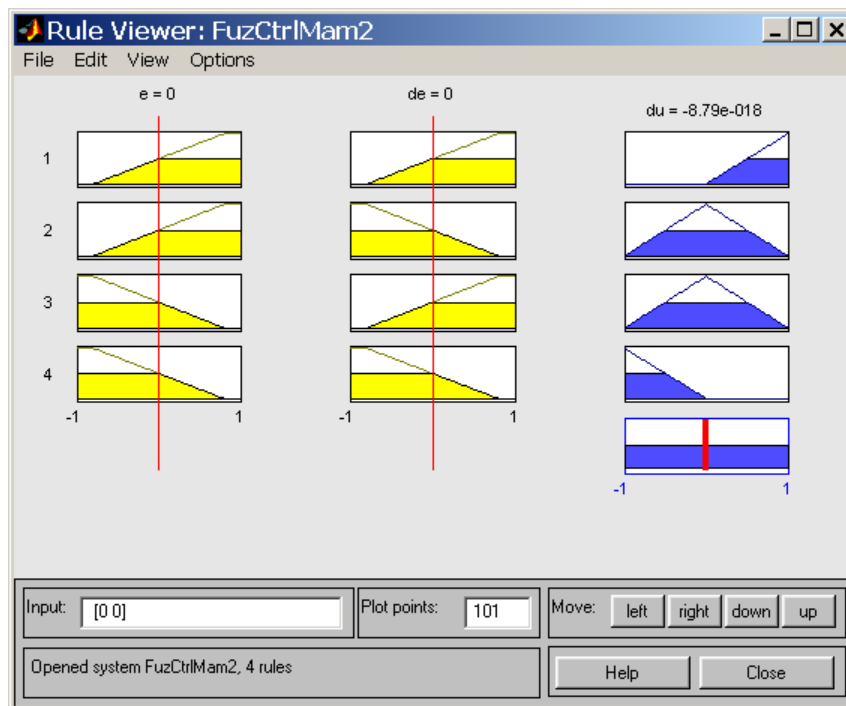


Figura 12. Vizualizarea grafică a regulilor FIS de tip Mamdani.

Figura 13 prezintă reprezentarea grafică a dependenței ieșirii de cele două intrări pentru acest FIS de tip Mamdani.

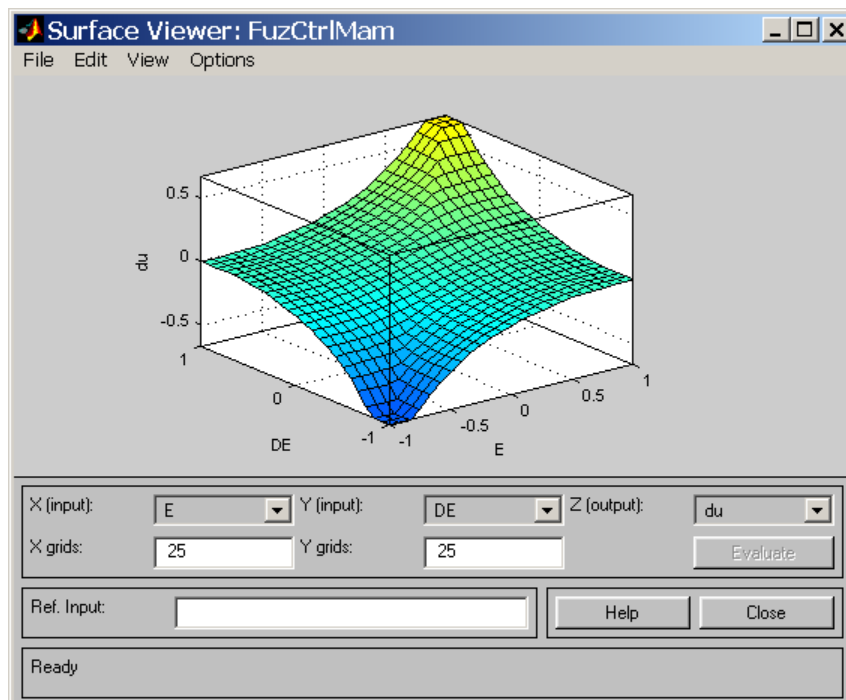


Figura 13. Reprezentarea grafică a dependenței ieșirii de cele două intrări pentru FIS de tip Mamdani.

Figura 14.a prezintă reprezentarea grafică a referinței și ieșirii procesului considerat dacă se utilizează un regulatorul fuzzy de tip Mamdani prezentat anterior.

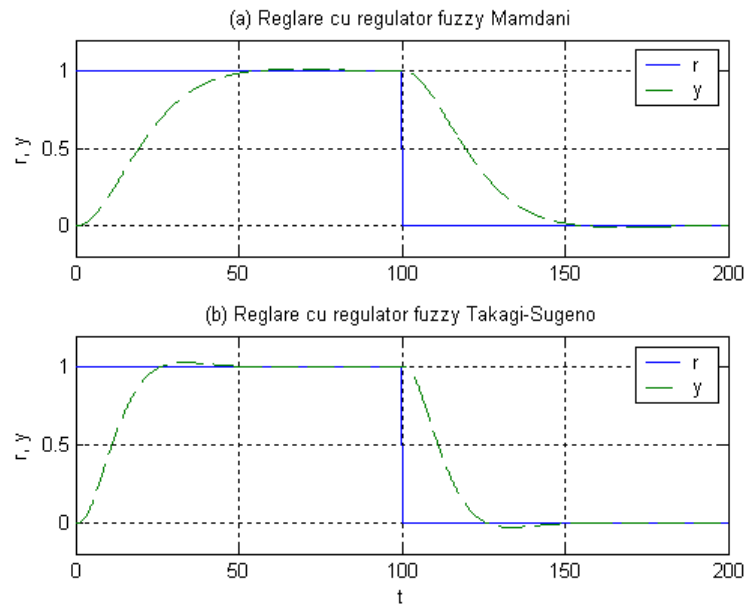


Figura 14. Reprezentarea grafică a referinței și ieșirii procesului dacă se utilizează un regulator fuzzy (a) de tip Mamdani și (b) de tip TSK.

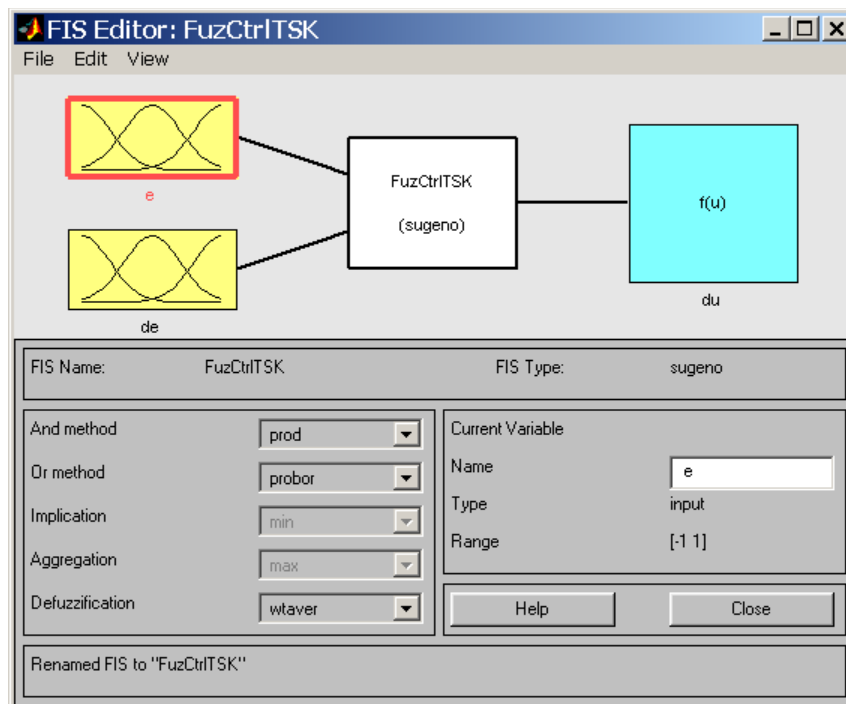


Figura 15. Sistemul de inferență de tip Takagi-Sugeno utilizat.

Pentru a ilustra modul de implementare în MATLAB a unui regulator fuzzy de tip TSK de ordin zero considerăm același proces, condus de un regulator de tip PID. Se utilizează aceleași mărimi de intrare în FIS (figura 15), caracterizate prin aceiași termeni lingvistici ca și în cazul utilizării FIS de tip Mamdani. Pentru mărimea de ieșire din FIS se consideră funcțiile constante

$$f_1(E, DE) = 1, \quad f_2(E, DE) = -1, \quad f_3(E, DE) = 0.$$

Pentru regulatorul de tip Mamdani se consideră cele patru reguli reprezentate în tabelul 2 și figura 16, și anume:

R_1 : IF E este Poz AND DE este Poz THEN $du = f_1(E, DE)$

R_2 : IF E este Poz AND DE este Neg THEN $du = f_3(E, DE)$

R_3 : IF E este Neg AND DE este Poz THEN $du = f_3(E, DE)$

R_4 : IF E este Neg AND DE este Neg THEN $du = f_2(E, DE)$

Tabelul 2. Tabelul de inferență pentru regulatorul fuzzy de tip TSK.

E \ DE	N	P
	N	P
N	$\dot{u} = -1$	$\dot{u} = 0$
P	$\dot{u} = 0$	$\dot{u} = 1$

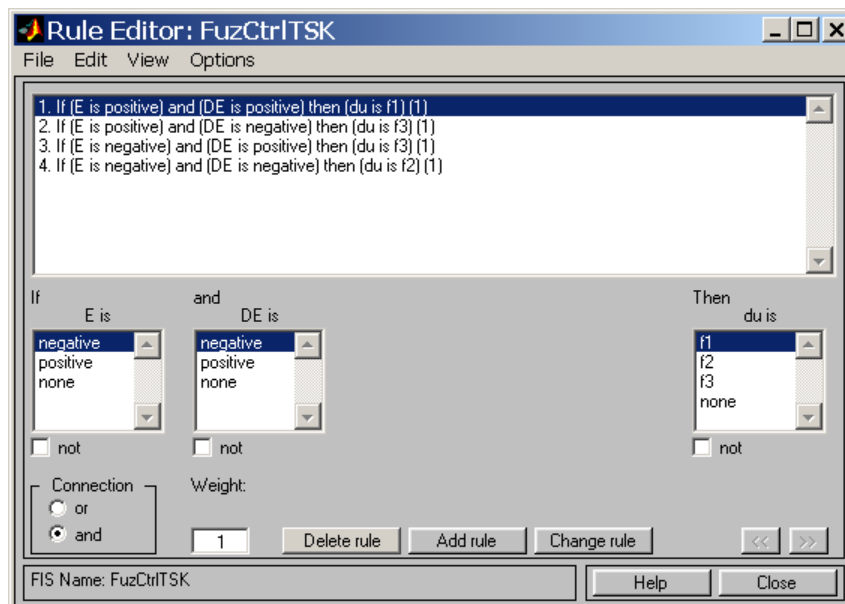


Figura 16. Baza de reguli a regulatorului fuzzy de tip TSK utilizat.

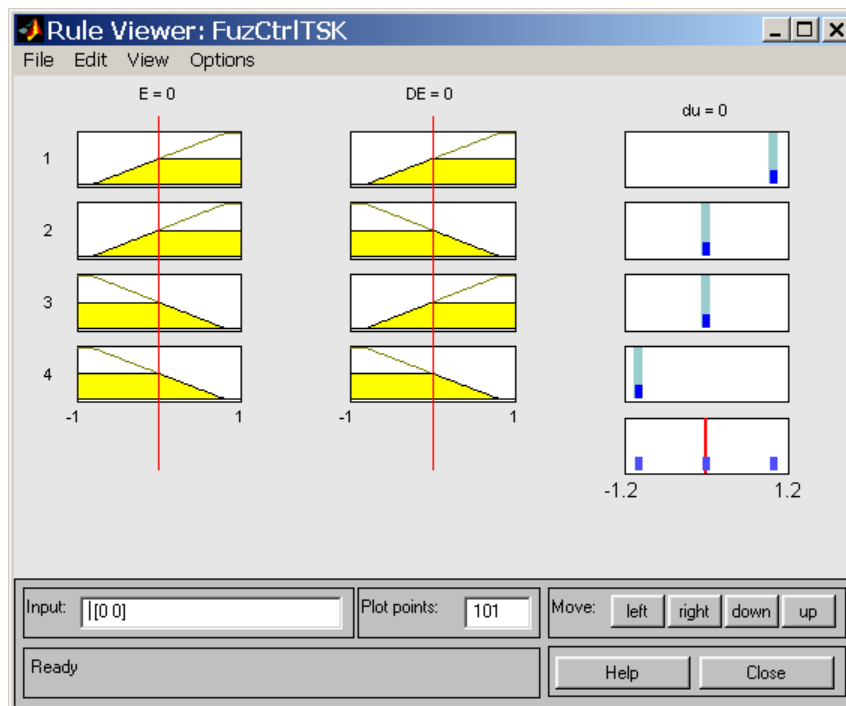


Figura 17. Vizualizarea grafică a regulilor FIS de tip TSK utilizat.

Pentru acest FIS de tip TSK se utilizează operatorii logici fuzzy produs pentru conjuncție și suma algebrică (probabilistic OR) pentru disjuncție (așa cum se poate observa în figura 15). Metoda utilizată pentru defuzzificarea rezultatului este media ponderată a ieșirilor. Figura 17 permite vizualizarea grafică a regulilor acestui sistem de inferență iar figura 18 prezintă grafic dependența ieșirii de cele două intrări.

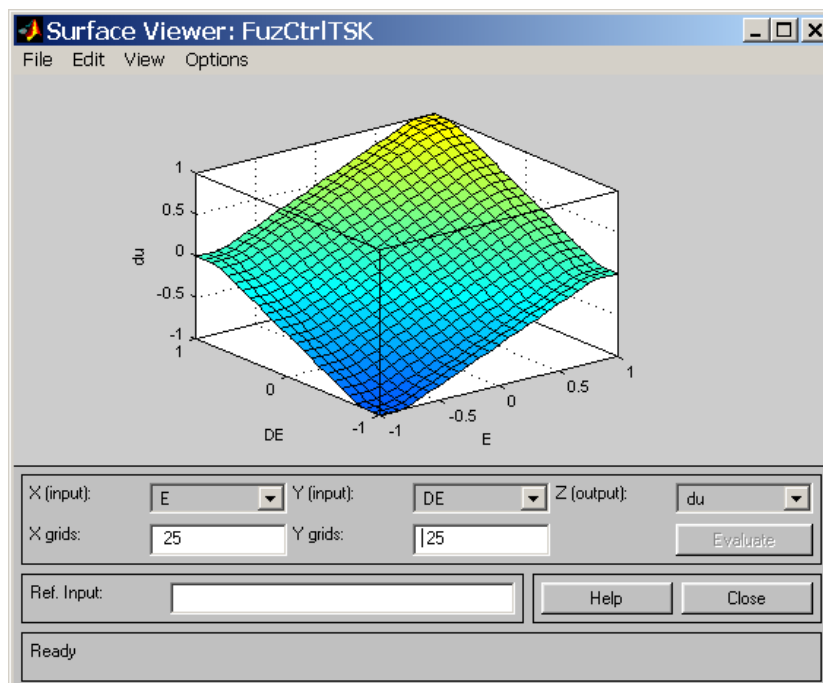


Figura 18. Reprezentarea grafică a dependenței ieșirii de cele două intrări pentru FIS de tip TSK utilizat.

Problema 18.

Utilizând inferența, să se găsească valorile funcției de apartenență pentru I-triunghi isoscel, R-triunghi dreptunghic, O-triunghi oarecare în cazul triunghiului cu unghiurile

- i) $45^\circ, 65^\circ, 70^\circ$;
- ii) $60^\circ, 40^\circ, 80^\circ$;
- iii) $75^\circ, 55^\circ, 50^\circ$.

Indicații:

$$45^\circ, 75^\circ, 60^\circ;$$

$$U = \{x = 75^\circ \geq y = 60^\circ \geq z = 45^\circ, x + y + z = 180^\circ\}$$

$$\mu_I(u) = 1 - \frac{1}{60} \min(x - y, y - z)$$

$$\mu_R(u) = 1 - \frac{1}{90} |x - 90|$$

$$\mu_O(u) = \min(1 - \mu_I(u), 1 - \mu_R(u))$$

Problema 19.

Se dau două seturi fuzzy P și Q definite pe X după cum urmează:

$\mu(x)$	x_1	x_2	x_3	x_4	x_5
P	0.1	0.2	0.7	0.5	0.4
Q	0.9	0.6	0.3	0.2	0.8

Să se calculeze tăieturile lambda pentru:

- (i) $(\bar{P})_{0.2}$ (iii) $(P \cup Q)_{0.5}$ (v) $(Q \cup \bar{P})_{0.8}$
- (ii) $(\bar{Q})_{0.3}$ (iv) $(P \cap Q)_{0.4}$ (vi) $(P \cup \bar{P})_{0.2}$

Indicații

Tăietura lambda $A_\lambda = \{x | \mu_A(x) \geq \lambda\}$

$$\bar{Q} = \left\{ \frac{0.1}{x_1} + \frac{0.4}{x_2} + \frac{0.7}{x_3} + \frac{0.8}{x_4} + \frac{0.2}{x_5} \right\}$$

$$(\bar{Q})_{0.3} = \left\{ \frac{0}{x_1} + \frac{1}{x_2} + \frac{1}{x_3} + \frac{1}{x_4} + \frac{0}{x_5} \right\}$$

Problema 20.

Pe universul de discurs X, se definesc seturile fuzzy A, B și C având următoarele funcții de apartenență:

(i) $X = \{0, 1, 2, 3\}$

$$\mu_A(x) = \frac{2}{x+3} \quad \mu_B(x) = \frac{4x}{2(x+5)}$$

- a) Desenați funcțiile de apartenență.
- b) Definiți elementele de pe axa x corespunzătoare tăieturilor λ pe seturile fuzzy A și B pentru următoarele valori ale lui λ : 0.2, 0.5 și 0.6.

(ii) $X = [0, 5]$

$$\mu_A(x) = \frac{1}{1+5(x-5)^2} \quad \mu_B(x) = 2^{-x} \quad \mu_C(x) = \frac{2x}{x+5}$$

- a) Desenați funcțiile de apartenență.
- b) Definiți elementele de pe axa x corespunzătoare tăieturilor λ pe seturile fuzzy A, B și C pentru următoarele valori ale lui λ : 0.2, 0.4, 0.7 și 0.9.

Problema 21.

Pentru relația fuzzy R

(i)

$$R = \begin{bmatrix} 1 & 0.2 & 0.3 \\ 0.5 & 0.9 & 0.6 \\ 0.4 & 0.8 & 0.7 \end{bmatrix}$$

Să se calculeze tăietura lambda pentru următoarele valori ale lui λ : a) 0^+ ; b) 0.2; c) 0.9 și d) 0.5.

(ii)

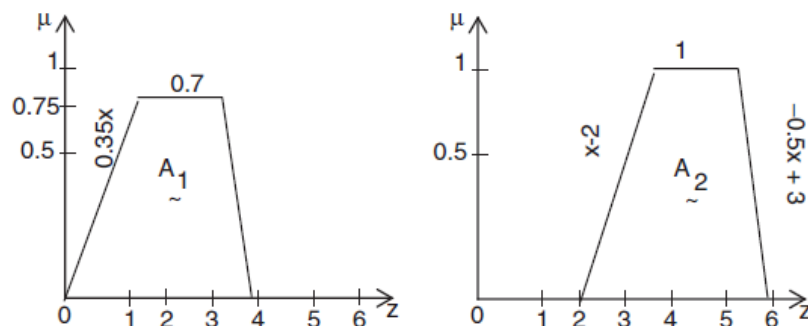
$$R = \begin{bmatrix} 0.2 & 0.5 & 0.47 & 1 & 0.9 \\ 0.3 & 0.5 & 0.6 & 1 & 0.8 \\ 0.4 & 0.6 & 0.4 & 0.5 & 0.3 \\ 0.9 & 1 & 0.3 & 0.3 & 0.2 \end{bmatrix}$$

Să se calculeze tăietura lambda pentru următoarele valori ale lui λ : a) 0.4; b) 0.7 și c) 0.8.

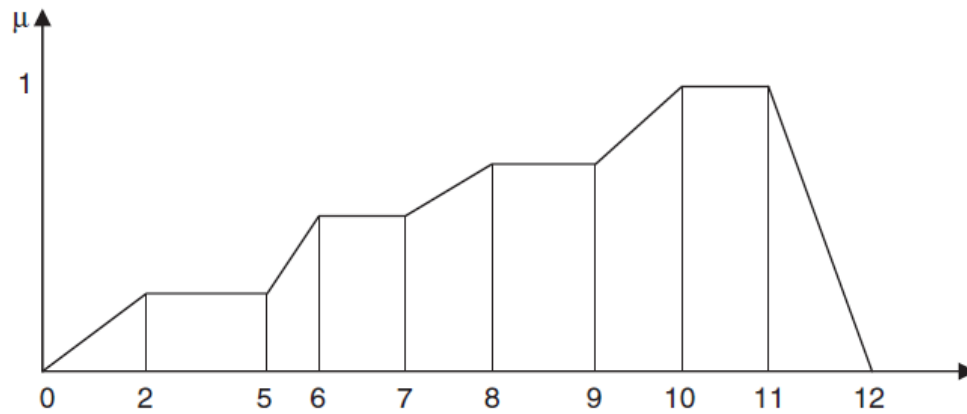
```
clear all;
R=input('Introduceti matricea R = ');
lambda=input('Introduceti valoarea lui lambda = ');
[m, n] = size(R);
for i = 1:m
    for j = 1:n
        if(R(i,j) < lambda)
            b(i,j) = 0;
        else
            b(i,j) = 1;
        end
    end
end
display(b)
```

Problema 22.

Pentru funcția de apartenență din figura următoare să se determine valoarea defuzzificată a ieșirii prin șase metode:

**Problema 23.**

- (i) Folosind defuzzyficarea prin metoda centroidului, să se convertească valoarea fuzzy z la valoarea Z^* pentru următoarea reprezentare grafică a lui z :



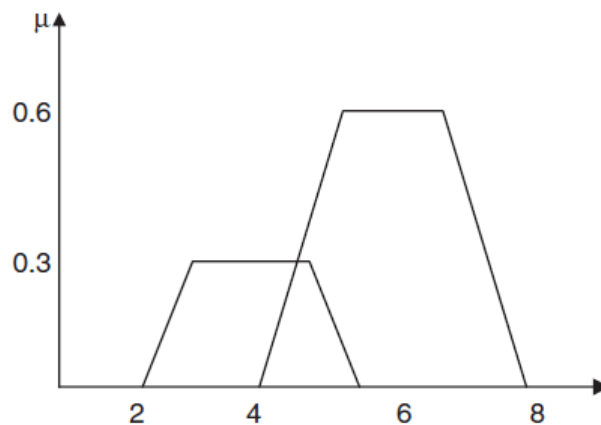
Se știe că: $\mu_z(2) = 0.25$; $\mu_z(6) = 0.5$; $\mu_z(8) = 0.75$; $\mu_z(10) = 1$.

```
x = 0:0.1:12;
mf1 = trapmf(x,[0 2 5 5]);
mf2 = trapmf(x,[4 6 7 7]);
mf3 = trapmf(x,[5 8 9 9]);
mf4 = trapmf(x,[6 10 11 12]);

mf6 = max(max(0.25*mf1,0.5*mf2),max(0.75*mf3,mf4));
plot(x,0.25*mf1,x,0.5*mf2,x,0.75*mf3,x,mf4);
hold on;
plot(x,mf6,'LineWidth',3);
set(gca,'YLim',[0 1.5],'YTick',[0 .5 1]);

x1 = defuzz(x,mf3,'centroid')
h1 = line([x1 x1],[-0.2 1.2],'Color','k');
t1 = text(x1,-0.2,' centroid','FontWeight','bold');
x1 = 7.62
```

(ii) Să se calculeze valoarea defuzzyficată prin metoda mediei maximelor pentru:



```
x = 0:0.1:8;
mf1 = trimf(x,[2 4 6]);
mf2 = trimf(x,[4 6 8]);

mf3 = max(min(0.3,mf1),min(0.6,mf2));
plot(x,min(0.3,mf1),x,min(0.6,mf2));
hold on;
plot(x,mf3,'LineWidth',3);
set(gca,'YLim',[0 1.5],'YTick',[0 .5 1]);

x3 = defuzz(x,mf3,'mom')
h3 = line([x3 x3],[-0.7 1.2],'Color','k');
t3 = text(x3,-0.7,' MOM','FontWeight','bold');
x3 = 6
```

Problema 24.

$$\text{Fie } y = -2x + x^2.$$

Să se formeze un system fuzzy care aproximează funcția f , când $x \in [-10, 10]$.

Metoda 1.

```
% Generarea datelor de intrare si de iesire
x = [-10:.5:10]';
y = -2 * x - x .* x;
figure(1);
plot(x,y);
grid;
xlabel('x');
ylabel('f(x)');
title('Caracteristica neliniara');
% Formatarea datelor pentru genfis1 si anfis
data = [x y];
trndata = data(1:2:size(x),:);
chkdata = data(2:2:size(x),:);
figure(2);
plot(trndata(:,1),trndata(:,2),'o',chkdata(:,1),chkdata(:,2),'x');
xlabel('x');
ylabel('f(x)');
title('Datele masurate');
grid;
% Initializarea sistemului fuzzy cu comanda genfis1
% Se utilizeaza 5 functii de apartenenta bell
nu = 5;
mftype = 'gbellmf';
fismat = genfis1(trndata, nu, mftype);
figure(3);
plotmf(fismat, 'input',1);
xlabel('x');
ylabel('f(x)');
title('Funcțiile de apartenenta initiale');
grid;
% Aplicarea comenzii anfis pentru gasirea celui mai potrivit sistem FIS
iterations = 100;
numep = 100;
[parab, trnerr, ss, parabcheck, chkerr] = anfis(trndata, fismat, numep, [], chkdata);
% Evaluarea iesirii sistemului FIS pentru x la intrare
anfi = evalfis(x, parab);
figure(4);
plot(trndata(:,1),trndata(:,2),'o',chkdata(:,1),chkdata(:,2),'x',x,anfi,'-');
grid;
xlabel('x');
ylabel('f(x)');
title('Calitatea aproximarii');
```

ANFIS info:

Number of nodes: 24
 Number of linear parameters: 10
 Number of nonlinear parameters: 15
 Total number of parameters: 25
 Number of training data pairs: 21
 Number of checking data pairs: 20
 Number of fuzzy rules: 5

Warning: number of data is smaller than number of modifiable parameters

Start training ANFIS ...

1	1.11035	1.11725
2	1.1055	1.11205
3	1.10065	1.10685
4	1.09578	1.10165

5	1.09091	1.09644
Step size increases to 0.011000 after epoch 5.		
6	1.08602	1.09123
7	1.08063	1.08548
8	1.07523	1.07974
9	1.06982	1.07399
Step size increases to 0.012100 after epoch 9.		
10	1.0644	1.06823
11	1.05842	1.06189
12	1.05242	1.05555
13	1.04642	1.0492
Step size increases to 0.013310 after epoch 13.		
14	1.0404	1.04285
15	1.03376	1.03586
16	1.02711	1.02886
17	1.02045	1.02186
Step size increases to 0.014641 after epoch 17.		
18	1.01378	1.01485
19	1.00642	1.00714
20	0.999053	0.999431
21	0.991671	0.991716
Step size increases to 0.016105 after epoch 21.		
22	0.984276	0.983999
23	0.976129	0.975509
24	0.96797	0.967018
25	0.959799	0.958527
Step size increases to 0.017716 after epoch 25.		
26	0.951619	0.950038
27	0.942609	0.940701
28	0.93359	0.931368
29	0.924563	0.92204
Step size increases to 0.019487 after epoch 29.		
30	0.91553	0.912717
31	0.905587	0.90247
32	0.89564	0.892233
33	0.88569	0.882008
Step size increases to 0.021436 after epoch 33.		
34	0.875741	0.871796
35	0.864798	0.860579
36	0.853859	0.849382
37	0.842927	0.838206
Step size increases to 0.023579 after epoch 37.		
38	0.832004	0.827053
39	0.820002	0.814815
40	0.808017	0.80261
41	0.796053	0.790441
Step size increases to 0.025937 after epoch 41.		
42	0.784111	0.77831
43	0.771006	0.765013
44	0.757936	0.751767
45	0.744905	0.738577
Step size increases to 0.028531 after epoch 45.		

46	0.731916	0.725445
47	0.717682	0.711069
48	0.703507	0.696771
49	0.689397	0.682553
Step size increases to 0.031384 after epoch 49.		
50	0.675355	0.668419
51	0.659993	0.652972
52	0.644722	0.637635
53	0.629548	0.622409
Step size increases to 0.034523 after epoch 53.		
54	0.614475	0.6073
55	0.598016	0.590817
56	0.581688	0.574482
57	0.565496	0.558299
Step size increases to 0.037975 after epoch 57.		
58	0.549443	0.54227
59	0.531951	0.524819
60	0.514637	0.507561
61	0.497503	0.490498
Step size increases to 0.041772 after epoch 61.		
62	0.480553	0.473633
63	0.462123	0.45531
64	0.44392	0.437229
65	0.425947	0.41939
Step size increases to 0.045950 after epoch 65.		
66	0.408204	0.401794
67	0.388953	0.382718
68	0.369981	0.363933
69	0.351286	0.345439
Step size increases to 0.050545 after epoch 69.		
70	0.332868	0.327232
71	0.312925	0.307534
72	0.293311	0.288179
73	0.274022	0.269162
Step size increases to 0.055599 after epoch 73.		
74	0.255055	0.250481
75	0.234557	0.230315
76	0.21444	0.210549
77	0.194699	0.191182
Step size increases to 0.061159 after epoch 77.		
78	0.175332	0.172215
79	0.15446	0.151822
80	0.134046	0.131939
81	0.114104	0.112599
Step size increases to 0.067275 after epoch 81.		
82	0.0946681	0.0938651
83	0.0739639	0.0741053
84	0.0541988	0.0555814
85	0.0360048	0.0391241
Step size increases to 0.074002 after epoch 85.		
86	0.0211079	0.026463
87	0.0107584	0.0173792

88	0.0173364	0.0221371
89	0.00977813	0.0162056
90	0.0165201	0.0217109
91	0.00934652	0.0162587

Step size decreases to 0.066602 after epoch 91.

92	0.016098	0.021832
93	0.00734626	0.0157186
94	0.0155904	0.0219795
95	0.00732519	0.0161811

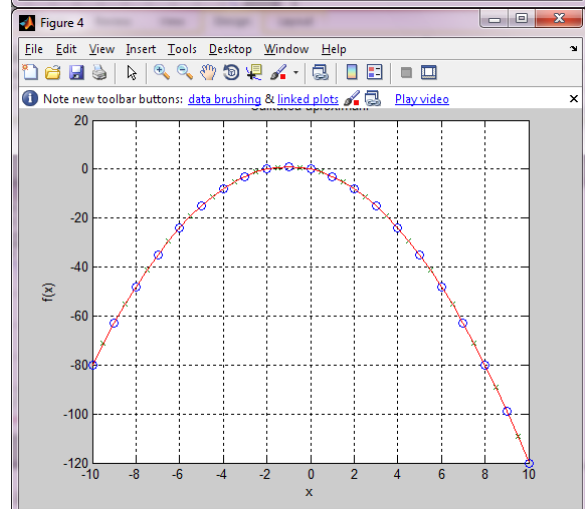
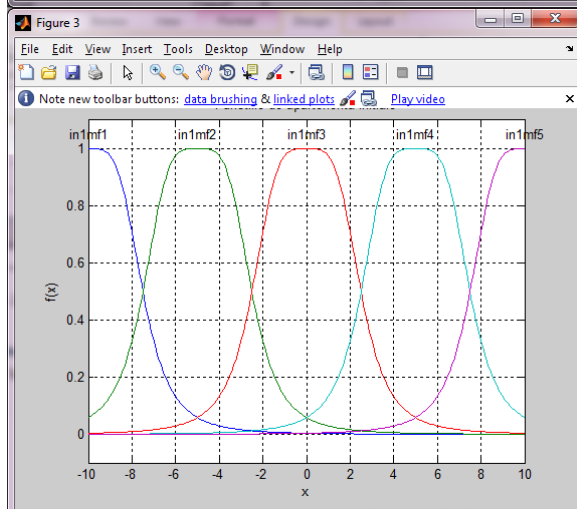
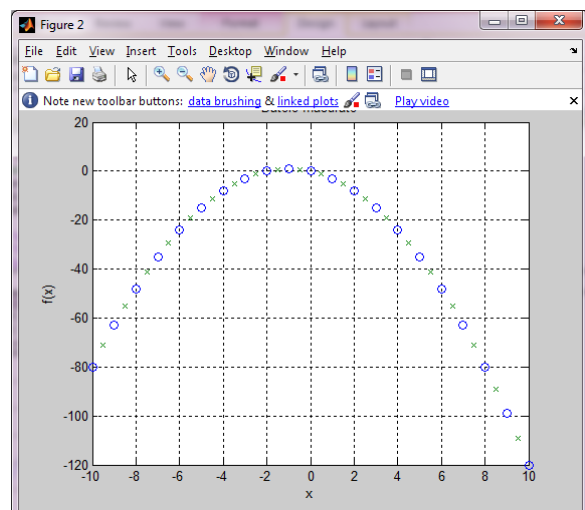
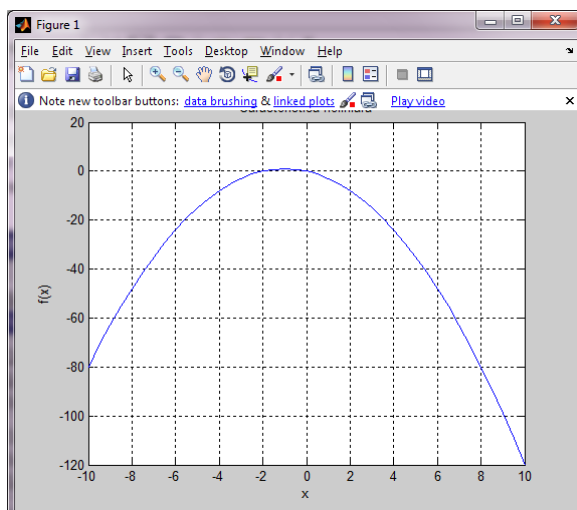
Step size decreases to 0.059942 after epoch 95.

96	0.0152275	0.0222665
97	0.00610364	0.0163279
98	0.0147314	0.0225527
99	0.00645138	0.0170143

Step size decreases to 0.053948 after epoch 99.

100	0.0144048	0.0229302
-----	-----------	-----------

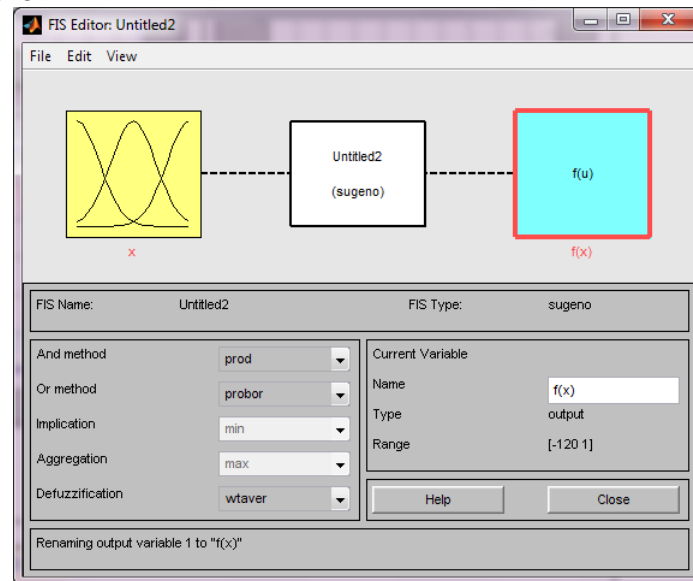
Designated epoch number reached --> ANFIS training completed at epoch 100.



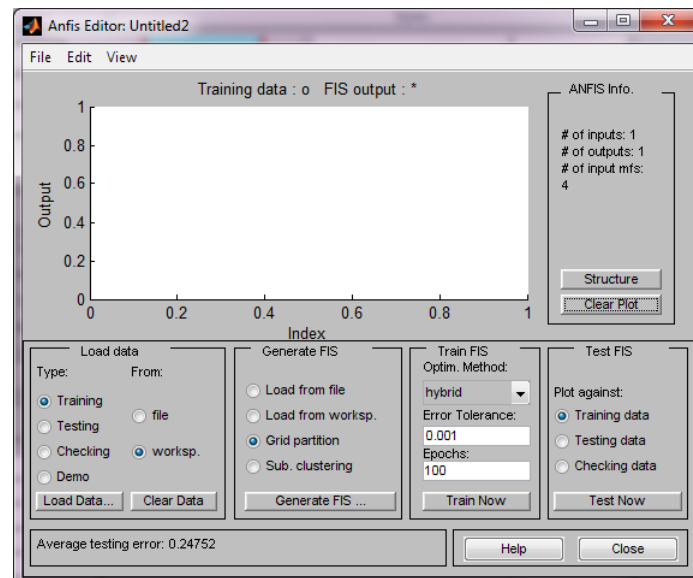
Metoda 2.

>> fuzzy

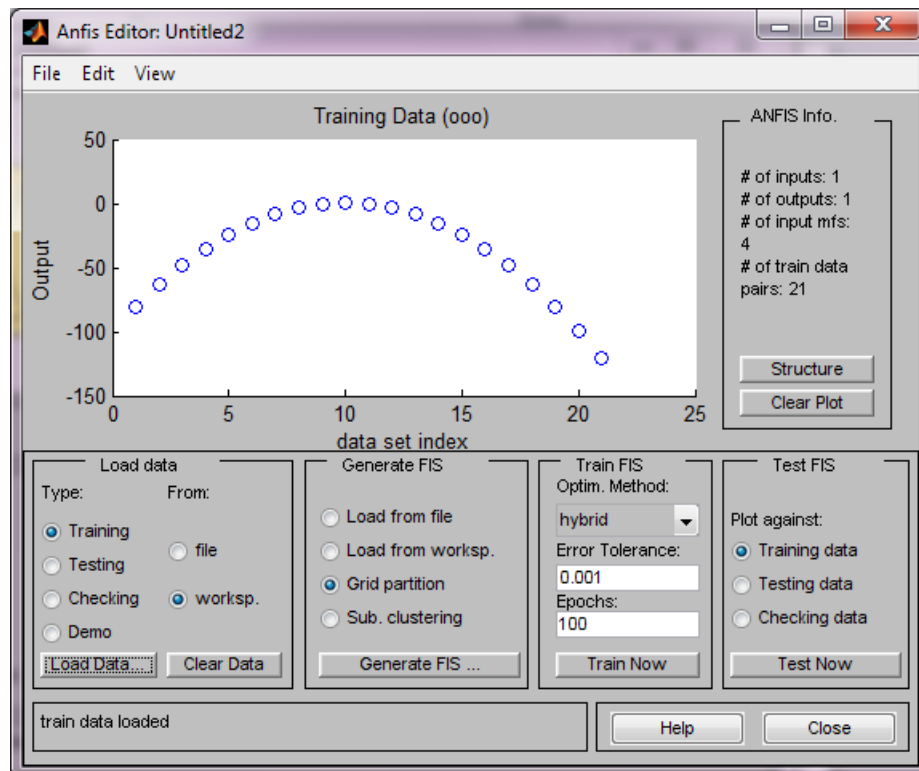
File > New FIS > Sugeno



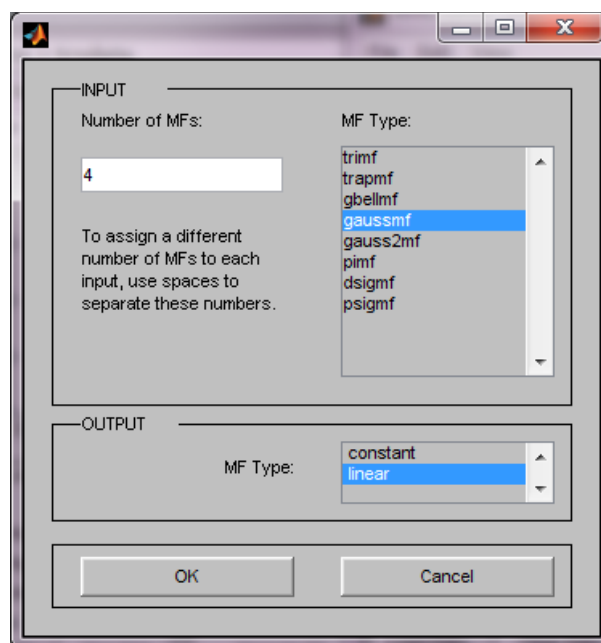
Edit > Anfis

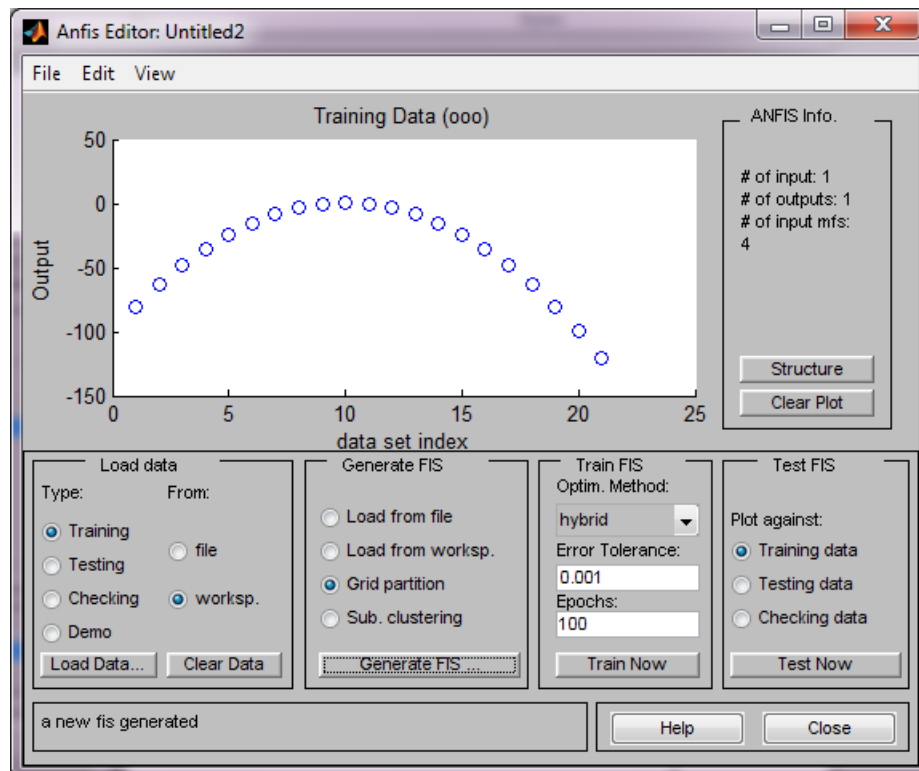


Load data > trndata

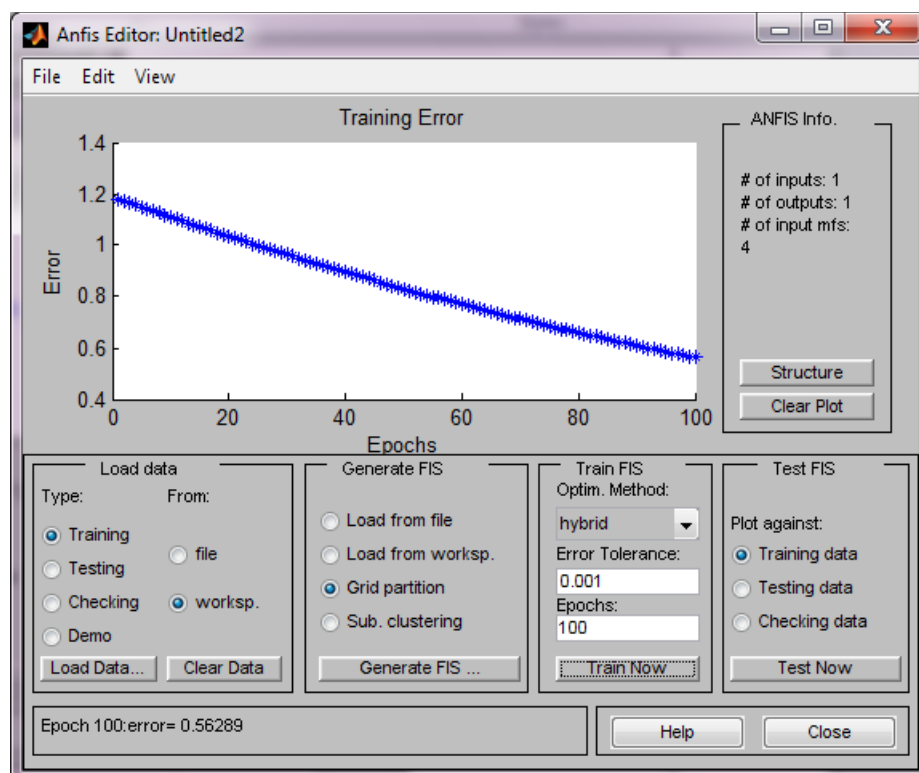


Generate FIS





Train now



Test now

