

8. Logica fuzzy

Funcționarea sistemelor fuzzy au la bază logica fuzzy.

Logica fuzzy “spune” că toate lucrurile, faptele sunt caracterizate în grade de relativitate. Kosko[8.1] afirmă că acest tip de logică “mecanizează” o mare parte din “filosofia populară”. Logica fuzzy transformă logica și matematica alb-negru (0-1, adevărat-fals) la un număr limitat de cazuri de legături “gri” între elementele relaționale. Matematica de tip fuzzy înseamnă valoarea multiplă sau multivalență. Cu alte cuvinte, abordarea cu trei valențe fuzzy a unei afirmații poate corespunde noțiunilor de adevăr, fals sau nedeterminare. Multivalorile fuzzy corespund gradelor de nedeterminare sau ambiguitate, de realizare parțială a unor evenimente sau relații. Legat de aceste noțiuni să folosim noțiunea de *paradox bivalent* pornind de la un exemplu clasic:

Un mincinos din Creta afirmă că toți locuitorii Cretei sunt mincinoși; dacă el minte înseamnă că spune adevărul. Dacă acesta spune adevărul înseamnă că el minte.

Paradoxul se poate reprezenta matematic dacă pornim de la valoarea de adevăr a afirmației S ce este obiectul paradoxului și a negației ei NS care are aceeași valoare de adevăr. Deci ecuația matematică care caracterizează acest paradox este următoarea:

$$t(S) = t(NS)$$

unde $t(*)$ este valoarea de adevăr a afirmației.

Ecuația de mai sus “violetă” legea necontradicției ce activează în cadrul logicii clasice de tip “alb-negru”. În mod normal ar trebui să avem următoarea ecuație care respectă această logică clasică:

$$t(NS) = 1 - t(S)$$

Această ecuație poate să ducă la relația:

$$2 * t(S) = 1 \rightarrow t(S) = 1/2$$

Am putea spune că un paradox se reduce la jumătăți de adevăr.

8.1. Mulțimile clasice

Înainte de a aborda teoria mulțimilor fuzzy, să facem o scurtă incursiune în teoria clasică a mulțimilor. O mulțime clasică poate fi finită, numărabilă sau nenumărabilă. Definirea ei se poate face în următoarele moduri:

- Prin enumerare individuală;
de exemplu: mulțimea fetelor din clasă ale caror nume începe cu M

$$MF=\{\text{Maria, Magdalena, Marcela}\}$$

- Enunțând proprietatea ce face ca elementele să aparțină mulțimii respective

$$T=\{x \in \mathbb{Z} \mid x \geq 0\}$$

În ultimul exemplu, delimitarea elementelor ce aparțin lui T se poate face și astfel:

$$T=\{x \mid P(x)\} \text{ unde } P(x)=x \in \mathbb{Z} \& x \geq 0$$

P(x) purtând numele de *predicat al mulțimii T*.

- O a treia metodă de definire a mulțimilor clasice este și cea care va face legătura cu mulțimile fuzzy.

Definiție: $\mu_A: X \rightarrow \{0,1\}$ este funcția caracteristică a mulțimii A dacă pentru toate elementele x:

$$\mu_A = \begin{cases} 1 & \text{dacă } x \in A \\ 0 & \text{dacă } x \notin A \end{cases}$$

Diferențele care există între varianta definirii unei mulțimi prin predicat și cea folosind funcția caracteristică sunt mici. Valoarea adevărată a lui P(x) este echivalentă cu $\mu(x)=1$. Folosind funcția caracteristică se pot găsi definiții alternative ale variantelor predicat pentru operațiile cu mulțimi cum ar fi complementarea, intersecția sau reuniunea.

Operație	Forma cu predicat	Forma cu funcție indicator
Complement al mulțimii A	$A'=\{x \mid x \notin A\}$	$\mu_{A'}(x) = 1 - \mu_A(x)$
Intersecția dintre A și B	$A \cap B = \{x \mid x \in A \& x \in B\}$	$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x))$
Reuniunea dintre A și B	$A \cup B = \{x \mid x \in A \parallel x \in B\}$	$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x))$

Avantajul utilizării funcției indicator este dat de faptul că aceste operațiuni pot fi definite în diferite variante. De exemplu, operațiunea de intersecție mai poate fi exprimată și în următoarea formă:

$$\mu_{A \cap B}(x) = \begin{cases} \mu_A(x) * \mu_B(x) \\ \text{sau} \\ \max(0, \mu_A(x) + \mu_B(x) - 1) \end{cases}$$

8.2. Mulțimi fuzzy

În anul 1965, în cadrul articolului “Mulțimi Fuzzy” Lofti Zadeh extinde sensul funcției indicator bivalente din cadrul teoriei clasice a mulțimilor la o funcție numită de apartenență (membership

function) ce are valori multiple. În cadrul acestei lucrări Zadeh introduce teoria mulțimilor cu valori multiple. Primele abordări legate de teoria mulțimilor multivaloare au fost făcute în anii '30 de polonezul Ian Lukasiewicz care a introdus sistemul logic cu 3 valori – $\{0, \frac{1}{2}, 1\}$ - extinzându-l apoi la intervalul $[0,1]$ și filosoful Max Black care a anticipat teoria mulțimilor multivaloare.

Pornind de la teoria clasică, considerăm în universul general U (numit și universul de discurs) o mulțime A . Orice element u din universul de discurs U poate să aparțină sau nu mulțimii A . Mulțimea A este definită de o funcție caracteristică $\mu_A: U \rightarrow \{0,1\}$. Dar pentru o mulțime fuzzy nu se poate spune cu certitudine că elementul u este sau nu inclus în F (am notat o mulțime fuzzy). Definirea mulțimii fuzzy se realizează prin intermediul funcției de apartenență μ_F .

Definiție: funcția de apartenență μ_F a mulțimii fuzzy F este o funcție:

$$\mu_F: U \rightarrow [0,1]$$

Enumerarea elementelor unei mulțimi fuzzy se face sub formă de perechi, o pereche fiind constituită din elementul mulțimii și valoarea corespunzătoare a funcției de apartenență:

$$F = \{ (u, \mu_F(u)) \mid u \in U \}$$

Pentru exemplificare să considerăm o mulțime fuzzy M a numerelor ce sunt apropiate de cifra 6 și care este definită de funcția de apartenență:

$$\mu_M(u) = \frac{1}{1 + (u - 6)^2}$$

În această situație universul de discurs este mulțimea numerelor naturale N , iar elementele lui M ar fi:

$$M = (3, 0.1) + (4, 0.2) + (5, 0.5) + (6, 1) + (7, 0.5) + (8, 0.2) + (9, 0.1)$$

Această metodă de enumerare a elementelor folosind semnul “+” a fost introdusă de Zadeh.

Un alt mod de reprezentare al aceleași mulțimi ar fi următorul:

$$M = \{ 0.1/3, 0.2/4, 0.5/5, 1/6, 0.5/7, 0.2/8, 0.1/9 \}$$

Generalizând modul de reprezentare al unei mulțimi, avem următoarele două situații:

- universul de discurs U este discret:

$$F = \sum_{u \in U} \frac{\mu_F(u)}{u}$$

- universul de discurs U este nemumarabil sau continuu:

$$F = \int_U \frac{\mu_F(u)}{u}$$

În cazul mulțimilor continue sau a celor numărabile nu se poate recurge la o enumerare ca cele prezentate anterior pentru a descrie o mulțime fuzzy și atunci se poate apela la reprezentarea grafică. Astfel, dacă considerăm că mulțimea M prezentată anterior este definită în universul numerelor reale R, avem o reprezentare a lui M ca în figura 8.1 .

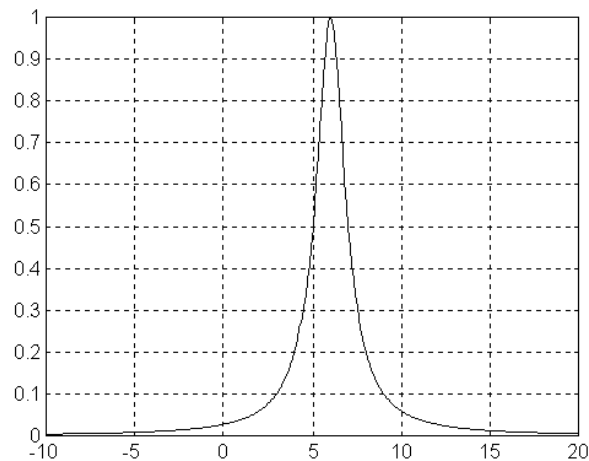


Fig. 8.1 – Funcția M

Pe același univers general se pot defini mai multe mulțimi fuzzy legate de diferite proprietăți ale aceluiasi obiect sau aceleași trăsături (caracteristici). De exemplu, pentru caracteristica temperatură, se poate defini o mulțime fuzzy pentru rece, una pentru cald, una pentru confortabil sau una pentru fierbinte; pentru caracteristica înălțime se poate defini o mulțime fuzzy pentru scund, una pentru înalt, etc.

Exemplu:

Pe domeniul vârstei se definesc 5 mulțimi fuzzy: mulțimea fuzzy a celor care sunt copii, mulțimea fuzzy a celor care sunt adolescenți, mulțimea fuzzy a celor care sunt tineri, mulțimea fuzzy a celor care sunt maturi, mulțimea fuzzy a celor care sunt bătrâni.

Reprezentarea grafică a celor 5 mulțimi fuzzy arată în felul următor:

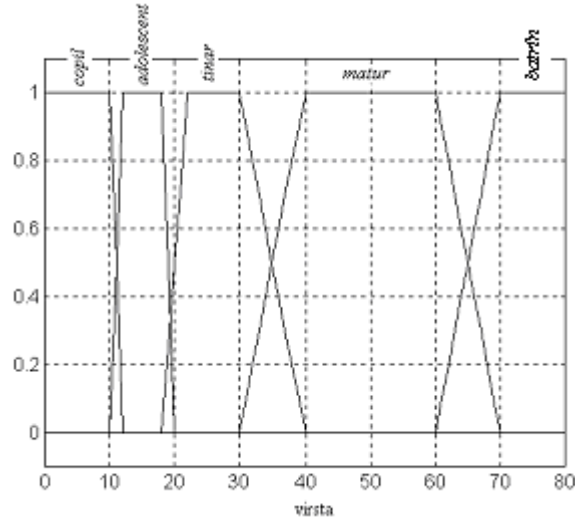


Fig. 8.2 – Reprezentarea funcțiilor fuzzy pe domeniul vârstei

O persoană în vârstă de 25 de ani este considerată tânără deoarece funcția de apartenență pentru mulțimea tânăr este de 1, iar pe toate celelalte este 0. Pentru un individ la 34 de ani, funcția de apartenență la mulțimea tânăr este 0.6, iar la cea de matur este de 0.4, în rest fiind 0. Se poate spune – conform acestei reprezentări – că persoana mai este încă tânără, dar se apropie de maturitate.

Pe grafic se pot remarca 3 tipuri de reprezentare a mulțimilor fuzzy. În general, sunt folosite 4 tipuri de grafice ce sunt prezentate în continuare[8.2]:

$$\Gamma(u; \alpha, \beta) = \begin{cases} 0 & u < \alpha \\ \frac{u - \alpha}{\beta - \alpha} & \alpha \leq u \leq \beta \\ 1 & u > \beta \end{cases}$$

Reprezentarea grafică a acestei funcții este redată în figura 8.3:

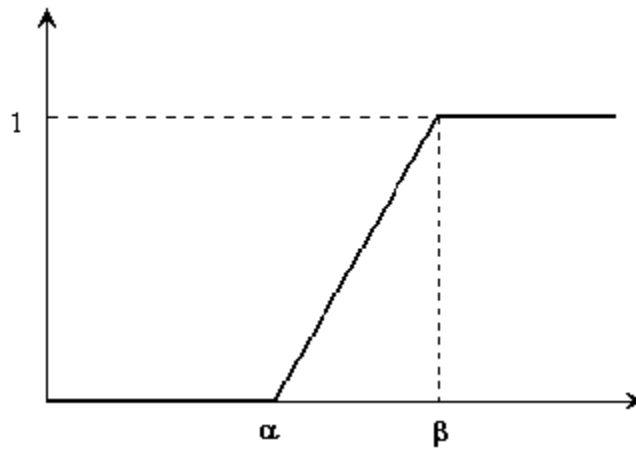


Fig. 8.3 – Funcția Γ

$$L(u; \alpha, \beta) = \begin{cases} 1 & u < \alpha \\ \frac{\beta - u}{\beta - \alpha} & \alpha \leq u \leq \beta \\ 0 & u > \beta \end{cases}$$

Reprezentarea grafică a acestei funcții este în figura 8.4:

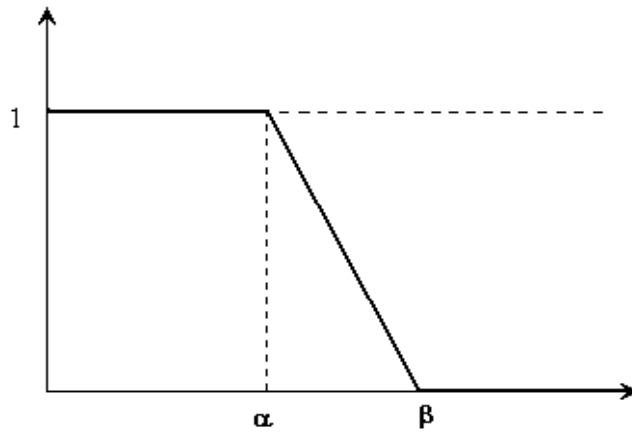


Fig. 8.4 – Funcția L

$$\Lambda(u; \alpha, \beta, \gamma) = \begin{cases} 0 & u < \alpha \\ \frac{u - \alpha}{\beta - \alpha} & \alpha \leq u \leq \beta \\ \frac{\gamma - u}{\gamma - \beta} & \beta \leq u \leq \gamma \\ 0 & u > \gamma \end{cases}$$

Reprezentarea grafică a acestei funcții este următoarea:

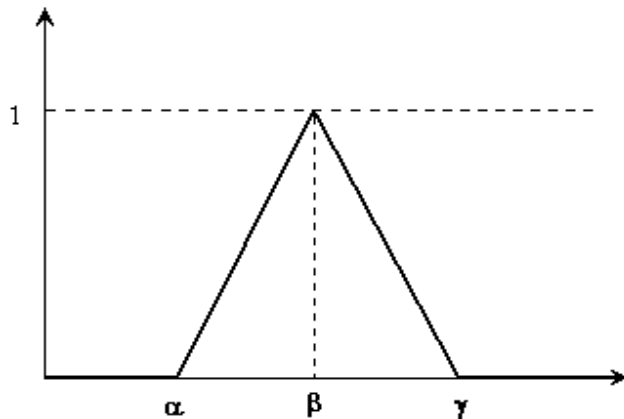


Fig. 8.5 – Funcția Λ

$$\Pi(u; \alpha, \beta, \gamma, \delta) = \begin{cases} 0 & u < \alpha \\ \frac{u - \alpha}{\beta - \alpha} & \alpha \leq u \leq \beta \\ 1 & \beta \leq u \leq \gamma \\ \frac{\delta - u}{\delta - \gamma} & \gamma \leq u \leq \delta \\ 0 & u > \delta \end{cases}$$

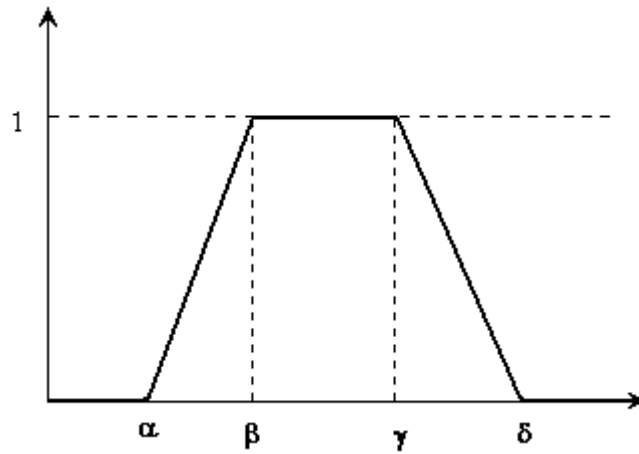


Fig. 8.6 – Funcția Π

Zadeh a definit aceleași funcții folosind puncte de inflexiune – de exemplu pentru funcția $\Gamma(u; \alpha, \beta)$:

$$S(u; \alpha, \beta, \gamma) = \begin{cases} 0 & u < \alpha \\ 2 \left(\frac{u - \alpha}{\gamma - \alpha} \right)^2 & \alpha \leq u \leq \beta \\ 1 - 2 \left(\frac{u - \gamma}{\gamma - \alpha} \right)^2 & \beta \leq u \leq \gamma \\ 1 & u > \gamma \end{cases}$$

Reprezentarea grafică a acestei funcții este conform fig. 8.7:

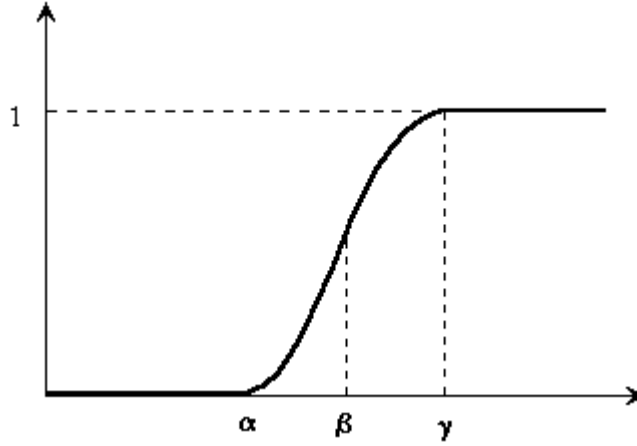


Fig. 8.7 – Funcția S definită de Zadeh

8.3. Proprietățile seturilor fuzzy

Supportul unei mulțimi fuzzy A este format din elementele universului de discurs U ce au funcția de apartenență mai mare decât 0:

$$\mathbf{S(A)} = \{ u \in U \mid \mu_A(u) > 0 \}$$

Lățimea unei mulțimi A care este caracterizată de suportul $\mathbf{S(A)}$ este definită astfel:

$$\mathbf{width(A)} = \sup(\mathbf{S(A)}) - \inf(\mathbf{S(A)})$$

unde:

- $\sup(\mathbf{S(A)})$ poate fi definit astfel: dacă $\alpha = \sup(\mathbf{S(A)})$ atunci

$$(\forall) x \in \mathbf{S(A)}, x < \alpha, \mu_A(x) > 0$$

- $\inf(\mathbf{S(A)})$ poate fi definit astfel: dacă $\beta = \inf(\mathbf{S(A)})$ atunci

$$(\forall) x \in \mathbf{S(A)}, x > \beta, \mu_A(x) > 0$$

Nucleul unei mulțimi fuzzy A este format din toate elementele care au funcția de apartenență egală cu 1.

$$\mathbf{nucleu(A)} = \{ u \in U \mid \mu_A(u) = 1 \}$$

Înălțimea unui set fuzzy A este definită de relația:

$$hgt(\mathbf{A}) = \sup_{u \in X} \mu_A(x)$$

caz în care $\sup_{u \in X} \mu_A(u)$ are următoarea semnificație:

$$\text{daca } \alpha = \sup_{u \in X} \mu_A(u) \text{ atunci } (\forall) x \in S(A), \mu_A(x) \leq \alpha$$

Pentru exemplificarea acestor proprietăți considerăm următorul exemplu:

$$\mu_A(x) = \begin{cases} 0 & x < 1 \\ x-1 & 1 \leq x \leq 2 \\ 1 & 2 \leq x \leq 3 \\ 4-x & 3 \leq x \leq 4 \\ 0 & x > 4 \end{cases}$$

din care ne rezultă următoarele proprietăți ale seturilor fuzzy:

$$\begin{aligned} S(A) &= (1,4) \\ \text{width}(A) &= 3 \\ \text{nucleu}(A) &= [2,3] \\ \text{hgt}(A) &= 1 \end{aligned}$$

8.4. Operații cu seturile fuzzy

Două mulțimi fuzzy sunt egale, de exemplu $A = B$, dacă și numai dacă:

$$(\forall) x \in X, \mu_A(x) = \mu_B(x)$$

Mulțimea fuzzy A este inclusă în mulțimea fuzzy B ($A \subseteq B$) dacă și numai dacă:

$$(\forall) x \in X, \mu_A(x) \leq \mu_B(x)$$

Suma limitată a două mulțimi fuzzy A și B ($A \oplus B$) este definită de relația:

$$A \oplus B = \int_X \frac{\min(1, \mu_A(x) + \mu_B(x))}{x}$$

Diferența limitată a două mulțimi fuzzy A și B ($A - B$) este dată de relația:

$$A - B = \int_X \frac{\max(0, \mu_A(x) - \mu_B(x))}{x}$$

Produsul a două mulțimi fuzzy A și B ($A * B$) este:

$$A * B = \int_X \frac{\mu_A(x) * \mu_B(x)}{x}$$

Puterea unei mulțimi fuzzy A ($\alpha \in \mathbb{R}$) se definește prin:

$$A^\alpha = \int_X \frac{(\mu_A(x))^\alpha}{x}$$

8.5. Operații cu relații fuzzy

Considerăm două relații R și S definite pe $X \times Y$ și putem defini următoarele operații cu relații fuzzy:

Intersecția dintre două relații fuzzy este dată prin următoarea formulă:

$$(\forall)(x, y) \in X \times Y : \mu_{R \cap S}(x, y) = \min(\mu_R(x, y), \mu_S(x, y))$$

Reuniunea dintre două relații fuzzy R și S este definită în următoarea formă:

$$(\forall) (x, y) \in X \times Y : \mu_{R \cup S}(x, y) = \max(\mu_R(x, y), \mu_S(x, y))$$

Exemplu: fie R , respectiv S , relațiile definite între mulțimile X și Y și care sunt prezentate în tabelele de mai jos:

xRy

	y_1	y_2	y_3	y_4
x_1	0.1	0	0.5	0.6
x_2	0.8	1	0.9	0.7
x_3	0.5	0.4	0.3	0.6
x_4	0.1	0.2	0.8	0.3

Tabel 8.1

xSy

	y_1	y_2	y_3	y_4
x_1	0.8	0.1	0.9	0.7
x_2	0.9	0.8	0.5	0.7
x_3	0	1	0.7	0.2
x_4	0.3	0	0.8	0.8

Tabel 8.2

$X \cap Y$

	y_1	y_2	y_3	y_4
x_1	0.1	0	0.5	0.6
x_2	0.8	0.8	0.5	0.7
x_3	0	0.4	0.3	0.2
x_4	0.1	0	0.8	0.3

Tabel 8.3

$X \cup Y$

	y_1	y_2	y_3	y_4
x_1	0.8	0.1	0.9	0.7
x_2	0.9	1	0.9	0.7
x_3	0.5	1	0.7	0.6
x_4	0.3	0.2	0.8	0.8

Tabel 8.4

Definiție: Proiecția relației R pe mulțimea fuzzy Y este:

$$proj R / Y = \int_Y \sup \left(\frac{\mu_R(x, y)}{y} \right)$$

Exemplu:

$$proj R / X = 0.6/x_1 + 1/x_2 + 0.6/x_3 + 0.8/x_4$$

$$proj R / Y = 0.8/y_1 + 1/y_2 + 0.9/y_3 + 0.7/y_4$$

Operația de proiecție este, în general, folosită pentru a obține extensia cilindrică.

$Ce(P_x)$

	y_1	y_2	y_3	y_4
x_1	0.6	0.6	0.6	0.6
x_2	1	1	1	1
x_3	0.6	0.6	0.6	0.6
x_4	0.8	0.8	0.8	0.8

Tabel 8.5

$Ce(P_y)$

	y_1	y_2	y_3	y_4
x_1	0.8	1	0.9	0.7
x_2	0.8	1	0.9	0.7
x_3	0.8	1	0.9	0.7
x_4	0.8	1	0.9	0.7

Tabel 8.6

8.6. Raționarea aproximativă

Raționarea aproximativă este una dintre cele mai folosite forme de fuzzy logic și acoperă o varietate de reguli de inferență a căror premise conțin propoziții fuzzy.

Pentru a putea lucra cu mecanismele gândirii aproximative este necesar să fie introduse elementele de bază ale acestora: variabilele lingvistice și prozițiile fuzzy.

8.6.1. Variabile lingvistice

Zadeh definește variabilele lingvistice astfel: “prin variabilă lingvistică înțelegem o variabilă a cărei valoare este reprezentată de cuvinte sau propoziții într-un limbaj natural sau artificial”. De exemplu, **vârsta** este o variabilă lingvistică dacă variabilele atașate ei sunt mai degrabă lingvistice decât numerice, și anume: tânăr, foarte tânăr, bătrân, foarte bătrân, nu foarte tânăr, matur, etc. față de 20, 30, 40, ... de ani.

În cadrul raționării aproximative există următoarea structură asociată noțiunii de variabilă lingvistică:

$$\langle X, LX, X, M_x \rangle$$

unde:

- X este numele simbolic al variabilei lingvistice: vârstă, înălțime, temperatură, eroare, variația erorii;
- LX este setul de valori lingvistice pe care-l poate lua X ;
Exemplu:
 - pentru înălțime: $L\hat{I} = \{\text{pitic, scund, mediu, nu foarte înalt, înalt, uriaș}\}$
- X este domeniul de valori clasice în care variază variabila lingvistică;
Exemplu:
 - $\hat{I}\text{ înălțime} = [20\text{cm}, 300\text{cm}]$
 - $\text{Temperatură} = [-10^\circ\text{C}, 30^\circ\text{C}]$
- M_x este funcția care face legătura între valorile lingvistice fuzzy cu valorile cantitative.

$$M_x : LX \rightarrow LX \square$$

unde $LX \square$ este funcția ce definește fiecare valoare lingvistică.

În cazul exemplului din paragraful 8.2, variabila adolescent are ca valoare a funcției ce definește o valoare lingvistică $LX \square$ funcția $\Pi(x, 18, 22, 30, 40)$, iar variabila bătrân are funcția $LX \square = \Gamma(x, 60, 70)$.

Referitor la simbolurile variabilei lingvistice, în general, prin eroare se înțelege, în cadrul unui proces, diferența dintre mărimea de referință și mărimea achiziționată, iar cele mai uzitate reprezentări de valori lingvistice pentru eroare sunt:

$$\langle \text{NB, NM, NS, ZO, PS, PM, PB} \rangle$$

aceste componente având semnificația următoare:

NB – negative big;
 NM – negative medium;
 NS – negative small;
 ZE – zero;
 PS – positive small;
 PM – positive medium;
 PB – positive big.

Programele informatice folosite pentru implementarea sistemelor expert își bazează funcționarea pe reguli care, la rândul lor, sunt compuse din propoziții. În cazul logicii fuzzy aceste propoziții au următoarea formă:

E este NB

Variabila *E* are valoarea negativ mare NB. Implicația acestei propoziții este că pentru valoarea *e* a lui *E* rezultă valoarea $\mu_{NB}(e)$ a mulțimii NB.

Exemplu: dacă negative big (NB) este chiar mulțimea *A*, definită în exemplul de la proprietățile mulțimilor fuzzy, iar $val(e) = 1.5$, atunci $\mu_{NB}(e) = 0.5$.

Pornind de la noțiunea de propoziție fuzzy și elementele de conexiune lingvistică (și, sau, nu) se pot construi fraze fuzzy.

Fie două propoziții:

$p : X \text{ este NB}$

$q : Y \text{ este PS}$

Relația de conjuncție dintre aceste două propoziții este următoarea:

$r : p \text{ și } q$

aceasta fiind reprezentată prin funcția fuzzy:

$$\mu_r(x, y) = \int_{X \times Y} \frac{\min(\mu_{NB}(x), \mu_{PS}(y))}{(x, y)}$$

Dacă aceleași propoziții se află în relația de disjuncție, atunci vom avea relația:

$r : p \text{ sau } q$

Reprezentarea matematică a acestei operații prin funcție fuzzy este următoarea:

$$\mu_r(x, y) = \int_{X \times Y} \frac{\max(\mu_{NB}(x), \mu_{PS}(y))}{(x, y)}$$

iar pentru operația de negație avem următoarea formulă:

$$r : NOT(p)$$

Operația de negație prin funcție fuzzy este implementată prin expresia:

$$\mu_r(x) = \int_x \frac{\max(0, 1 - \mu_{NB}(x))}{x}$$

Folosind propozițiile fuzzy în cadrul structurilor de tipul *dacă...atunci* (*if...then*) duce la crearea de reguli ce stau la baza controllerelor fuzzy:

$$dacă \langle \text{propoziție fuzzy} \rangle \text{ atunci } \langle \text{propoziție fuzzy} \rangle$$

O exemplificare pentru această structură este fraza folosită în controlul proceselor:

$$dacă \dot{e} \text{ este } NB \text{ și } \dot{e} \text{ este } PB \text{ atunci } \dot{u} \text{ este } NS$$

unde e este eroarea, \dot{e} este variația erorii, iar \dot{u} este variația semnalului de comandă (evident, în loc de NB, PB și NS se poate afla oricare valoare din setul de valori lingvistice). Altfel spus, la o schimbare negativ mare a erorii și pozitiv mare a variației erorii, semnalul de comandă trebuie modificat negativ mic.

Prima parte a frazei fuzzy este descrisă de valoarea funcției de apartenență rezultantă a relației ce caracterizează premisa. Pornind de la această valoare se poate determina structura funcției ce reprezintă concluzia frazei, și astfel, se stabilește variația semnalului de comandă. Determinarea structurii funcției concluzie se poate face prin mai multe metode numite și implicații fuzzy, iar dintre acestea ar fi de menționat: Mandami, Kleene-Dienes, Zadeh, Lukasiewicz, Gödel.

Cea mai folosită în controlul fuzzy este implicația Mandami, conform căreia funcția rezultantă arată astfel:

$$\mu_{CNS}(\dot{u}) = \min \left(\left(\min \frac{\mu_{NB}(e), \mu_{PB}(\dot{e})}{(e, \dot{e})} \right), \mu_{NS}(\dot{u}) \right)$$

Practic, din mulțimea fuzzy NS ce caracterizează concluzia se decupează un trapez de înălțime:

$$\min \left(\frac{\mu_{NB}(e), \mu_{PB}(\dot{e})}{(e, \dot{e})} \right)$$

Indicele C alături de NS provine din cuvântul “clipped” ce semnifică “a tăia” (“a decupa”).

Pentru o mai bună înțelegere, considerăm propoziția:

dacă e este NB atunci u este PS

De exemplu, dacă e ia valoarea e^* și îi va corespunde $\mu_{NB}(e^*) = a$ (fig.8.8) atunci structura funcției de apartenență pentru concluzie este:

$$\mu_{CPS}(u) = \min(a, \mu_{PS}(u))$$

Reprezentarea grafică a trapezului rezultat în urma aplicării funcției de apartenență este redată în fig.8.8.:

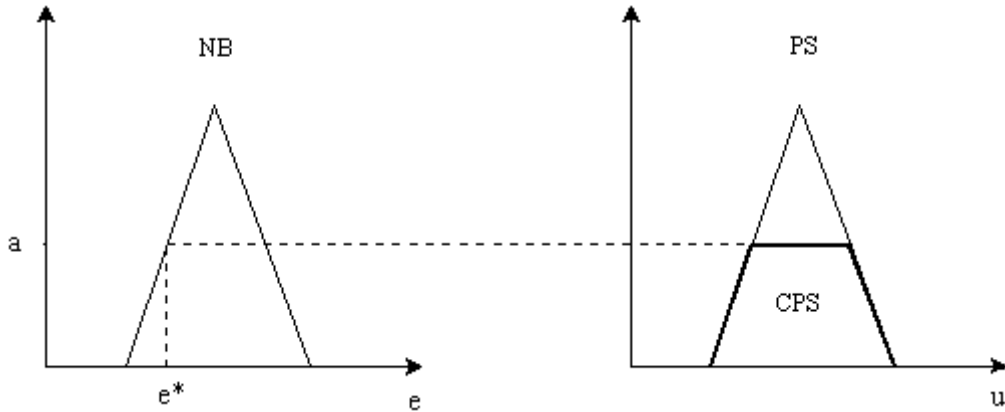


Fig. 8.8

Astfel, $\max(\mu_{CPS}(u)) = a, \forall u$.

8.7. Bibliografie

- [8.1] B. Kosko - Neural Networks and Fuzzy Systems, Prentice Hall International, 1993.
- [8.2] D. Driankov, H. Hellendroon, M. Reinfrank - An introduction to fuzzy control, Springer Verlag, New York, 1996.

9. Controlere fuzzy

În acest capitol se vor descrie principalele componente ale unui controler fuzzy și va cuprinde prezentarea factorilor implicați, a modulelor de fuzificare și a modulelor de defuzificare. Pe lângă aceste prezentări se va discuta și despre relevanța diferiților parametri incluși în controlerul fuzzy. Alegerea unei funcții de apartenență, metodele de fuzificare, motorul de inferență și forma și înțelegerea (cunoașterea) regulilor de bază se vor descrie pe scurt în acest capitol. Se va discuta în detaliu despre metodele de defuzificare cu avantajele și dezavantajele aplicării fiecăreia în parte într-un caz practic care va fi prezentat într-un capitol următor.

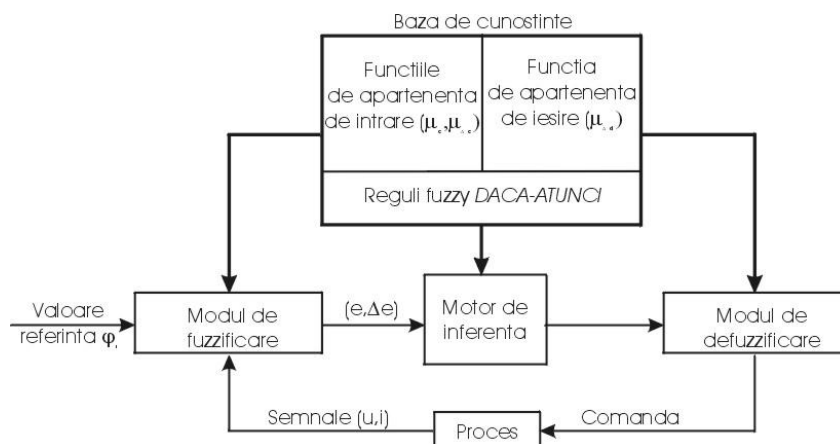


Fig. 9.1 – Structura de bază a unui controler fuzzy

Componentele principale ale unui controller fuzzy sunt[8.2]: modulul de fuzificare, baza de cunoștințe alcătuită din două module separate: baza de date și baza de reguli, motorul de inferență și modulul de defuzificare. În continuare se va descrie fiecare componentă a controllerului.

9.1. Modulul de fuzificare

Modulul de fuzificare (FM) realizează așa numita operație de fuzificare care convertește valoarea curentă a unei variabile a procesului într-o componentă fuzzy (mulțime fuzzy) care va fi folosită în cadrul regulilor de inferență ale controlerului. Modulul de fuzificare are rolul de a adapta semnalele de intrare ce provin de la proces în limitele universului de discurs pe care sunt definite mulțimile fuzzy precum și de a le transforma din mărimi clasice în mărimi fuzzy.

Reprezentarea parametrilor din modulul de fuzificare se face prin alegerea unei strategii corespunzătoare de fuzificare care depinde de natura procesului controlat și a mărimii de intrare.

9.2. Baza de cunoștințe

Baza de cunoștințe este nucleul de date și reguli ale unui controller fuzzy, aceasta fiind constituită dintr-un modul numit “baza de date” și un modul numit “baza de reguli”.

Principala funcție a bazei de date este aceea de a furniza informația necesară pentru funcționarea corespunzătoare a modului de fuzificare, a bazei de reguli și a modului de defuzificare. Baza de date conține funcțiile de apartenență pentru fiecare variabilă fuzzy, definite pe valori lingvistice (de exemplu dacă avem valorile {NS, ZE respectiv PS} atunci avem definită câte o funcție de apartenență $\mu(u)$ pentru NS, ZE respectiv PS).

În cele mai multe cazuri funcțiile de apartenență – fără ca acest lucru să fie regulă – au înălțimea egală cu 1, formă simetrică, iar punctul de intersecție dintre două mulțimi adiacente (de exemplu NS și ZE) este de 0.5. Cele mai folosite forme de funcții de apartenență sunt cele prezentate la capitolul precedent în paragraful 8.2, adică forma trapezoidală, forma triunghiulară sau forma tip clopot (formă introdusă de Zadeh). În funcție de natura cunoștințelor expertului și a modului de variație a acestor funcții de apartenență, se pot folosi funcții asimetrice, de înălțimi diferite, sau intersecția dintre două mulțimi adiacente se realizează la altă valoare, diferită de 0.5.

În modulul “baza de reguli” sunt definite variabilele de stare ce provin de la proces (de obicei definite ca eroarea mărimii respective), iar variabila (variabilele) de ieșire din baza de reguli se constituie în semnal de comandă pentru proces. Tot în acest modul sunt specificate valorile lingvistice atașate fiecărei variabile fuzzy în parte, precum și numărul de reguli anterioare și rezultante, ca de exemplu:

dacă e este NB și \dot{e} este PS atunci \dot{u} este NS

care se poate “traduce” prin: dacă eroarea mărimii y ($e = y - y_r$, unde y_r este mărimea de referință a semnalului) este NB și variația erorii \dot{e} ($\dot{e} = e(k) - e(k-1)$, unde $e(k)$ este eroarea la pasul k , iar $e(k-1)$ este eroarea la pasul precedent $k-1$) este PS, atunci variația semnalului de comandă \dot{u} este NS.

9.3. Motorul de inferență

Motorul de inferență calculează valoarea globală a variabilelor de ieșire. Acest calcul se bazează pe contribuția fiecărei reguli din modulul “baza de reguli”. Fiecare asemenea contribuție individuală reprezintă valorile variabilelor de ieșire calculate după o singură regulă.

Ieșirea modului de fuzificare, reprezentând regula curentă, este compusă din cele mai “tari” valori (crisp values) ale variabilelor de stare ale procesului. Această regulă (curentă) este corespunzătoare fiecărei reguli anterioare, iar gradul de corespondență al fiecărei reguli cu fiecare este stabilit în motorul de inferență.

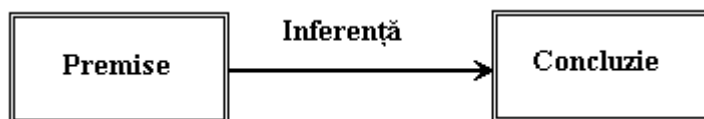


Fig. 9.2 – Procesul de inferență

Regulile de inferență sunt procedee prin care se determină consecințele unor anumite presupuneri ce se fac despre formule sau enunțuri într-o teorie. Inferența este operația prin care pe baza

acceptării unui număr finit de enunțuri denumite premise, se trece la acceptarea unui alt enunț numit concluzie. Într-o inferență premisele formează condiția suficientă a concluziei, concluzie ce poate rezulta din alte premise. Prin inferență, valoarea de adevăr a premisei se transmite concluziei.

În cadrul desfășurării procesului este posibil ca o pereche de variabile de intrare să activeze mai mult de o propoziție fuzzy, ceea ce produce un număr egal de funcții rezultante. De exemplu:

dacă e este NB și \dot{e} este PS atunci \dot{u} este NS

dar în același timp și

dacă e este PB și \dot{e} este PS atunci \dot{u} este NB

În urma aplicării, regulilor de inferență pot rezulta un număr de fraze selectate ce este egal cu 2^i , unde $i = 0 \div n$, unde n este numărul de variabile de intrare.

9.4. Modulul de defuzificare

Modulul de defuzificare este cel care stabilește variația semnalului de ieșire plecând de la rezultatele produse de regulile de inferență. Prin defuzificarea funcțiilor rezultante (am putea spune prin medierea lor) rezultă variația semnalului de ieșire.

Reprezentarea parametrului din modulul de defuzificare se realizează prin diferite metode de defuzificare. În funcție de aplicația (procesul) utilizată vom folosi o anumită metodă de defuzificare. Cele mai cunoscute și folosite metode de defuzificare sunt: metoda înălțimii, metoda centrului de greutate, metoda sumei, metoda primului maxim, metoda ultimului maxim și metoda mijlocul maximelor.

Înainte de a prezenta aceste metode vom considera un univers de discurs U al unei variabile u . Pe acest univers de discurs se definește mulțimea funcțiilor de apartenență corespunzătoare variabilelor atașate lui u , $U_\mu = \{ \mu_1, \mu_2, \dots, \mu_l \}$ și să notăm cu $U_{C\mu} = \{ \mu_{C1}, \mu_{C2}, \dots, \mu_{C3} \}$ mulțimea funcțiilor de apartenență rezultate ca urmare a aplicării regulilor de inferență valorilor variabilelor de intrare curente.

9.4.1. Metoda înălțimii

Defuzificarea înălțimii este o metodă care folosește mulțimea funcțiilor de apartenență rezultate ca urmare a aplicării regulilor de inferență valorilor variabilelor de intrare curente $U_{C\mu}(i)$, unde i ia valori de la '0' la 'n'. Această metodă ia în considerare valoarea de vârf a fiecărei funcții μ_{Ci} , poziția pe abscisă în care se atinge valoarea înălțimii și construiește suma ponderată a acestor valori. Metoda este foarte simplu de aplicat și totodată este o metodă rapidă.

Dacă funcțiile rezultate din aplicarea inferenței sunt de formă trapezoidală, atunci pentru fiecare se alege valoarea pe abscisă a mijlocului bazei mici a trapezului. Din acest punct se calculează valoarea înălțimii funcției de apartenență calculate.

Variabila de ieșire se calculează cu următoarea relație:

$$u = \frac{\sum_{k=1}^n c^{(k)} f_k}{\sum_{k=1}^n f_k}$$

Elementele implicate în această ecuație au următoarea semnificație:

- f_k este înălțimea funcției de apartenență;
- c_k este poziția pe abscisă a mijlocului intervalului pe care se atinge valoarea maximă pentru funcția de apartenență respectivă.

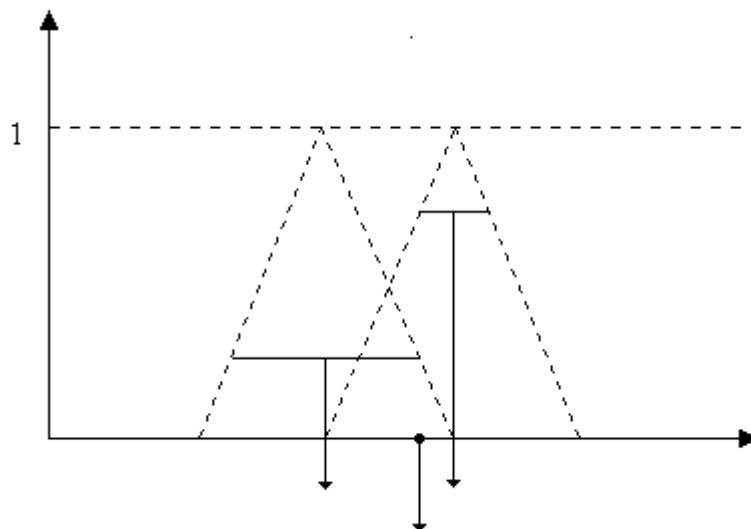


Fig. 9.3 – Metoda înălțimii

9.4.2. Metoda centrului de greutate

Metoda centrului de greutate este cea mai cunoscută metodă de defuzificare. Valoarea rezultată prin aplicarea acestei metode reprezintă poziția pe abscisă a centrului de greutate a suprafeței rezultate prin reunirea funcțiilor de apartenență produse de regulile de inferență.

În cazul în care mulțimea de discurs U este *discretă* (având n elemente), variația variabilei de ieșire este dată de relația:

$$u = \frac{\sum_{i=1}^n u_i \mu_{U_{\mu}}(u_i)}{\sum_{i=1}^n \mu_{U_{\mu}}(u_i)} = \frac{\sum_{i=1}^n u_i \max_k \mu_{U_{C_{\mu}(k)}}(u_i)}{\sum_{i=1}^n \max_k \mu_{U_{C_{\mu}(k)}}(u_i)}$$

În cazul unui univers de discurs *continuu*:

$$u = \frac{\int_U u \mu_{U_{\mu}}(u) du}{\int_U \mu_{U_{\mu}}(u) du} = \frac{\int_U u * \max_k \mu_{U_{C_{\mu}(k)}}(u) du}{\int_U \max_k \mu_{U_{C_{\mu}(k)}}(u) du}$$

unde $u_i \in U$ iar \int este operația clasică de integrare.

Când u_i este dintr-o zonă în care se intersectează (vezi fig. 9.4) două sau mai mult (mai rar) de două funcții de apartenență rezultante, se va lua în considerare valoarea cea mai mare ce apare ca urmare a aplicării funcțiilor de apartenență rezultante. Acest fapt este impus de regulile de calcul ale centrului de greutate.

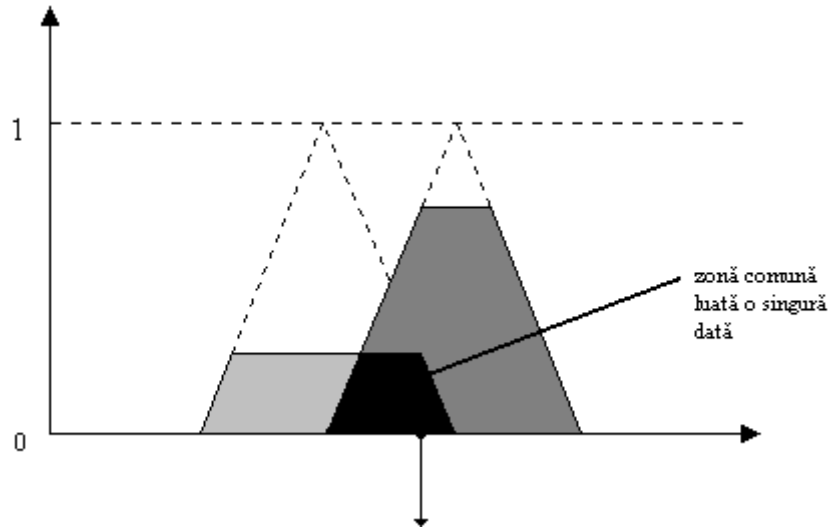


Fig. 9.4 – Metoda centrului de greutate

9.4.3. Metoda sumei

O metodă similară cu metoda centrului de greutate, dar mai rapidă, este metoda sumei. Motivul pentru care folosim această metodă este acela de a evita calculul mulțimii funcțiilor de apartenență rezultate ca urmare a aplicării regulilor de inferență valorilor variabilelor de intrare curente $U_{C_{\mu}}$. Ideea este să se considere contribuția fiecărei arii a funcțiilor de apartenență μ_{C_i} , $i = 0 \div n$, luate individual. Matematic vorbind, metoda centrului de greutate construiește $U_{C_{\mu}}$, luând reuniunea tuturor elementelor μ_{C_i} . Metoda sumei calculează suma dintre elementele μ_{C_i} .

Câteva dintre sistemele fuzzy care foloseau ca metodă de defuzificare metoda centrului de greutate, implementează acum ca metodă de defuzificare cea a sumei.

În cazul *discret*, metoda sumei este descrisă formal prin expresia:

$$u = \frac{\sum_{i=1}^n u_i \sum_{k=1}^l \mu_{U_{C\mu}(k)}(u_i)}{\sum_{i=1}^n \sum_{k=1}^l \mu_{U_{C\mu}(k)}(u_i)}$$

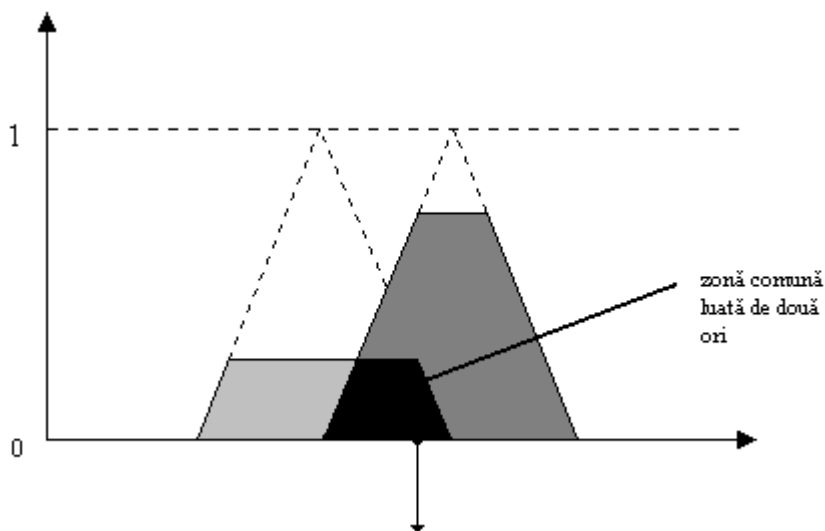


Fig. 9.5 – Metoda sumei

În cazul *continuu*, metoda sumei este descrisă prin următoarea expresie:

$$u = \frac{\int_U u * \sum_{k=1}^l \mu_{U_{C\mu}(k)}(u) du}{\int_U \sum_{k=1}^l \mu_{U_{C\mu}(k)}(u) du}$$

9.4.4. Metoda primului maxim

Metoda primului maxim folosește mulțimea funcțiilor de apartenență rezultate ca urmare a aplicării regulilor de inferență valorilor variabilelor de intrare curente $U_{C\mu}$ și ia cele mai mici valori ale domeniului U cu gradul de apartenență cel mai mare în $U_{C\mu}$.

Acest lucru este realizat formal în trei pași:

- 1) $hgt(U_{C\mu}) = \sup_{u \in U} \mu_{U_{C\mu}}(u)$ este cu cel mai mare grad de apartenență din $U_{C\mu}$;

- 2) $\{u \in U / \mu_{U_{C\mu}}(u) = hgt(U_{C\mu})\}$ este mulțimea elementelor domeniului cu gradul de apartenență egal cu $hgt(U_{C\mu})$;
- 3) variația variabilei de ieșire se calculează cu relația:

$$\dot{u} = \inf_{u \in U} \{u \in U / \mu_{U_{C\mu}}(u) = hgt(U_{C\mu})\}$$

Astfel se determină primul maxim întâlnit la parcurgerea funcțiilor de apartenență, sugestiv fiind chiar numele acestei metode.

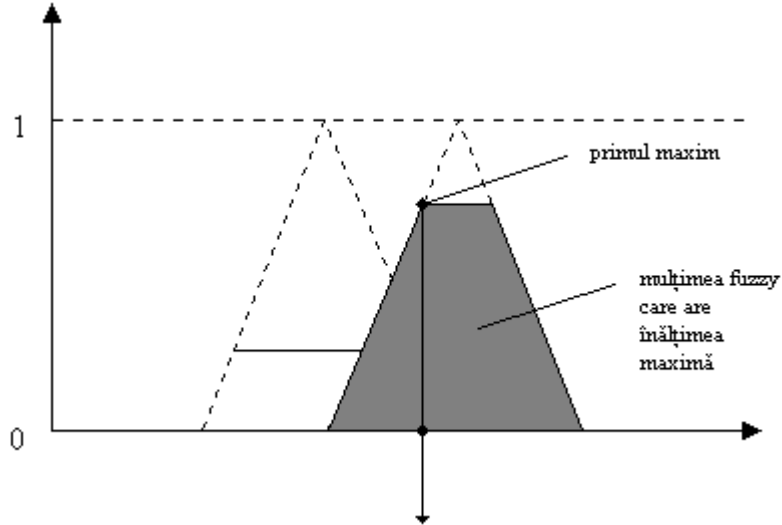


Fig. 9.6 – Metoda primului maxim

Versiunea alternativă a acestei metode se numește ultimul maxim.

9.4.5. Metoda ultimului maxim

Metoda este asemănătoare cu cea precedentă, singura diferență fiind aceea că aceasta calculează ultimul maxim întâlnit la parcurgerea funcțiilor de apartenență.

Variația variabilei de ieșire are expresia:

$$\dot{u} = \sup_{u \in U} \{u \in U / \mu_{U_{C\mu}}(u) = hgt(U_{C\mu})\}$$

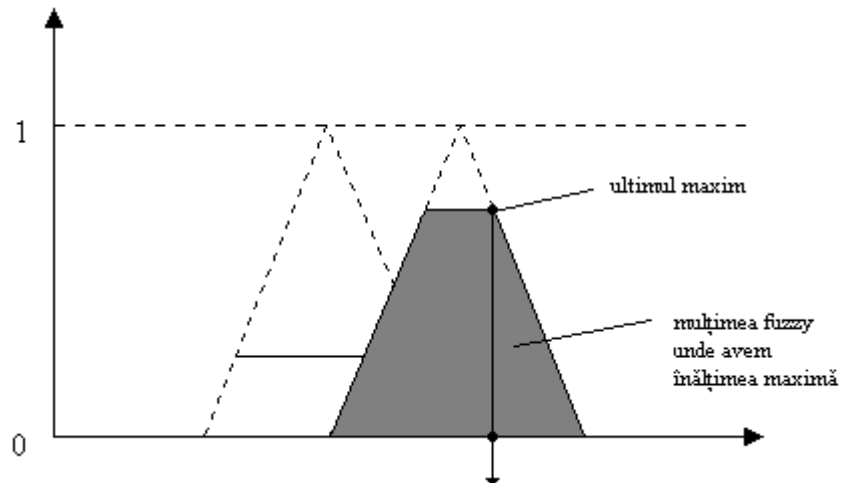


Fig. 9.7 – Metoda ultimului maxim

Termenii $\sup_{u \in U} \mu_U(u)$ și $\inf_{u \in U} \mu_U(u)$ au fost definiți în capitolul 8.

9.4.6. Metoda mijlocul maximelor

Metoda este foarte asemănătoare cu metoda primului maxim și cu metoda ultimului maxim, doar că pentru a-l determina \bar{u} , se realizează o mediere a celor două valori precedente ale variabilei de ieșire, adică valoarea de la metoda primului maxim și valoarea de la metoda ultimului maxim.

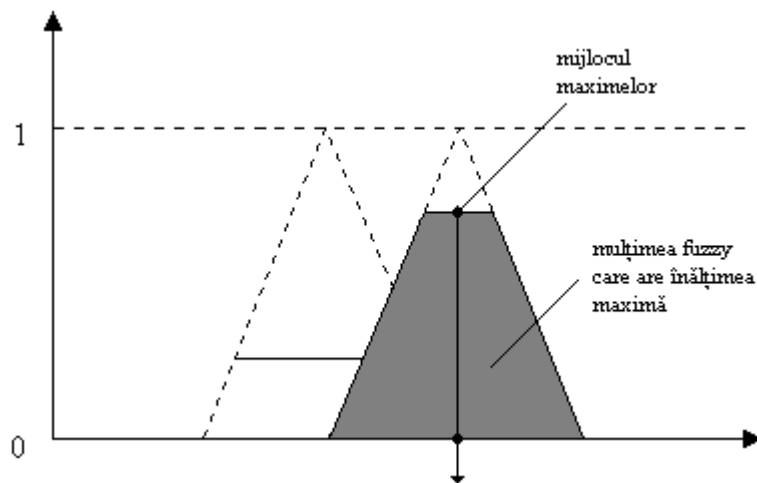


Fig. 9.8 – Metoda mijlocul maximelor

Expresia acestei variabile este:

$$u = \frac{\inf_{u \in U} \{u \in U / \mu_{U_{C\mu}}(u) = hgt(U_{C\mu})\} + \sup_{u \in U} \{u \in U / \mu_{U_{C\mu}}(u) = hgt(U_{C\mu})\}}{2}$$

9.5. Exemple de controlere fuzzy

În paragraful de urmează vor fi prezentate 2 controlere fuzzy. Primul a fost realizat practic și are rolul de controla defazajul curent tensiune într-un circuit RL al cărui parametrii nu sunt cunoscuți. Cel de la doilea, simulează parcare în marșalie a unui camion într-o zonă de lucru.

9.5.1. Controler fuzzy pentru reglarea factorului de putere.

Sistemul ce urmează a fi controlat prin tehnica fuzzy, a fost introdus în [9.1] și a fost conceput pentru a controla faza curentului în circuitul rotoric al unui motor trifazat asincron. În acest articol, sistemul este folosit pentru a regla defazajul dintre curent și tensiune într-un circuit de putere inductiv monofazat.

Acest sistem constă într-o punte H cu comutatoare bidirecționale, având plasat în mijloc un condensator de curent alternativ și permite emularea de efecte capacitive care sunt de zeci, sute de ori mai mari decât cel ce ar fi produs de condensatorul plasat în mijlocul punții în cazul unei configurații RLC simple.

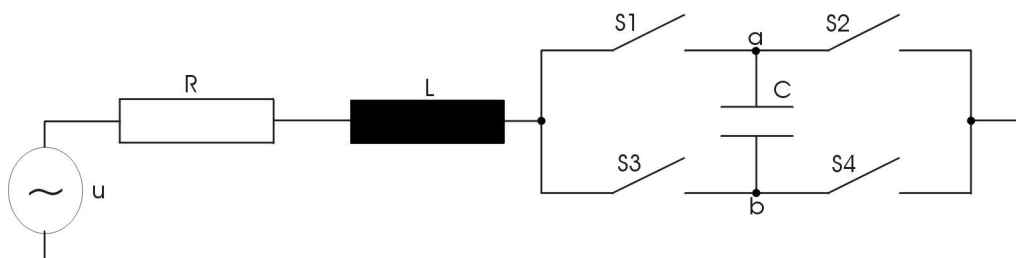


Fig. 9.9 – Circuit RL cu punte H

Perechile de comutatoare (S1,S4) și (S2,S3) sunt comutate complementar după o strategie de tip PWM. Factorul de umplere d aplicat punții pentru a obține un defazaj impus φ este dat de formula ($d=t_1/T$, unde T este perioada de comutație, t_1 este intervalul de timp când (S1,S4) conduc, t_2 este intervalul de timp când (S2,S3) conduc, $t_1+t_2=T$):

$$d = \frac{1}{2} \left(1 + \sqrt{\omega^2 LC - \omega CR \tan(\varphi)} \right)$$

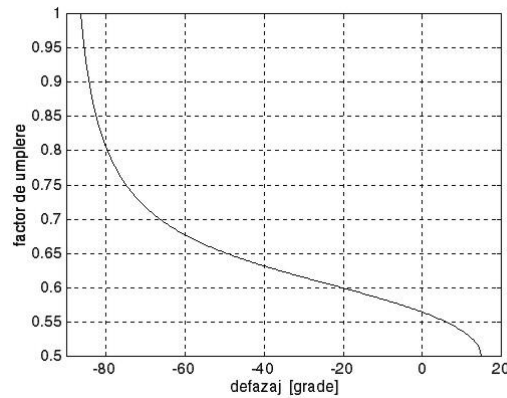


Fig. 9.10 – Factor de umplere versus defazaj impus

Valoarea lui d depinde de parametrii circuitului controlat. În cazul în care pe parcursul procesului aceștia se modifică, trebuie ca și valoarea lui d aplicată punții H să se modifice pentru ca defazajul să se păstreze egal cu cel impus. Evoluția lui d funcție de φ , indiferent de parametrii circuitului RL, este asemănătoare cu cea din fig. 9.10. La parametrii diferiți, curba va avea pante diferite. Curentul și tensiunea sunt aduse în fază la valori diferite ale lui d pentru inductanțe L sau pulsații ω diferite, considerând C constant. Deasemenea din fig. 9.10 se poate remarca că o creștere a lui d determină o scădere a defazajului, pe când o scădere a lui d implică o creștere a defazajului.

Pornind de la elementele prezentate mai sus și luând în considerare și faptul că în multe aplicații industriale parametrii sistemului nu sunt cunoscuți cu precizie sau își modifică valorile pe durata procesului (cum ar fi rezistența circuitului rotoric într-o mașină asincronă datorită încălzirii) se urmărește realizarea unui regulator fuzzy care să controleze defazajul dintre curent și tensiune.

Regulatorul are 2 intrări:

- prima este variația erorii și reprezintă diferența dintre defazajul de referință (φ_r) și defazajul calculat (φ_c) în circuit la momentul respectiv: $e_i = \varphi_r - \varphi_c$.
- a doua intrare este variația erorii Δe_i și reprezintă diferența dintre eroarea la momentul respectiv e_i și eroarea la momentul anterior e_{i-1} .

Ieșirea controlerului reprezintă variația factorului de umplere Δd . Astfel, valoarea factorului de umplere aplicată pentru a controla puntea H este $d_i = d_{i-1} + \Delta d$, unde d_{i-1} reprezintă factorul de umplere de la momentul anterior.

În ceea ce privește regulile de funcționare ale controlerului acestea sunt date în tabelul 9.1.

$\Delta e \rightarrow$ $e \downarrow$	NB	NM	NS	ZE	PS	PM	PB
NB	PB	PB	PB	PB	PB	PM	PM
NM	PM	PM	PM	PM	PS	PS	PS
NS	PS	PS	PS	PS	PS	PS	PS
ZE	ZE	ZE	ZE	ZE	ZE	ZE	ZE
PS	NS	NS	NS	NS	NS	NS	NS
PM	NS	NS	NS	NM	NM	NM	NM
PB	NS	NS	NM	NM	NM	NM	NM

Tabel 9.1- Regulile de inferență

ex: dacă e este NM și dacă Δe este NS atunci Δd este PM

Funcțiile de apartenență ale variabilelor de intrare și ieșire sunt prezentate în figura 9.11.

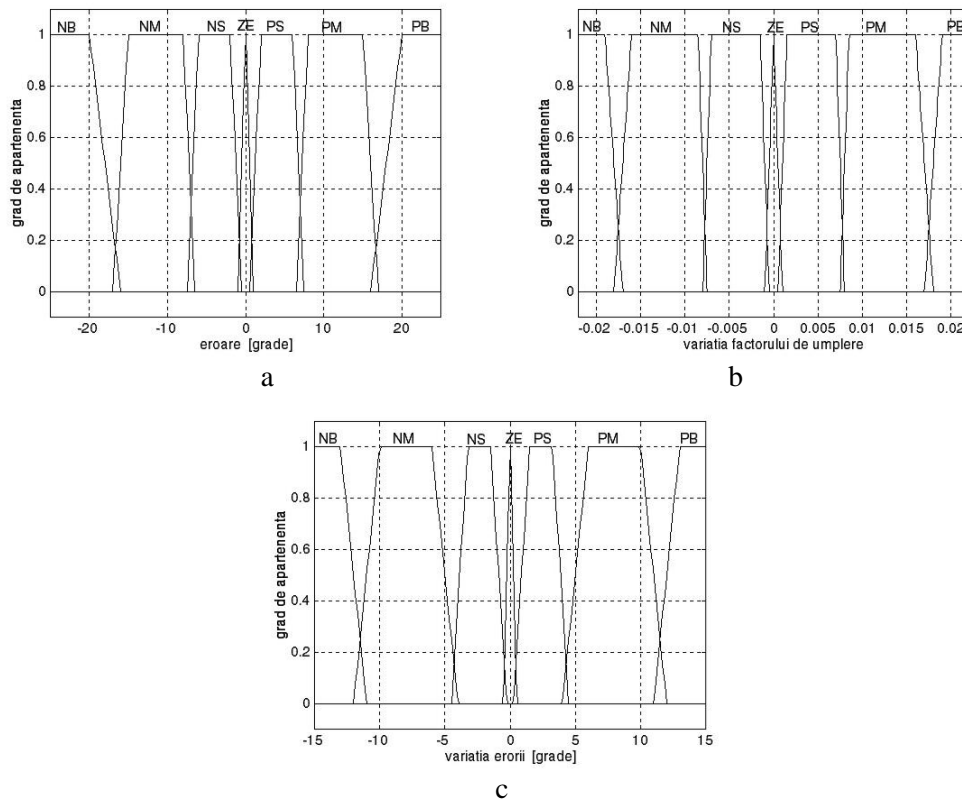


Fig. 9.11 – Funcțiile de apartenență pentru
a) eroare, b) variația erorii, c) variația factorului de umplere

Testarea controlerului s-a făcut pe un circuit cu parametrii $L=56.5\text{mH}$, $R=23.5\Omega$, $\omega=314\text{rad/s}$, $U=15.5\text{V RMS}$. Pentru aceasta s-a folosit sistemul din figura 9.12.

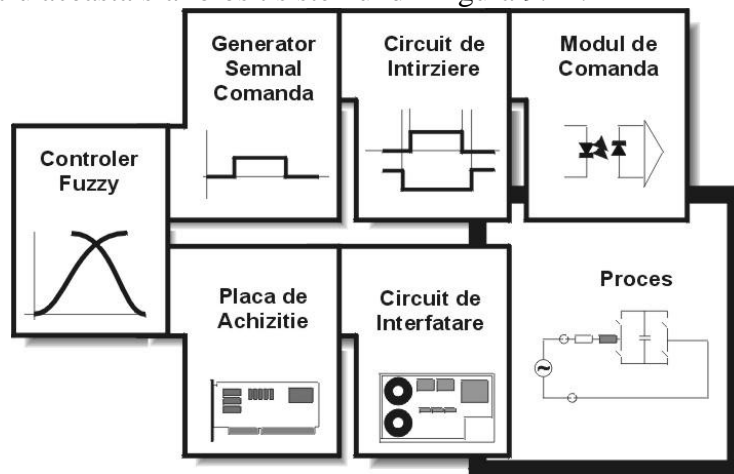


Fig. 9.12 – Schema bloc a experimentului

Comutatoarele bidirecționale ale punții sunt implementate folosind o structură ca cea din fig. 9.13. În mijlocul punții s-a plasat un condensator $C=15\mu\text{F}$.

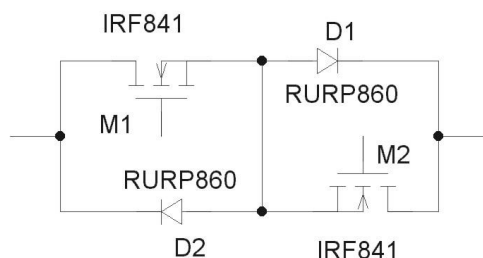


Fig. 9.13 – Implementare comutator bidirecțional

Modulul de comandă are rolul de a realiza izolarea între partea care generează semnalul de comandă și partea de putere. Circuitul de întârziere împiedică conducția simultană a perechilor (S_1, S_4) și (S_2, S_3), prevenind o descărcare nedorită a condensatorului. Generatorul de semnal comandă produce un semnal PWM în concordanță cu referința furnizată de către controlerul fuzzy.

Controlerul fuzzy este implementat software în limbajul C pe un computer de tip PC. Programul constă dintr-o buclă de tip *while* care în prima fază realizează eșantionarea și memorarea simultană a tensiunii de intrare și curentului din circuitul din fig. 9.12 pe interval de o perioadă (20ms). Folosind seriile Fourier se calculează defazajul dintre fundamentalele de curent și tensiune, fiind posibilă astfel calcularea erorii defazajului și variația acesteia și care se vor constitui în intrări ale algoritmului fuzzy. În prima fază, acesta calculează valoarea tuturor funcțiilor de apartenență pentru fiecare din aceste două mărimi în parte, apoi deduce care dintre regulile de inferență sunt activate și, în final, se obține variația factorului de umplere Δd ca urmare a defuzzificării prin metoda centrului de greutate. Cunoșcând Δd se calculează noua valoare a factorului de umplere d care este transmisă către generatorul de semnal de comandă.

Placa de achiziție este un produs National Instruments și este folosită la o rată de achiziție de 16kHz pe canal iar pe circuitul de interfațare se află două traductoare LEM, unul de tensiune și unul de curent.

Pornind de la parametrii circuitului RL se poate stabili că defazajul natural este de aproximativ 35° . Introducerea condensatorului comutat la un factor de umplere de 0.5 afectează neesențial caracteristicile circuitului.

În prima fază se setează defazajul de referință la 0° . În fig. 9.14a este prezentată evoluția defazajului funcție de iterații, iar în 9.14b evoluția factorului de umplere d funcție de iterații. La fiecare iterație o nouă valoare calculată, de către controler, a lui d este aplicată punții H. Intervalul dintre 2 iterații succesive este determinat de timpul de eșantionare/memorare a tensiunii și curentului pentru analiza Fourier și timpul de calcul solicitat de algoritmul fuzzy. În fig. 9.14c sunt prezentate tensiunea și curentul la iterația 22. Formele undă sunt afișate cu valorile furnizate de către traductoare. Calculul defazajului se face având fazorul de curent ca referință.

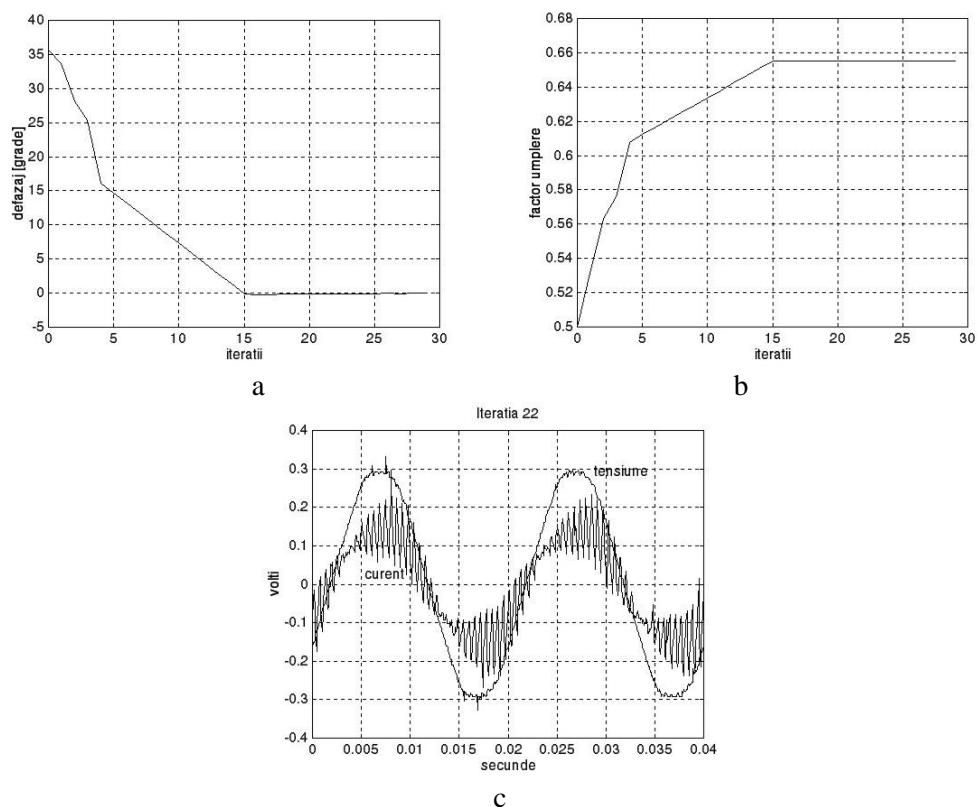


Fig. 9.14 – Defazaj (a), factor de umplere(b) și forme de undă(c) la factor de putere unitate

În cea de a doua fază, defazajul este setat inițial la -10^0 ca pe durata procesului, la iterația 24, să fie schimbat la 10^0 . Reacția defazajului și factorului de umplere sunt reprezentate în fig. 9.15.

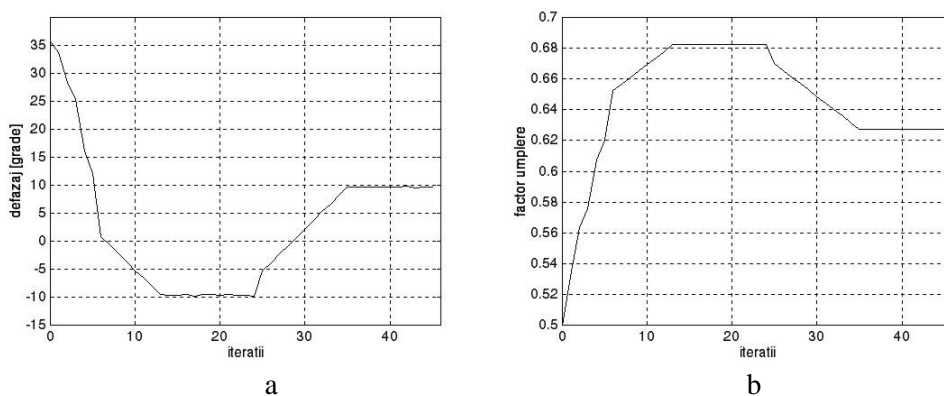


Fig. 9.15 - Defazaj (a) și factor de umplere (b) la defazaj impus variabil

9.5.2. Parcarea unui camion.

Această aplicație este un exemplu clasic în cadrul reglării fuzzy. Un camion trebuie parcat cu spatele perpendicular pe rampa de lucru într-o zonă de încărcare. Camionul realizează această operațiune numai în marșalier.

Variabilele de intrare pentru această aplicație sunt: x – poziția camionului relativ la abscisă și ϕ unghiul dintre axul mașinii și abscisă (se presupune că există suficient spațiu de-a lungul axei y).

pentru a fi parcat camionul, astfel că această variabilă este neglijată). Ca și variabilă de ieșire avem unghiul θ cu care este rotit volanul sau, echivalentul său, unghiul dintre roțile camionului și axul mașinii.

După ce au fost stabilite variabilele lingvistice (x, ϕ, θ) , urmează a fi atașate și valorile fuzzy.

Pentru $\phi = \{ \text{RB /*dreapta jos*/, RU /* dreapta sus*/, RV /*dreapta vertical*/, VE, /*vertical*/, LV /*stînga vertical*/, LU /*stînga sus*/, LB /*stînga jos*/} \}$

Pentru $x = \{ \text{LE, /*stînga*/, LC /*stînga centru*/, CE /*centru*/, RC /*dreapta centru*/, RI /*dreapta*/} \}$

Pentru $\theta = \{ \text{NB, NM, NS, ZE, PS, PM, PB} \}$ ce au semnificația prezentată anterior.

Domeniile clasice de variație a variabilelor lingvistice sunt:

$$x \in [0m, 100m], \phi \in [-90^0, 270^0], \theta \in [-30^0, 30^0]$$

În figura 9.16 sunt reprezentate funcțiile de apartenență pentru fiecare valoare a variabilelor lingvistice.

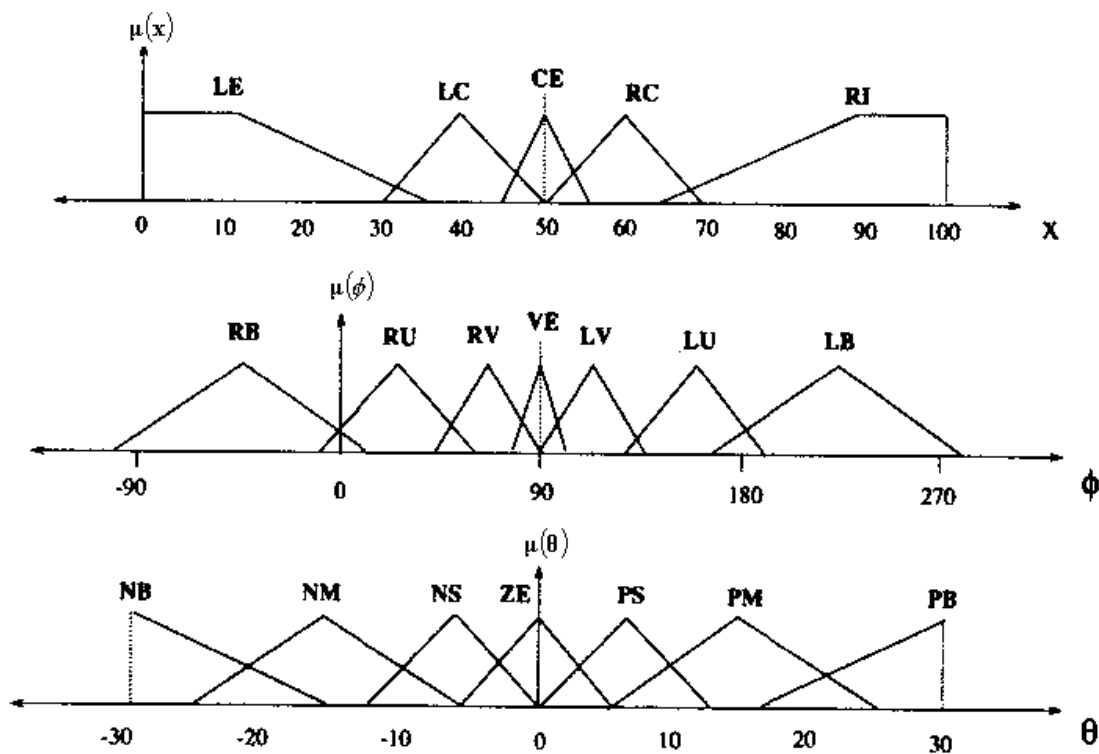


Fig. 9.16 - Funcțiile de apartenență folosite în operația de parcare

Pentru exemplificare: $\theta = 20^0$ este PM în proporție de 0.5 și PB în proporție de 0.3.

Regulile de bază ce guvernează controlerul sunt prezentate în tabela de mai jos:

		x				
		LE	LC	CE	RC	RI
ϕ	RB	PS	PM	PM	PB	PB
	RU	NS	PS	PM	PB	PB
	RV	NM	NS	PS	PM	PB
	VE	NM	NM	ZE	PM	PM
	LV	NB	NM	NS	PS	PM
	LU	NB	NB	NM	NS	PS
	LB	NB	NB	NM	NM	NS

Tabelul 9.2 – Regulele de inferență
ele fiind de de tipul:

Dacă x este CE și ϕ este LV atunci θ este NS.

în total existînd 35 de reguli.

Calcularea variației semnalului de ieșire se face folosind metoda *centrului de greutate*.

În fig.9.17 este prezentat calculul variabilei de ieșire în cazul în care variabilele de intrare ϕ și x se încadrează în domeniile fuzzy RV și VE respectiv CE.

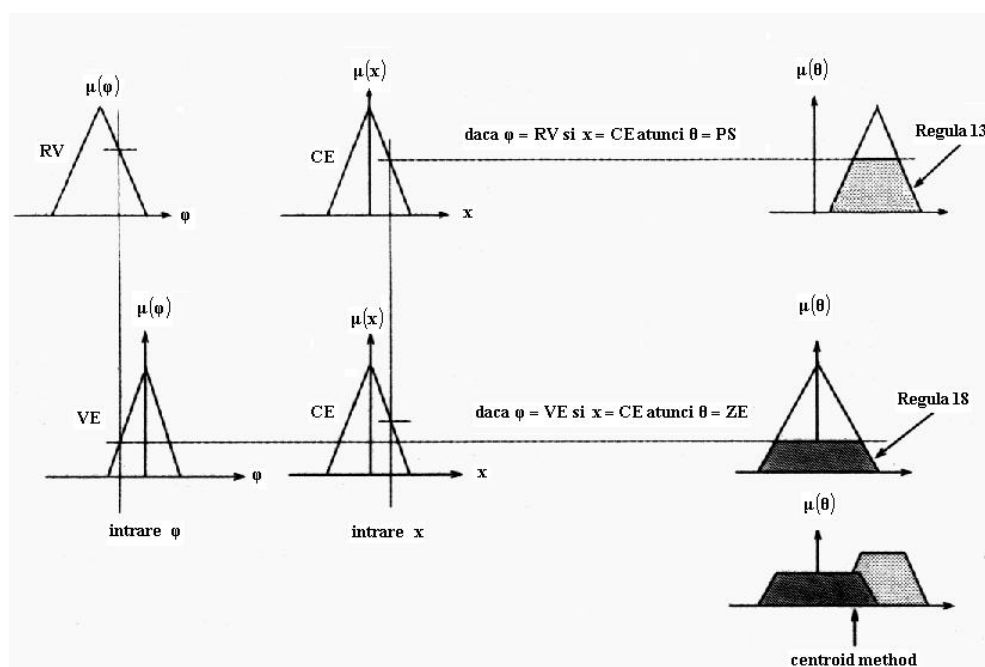


Fig.9.17 – Calculul variabilei de ieșire

Cîteva exemple de evoluție a traiectoriei camionului de la diferite poziții inițiale (x, ϕ) sunt prezentate. (fig.9.18 – fig. 9.20)

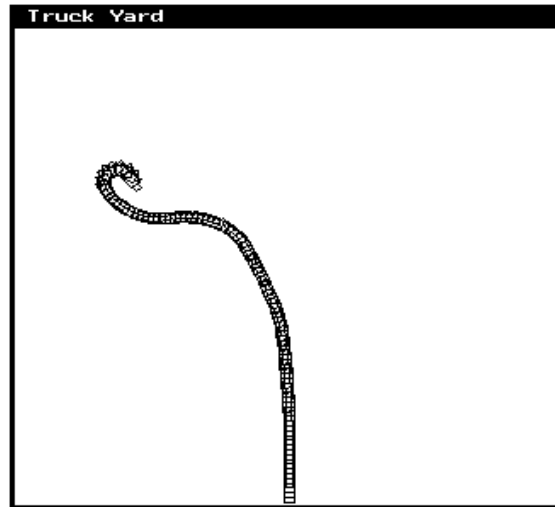


Fig.9.18 – $x = 20, \phi = 235^0$

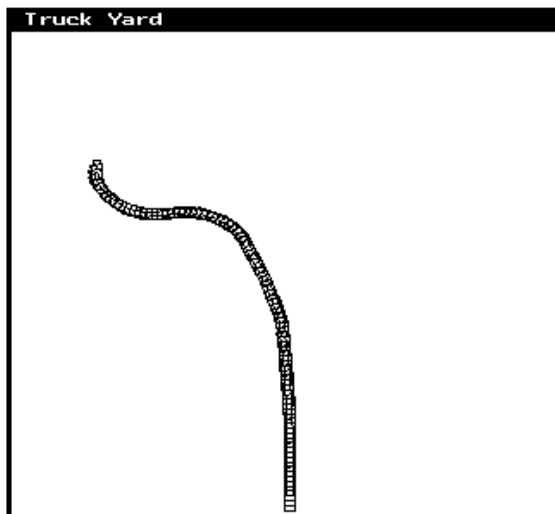


Fig.9.19 – $x = 15, \phi = 90^0$

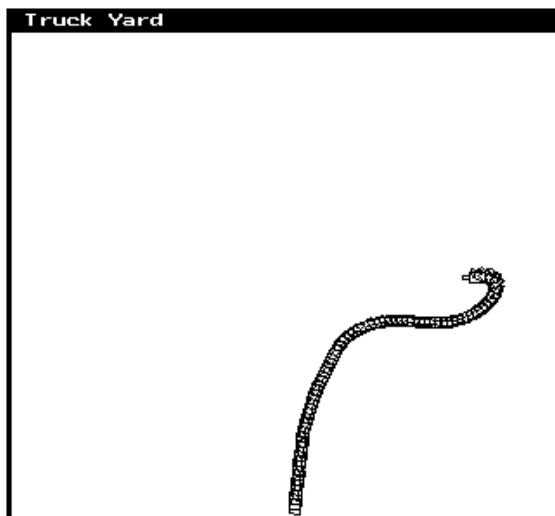


Fig.9.20 – $x = 85, \phi = 0^0$

9.6. Bibliografie

[9.1] C. Suciu, M. Kansara, P. G. Holmes, and W. Szabo, “Phase advancing for current in R-L circuits using switched capacitors” , IEE Electronic Letters, vol. 35, pp. 1296–1297, Aug. 1999.