

SISTEM DE CONTROL FUZZY

dr. ing. Florin Stănciulescu

Institutul de Cercetări în Informatică

Rezumat: Se prezintă un sistem de control fuzzy care utilizează un regulator bazat pe prelucrarea de cunoștințe fuzzy. Modelul care stă la baza funcționării sistemului de control fuzzy este un model de control hibrid (matematico-euristic) compus dintr-un model de simulare a procesului controlat (continuu sau discret), un model de simulare a proceselor cu evenimente discrete și un model de control fuzzy. Se prezintă schema bloc a sistemului de control fuzzy și o aplicație a acestui sistem la controlul unui sistem hidrologic.

Cuvinte cheie: control fuzzy, model hibrid, simulare hibridă, algoritm, aplicație.

1. Introducere

Este cunoscut specialiștilor în sisteme de control/ reglare automată faptul că, în ultimii ani, am asistat la o dezvoltare rapidă a sistemelor de control care utilizează concepte și tehnici derivate din teoria mulțimilor fuzzy [1,3,10,12]. Un astfel de sistem este și sistemul de control fuzzy, prezentat în această lucrare, și care este bazat pe logica fuzzy.

Elementele care stau la baza sistemelor de control fuzzy sunt următoarele: logica fuzzy, un set de variabile lingvistice, o colecție de reguli fuzzy, un graf fuzzy și un motor inferențial fuzzy.

Logica fuzzy, în sens restrâns, este un sistem logic, echivalent cu o extensie a logicii multivalente; dar în sensul mai larg, care este predominant astăzi, logica fuzzy este coextensivă cu teoria mulțimilor fuzzy și deci este mai mult decât un sistem logic; ea se bazează pe:

Un set de *variabile lingvistice*, include variabile ale căror valori sunt mai degrabă cuvinte (adjective), decât numere:

O colecție de *reguli if-then fuzzy*, include reguli euristice de forma:

$$\langle \text{if } X_{i1} \text{ este } A_{i1} \text{ și } \dots X_{in} \text{ este } A_{in}, \\ \text{then } Y_i \text{ este } B_i \rangle, i = 1, 2, \dots, m$$

unde: $X_{i1}, X_{i2}, \dots, X_{in}$ și Y_i sunt variabile lingvistice, iar $A_{i1}, A_{i2}, \dots, A_{in}$ și B_i sunt valorile lingvistice ale acestora.

Un *graf fuzzy* servește pentru a descrie înțelesul unei colecții de reguli fuzzy.

Motorul inferențial fuzzy / (expert system) shell fuzzy, este capabil să efectueze deducții logice pe baza inferării regulilor euristice fuzzy.

În contrast cu sistemele de control clasic, sistemele de control fuzzy au ca punct de plecare o soluție în care intervine experiența operatorului

uman, pe care creatorul sistemului de control fuzzy o formalizează cu ajutorul unui set de reguli euristice fuzzy. Aceste reguli pot fi apoi compilate cu ajutorul unui (expert system) shell într-un limbaj de programare standard (ex. TurboProlog, Clips).

Trebuie precizat faptul că, încă de la apariția lor, sistemele de control fuzzy au constituit un obiect de controversă, datorită mai ales, așa cum susține L. Zadeh [12], unei percepții greșite a ceea ce este și ceea ce nu este un sistem de control fuzzy. Pentru a clarifica încă din start poziția noastră față de controversa amintită, vom arăta ca sistemele de control clasic vor continua să se dezvolte în acele domenii în care sistemul este bine definit (și deci este posibilă elaborarea unui model matematic). În același timp, sistemele de control fuzzy se vor dezvolta în acele domenii în care sistemul nu este bine definit (și deci un model matematic pur nu poate fi bine elaborat, acesta fiind înlocuit cu un model hibrid, de exemplu un model matematico-euristic).

Avantajele sistemelor de control fuzzy constau din următoarele:

- pot fi utilizate în cazul unor sisteme în care dependentele dintre stări și comenzi sunt cunoscute doar calitativ;
- pot exploata toleranța la imprecizie, incertitudine și adevăr parțial, pentru a realiza robustețea soluției, un cost scăzut al acesteia și un raport bun cu realitatea;
- pot fi aplicate atât la probleme orientate pe seturi de date, cât și la probleme orientate pe sarcini (task).

2. Modelul de simulare și control fuzzy

Modelul de simulare și control fuzzy este de fapt un model hibrid, compus din mai multe modele diferite, interconectate. Spre deosebire de modelarea matematică clasică, modelarea hibridă include modele de tipuri diferite (trebuie subliniat faptul că în afara celor menționate anterior, modelul hibrid poate include și modele care derivă din statistica și calculul probabilităților, din cercetarea operațională). Pentru conciziunea expunerii, dar și pentru faptul că experiența ne-a arătat că sistemele mari, complexe pot fi descrise utilizând cele 3 modele de mai sus, în continuare vom dezvolta

aceste modele. Cititorul poate întregi acest tablou cu oricâte modele dorește, dacă din aceasta rezultă un câștig real în rezolvarea problemelor.

Modelul de simulare numerică cu timp continuu, al unui sistem mare, complex, este un sistem de ecuații diferențiale neliniare, a cărui formă standard este următoarea:

$$x_i(t) = A_i x_i(t) + B_i u_i(t) + f_i(x_i, \alpha_i) + v_i(x) \quad (1)$$

$$x_i(0) = x_{i0} \quad (2)$$

$$v_i(x) = \sum_{j=1, j \neq i}^n g_{ij}(x_j) \quad (3)$$

$$x_{i \min} \leq x_i \leq x_{i \max}, \quad (i = 1, 2, \dots, n) \quad (4)$$

unde: $x_i, v_i: R^{n_i} \rightarrow R$, $u_i: R^{m_i} \rightarrow R$ și unde A_i și B_i sunt matricile de stare și control, f_i - o funcție vectorială care descrie neliniaritățile din proces. Forma (1) a modelului este preferabilă unei forme mai generale deoarece, spre deosebire de aceasta din urmă, ecuația (1) pune în evidență interacțiunile $v_i(x)$ ale procesului modelat și separa partea liniară de cea neliniară, fapt care are importanța lui, în procesul de calcul.

Dubla egalitate (4) interpretează dorința expertului ca starea x_i să aparțină intervalului de suboptimalitate $[x_{i \min}, x_{i \max}]$, ale cărui limite sunt supuse unui anumit grad de incertitudine.

De remarcat faptul că în [9] s-au stabilit două intervale fundamentale, unul pentru x_i și altul pentru incertitudinea Δx_i .

$$x_i \in [x_{i \min}, x_{i \max}]$$

$$\Delta x_i \in \left[\left\| C_i \right\|^{-1} \Delta z_i, \left\| C_i^\# \right\| \Delta z_i \right]$$

Dacă se introduce valoarea staționară \bar{X}_i (a lui x_i) dată, de exemplu, de către expert sau obținută prin calcul sau simulare, astfel încât $X_i \in [x_{i \min}, x_{i \max}]$ (cu certitudine), se pot calcula limitele intervalului lui x_i , în funcție de limitele lui Δx_i :

$$x_{i \min} = \bar{x}_i - \left\| C_i \right\|^{-1} \Delta z_i$$

$$x_{i \max} = \bar{x}_i + \left\| C_i^\# \right\| \Delta z_i$$

Lungimea intervalului lui x_i este:

$$x_{i \max} - x_{i \min} = \left(\left\| C_i^\# \right\| + \left\| C_i \right\|^{-1} \right) \Delta z_i$$

Modelul de simulare cu timp discret, al unui sistem mare, complex este descris de un set de

ecuații neliniare cu timp discret, având forma standard:

$$x_i(k+1) = A_i x_i(k) + B_i u_i(k) + f_i(x_i(k), \alpha_i) + v_i(x(k)) \quad (5)$$

incluzând, de asemenea, relațiile (2)-(4); variabila timp discret k ia valori din mulțimea de timp discret $\{t_0, t_1, \dots, t_k, \dots, t_r\}$.

Propagarea incertitudinii. Pentru a simplifica calculul, se utilizează o formă liniarizată a modelului (5):

$$x_i(k+1) = A_i x_i(k) + B_i u_i(k) + \sum_{j=1, j \neq i}^n A_{ij} x_j(k)$$

$$X_i^*(k+1) = A_i X_i^*(k) + B_i u_i^*(k) + \sum_{j=1, j \neq i}^n A_{ij} X_j^*(k)$$

Scăzând $x_i(k+1)$ din $X_i^*(k+1)$ rezultă:

$$X_i^*(k+1) - x_i(k+1) = A_i (X_i^*(k) - x_i(k)) + B_i (u_i^*(k) - u_i(k)) + \sum_{j=1, j \neq i}^n A_{ij} (X_j^*(k) - x_j(k))$$

Utilizând operațiunea de normare și maximum dintr-o normă, rezultă:

$$\Delta x_i(k+1) \leq \sum_{j=1}^n \|A_{ij}\| \Delta x_j(k) + \|B_i\| \Delta u_i(k) \quad (6)$$

$$(A_{ii} \equiv A_i)$$

Inegalitatea (6) arată care este limita superioară a incertitudinii $\Delta x_i(k+1)$ și totodată cum se propagă incertitudinile $\Delta x_i(k)$ ($i = 1, 2, \dots, n$).

Modelul de simulare cu evenimente discrete

În modelul hibrid este inclus și modelul de simulare cu evenimente discrete, a cărui formă standard este un set de ecuații diferențiale booleene.

Un sistem cu evenimente discrete poate fi modelat ca un automat secvențial (vezi figura 1) și poate fi descris cu ajutorul unui sextuplu:

$$G = \{X, U, Y, \delta, \lambda, T\}$$

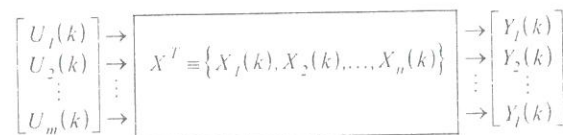


Figura 1. Sistem cu evenimente discrete modelat ca un automat secvențial

unde: X este mulțimea stărilor sistemului cu evenimente discrete, U este mulțimea de intrare, Y este mulțimea de ieșire, $\delta: X \times U \rightarrow X$ este funcția

de tranziție, $\lambda: X \times U \rightarrow Y$ este funcția de ieșire, iar T este mulțimea momentelor de timp discret.

Dacă $X_i \in X$, $U_i \in U$, $Y_i \in Y$ iau valori din mulțimea booleană $\{0, 1\}$, un model diferențial boolean al sistemului cu evenimente discrete poate fi construit, după cum urmează:

Fie $T = \{k_0, k_1, k_2, \dots, k_i, \dots, k_j\}$ mulțimea momentelor de timp discret. Atunci modelul care reprezintă evoluția sistemului cu evenimente discrete, poate fi un model diferențial Boolean:

$$X_i(k+1) = F_i(X_1(k), X_2(k), \dots, X_n(k), U_1(k), U_2(k), \dots, U_m(k)) \quad (7)$$

$$Y_j(k+1) = G_j(X_1(k), X_2(k), \dots, X_n(k), U_1(k), U_2(k), \dots, U_m(k)) \quad (8)$$

$(i=1, 2, \dots, n; j=1, 2, \dots, m)$

F_i este funcția de stare a sistemului cu evenimente discrete, iar G_j este funcția de ieșire a aceluiași sistem. Ambele funcții, F_i și G_j , sunt funcții booleene.

Fie $x_i(t)$ și $x_i(k)$ stările sistemului/procesului analizat (modelat, simulat, controlat); se presupune că variabila t , în cazul procesului continuu, este discretizată în aceeași manieră ca și T . Atunci, pentru ambele cazuri, se pot defini variabilele sistemului cu evenimente discrete astfel:

$$X_i(k) = \begin{cases} 1, & \text{if } x_i(t) \in [x_{i \min}, x_{i \max}] \\ 0, & \text{altfel} \end{cases} \quad (9)$$

$(i=1, 2, \dots, m)$

$$U_j(k) = \begin{cases} 1, & \text{if } u_j(t) \rightarrow x_i(t+1) \in [x_{i \min}, x_{i \max}] \\ 0, & \text{altfel} \end{cases} \quad (10)$$

$(j=1, 2, \dots, n)$

unde intervalul $[x_{i \min}, x_{i \max}]$ a fost definit anterior.

Calculul lui $X_i(k)$ utilizând formula (9) conduce la așa-numita Discrete Event State Matrix - DESM (Matricea Stărilor cu Evenimente Discrete), cu valori booleene (0 sau 1), ca de exemplu:

$$DESM = \begin{matrix} & \begin{matrix} X_1 & X_2 & \dots & X_i & \dots & X_n \end{matrix} \\ \begin{matrix} k_0 \\ k_1 \\ \vdots \\ k_i \\ \vdots \\ k_j \end{matrix} & \begin{bmatrix} 1 & 1 & \dots & 1 & \dots & 1 \\ 1 & 0 & \dots & 1 & \dots & 1 \\ \vdots & \vdots & \dots & \vdots & \dots & \vdots \\ 0 & 1 & \dots & 1 & \dots & 1 \\ \vdots & \vdots & \dots & \vdots & \dots & \vdots \\ 1 & 1 & \dots & 0 & \dots & 0 \end{bmatrix} \end{matrix}$$

Cu ajutorul matricii booleene DESM și al comenzii $U_i(k)$ rezultă $Y_j(k)$.

Modelul de control fuzzy

Modelele de simulare și control fuzzy s-au dezvoltat în ultimii ani, ca urmare a evoluției teoriei mulțimilor vagi. Aceste modele au apărut mai recent în domeniul simulării și al controlului sistemelor complexe, dar au reușit să se impună în unele domenii de aplicație, în care cel puțin o parte din variabilele de intrare, stare sau ieșire sunt de natură fuzzy.

O mulțime fuzzy F , într-un univers de discurs U , este caracterizată prin funcția de apartenență μ_F și este definită ca o mulțime de perechi ordonate ale unui element $u \in U$ și a gradului funcției de apartenență, astfel:

$$F = \{ (u, \mu_F(u)) \mid u \in U \} \quad (11)$$

Pentru astfel de parametri incerti, cum este și gradul funcției de apartenență, se poate utiliza informația adițională, definind conceptul de multinivel (figura 2).

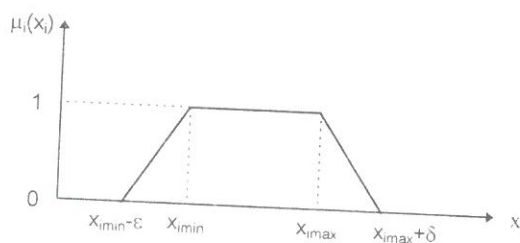


Figura 2. Reprezentarea trapezoidală a funcției de apartenență (Incertitudinea de tip fuzzy, ca distribuție de posibilitate trapezoidală).

Diferitele grade ale funcției de apartenență la o mulțime fuzzy corespund la diferite intervale, numite mulțimi de niveluri, în domeniul variabilei. Mulțimile fuzzy reprezintă situații în care funcțiile de apartenență la o mulțime nu pot fi definite pe baza lui *da-nu* sau cu alte cuvinte, limitele mulțimii sunt vagi. De exemplu, în figura 2, dacă elementul u se află în intervalul $[x_{i \min}, x_{i \max}]$, gradul funcției de apartenență la mulțimea fuzzy F este 1; dacă el se află în intervalul $[x_{i \min} - \epsilon, x_{i \min}]$ sau $[x_{i \max}, x_{i \max} + \delta]$ atunci gradul funcției de apartenență la F este între 0 și 1. Evident $\mu_F = 0$, dacă $u < x_{i \min}$ sau $u > x_{i \max}$.

Se revine la modelul care descrie dinamica unui sistem, în termenii unui sistem de ecuații cu timp discret, de data aceasta sub forma mai generală:

$$x(k+1) = f(x(k), u(k)), x(0) = x_0, \quad (12)$$

$$y(k+1) = g(x(k), u(k)), u(k) \in R^m; \quad (13)$$

unde f descrie dinamica sistemului, g este ieșirea acestuia.

Modelul de simulare fuzzy poate fi formalizat astfel:

$$f: X \times U \rightarrow F(X),$$

$$g: X \rightarrow F(Y),$$

unde f și g sunt relații fuzzy, cu valori în $\{0, 1\}$.

Un model de control fuzzy se formalizează astfel:

$$f: F(X) \times F(U) \rightarrow F(X),$$

$$g: F(X) \rightarrow F(Y),$$

unde X, U, Y reprezintă mulțimea stărilor, a comenzilor și, respectiv, a ieșirilor.

Gradul de apartenență $\mu_i(u)$ a unui element (mărire de stare, parametru, etc.) la o (sub)mulțime fuzzy, poate fi exprimat numeric, dacă universul de discurs este discret (adică un vector care depinde de pasul de discretizare), sau continuu (cu ajutorul unei funcții). Cel mai uzual caz, des folosit în aplicații, este cel al funcției de apartenență trapezoidală reprezentată în figura 2. Din această figură se poate deduce:

$$\mu_i(x_i) = \begin{cases} 1, & \text{daca } x_i \in [x_{i,\min}, x_{i,\max}] \\ 0, & \text{daca } x_i < x_{i,\min} - \varepsilon \text{ sau } x_i > x_{i,\max} + \delta \\ \frac{1}{\varepsilon} x_i - \frac{x_{i,\min} - \varepsilon}{\varepsilon}, & \text{daca } x_i \in [x_{i,\min} - \varepsilon, x_{i,\min}] \\ -\frac{1}{\delta} x_i + \frac{x_{i,\max} + \delta}{\delta}, & \text{daca } x_i \in [x_{i,\max}, x_{i,\max} + \delta] \end{cases} \quad (14)$$

Problema controlului variabilelor de stare $x_i(k)$, $i = 1, 2, \dots, n$; $k = 0, 1, \dots, k_f$, este aceea de a menține aceste variabile în intervalele de (sub)optimalitate $[x_{i,\min}, x_{i,\max}]$. Modelul de control este un model fuzzy. Variabilele de stare $x_i(k)$ aparțin unuia sau altuia din intervalele din tabloul 1, ca și diferența $dx_i(k) = x_i(k) - x_i(k-1)$.

Variabilele de stare x_i aparțin mulțimii calitative:

$$\chi = \{ VS, S, O, L, VL \},$$

unde: $vs=f.mic$, $s=mic$, $O=optim$, $L=mare$, $vl=f.mare$

în timp ce derivatele dx_i aparțin mulțimii calitative:

$$d\chi = \{ -L, -S, Z, S, L \}.$$

unde: $z=zero$.

Formula de calcul al noilor variabile de control $u_i(k+1)$, capabile să readucă variabilele de stare $x_i(k)$ în intervalele de (sub)optimalitate este:

$$u_i(k+1) = u_i(k) + K_{ui} \Delta u_i(k) \quad (15)$$

$$(i = 1, 2, \dots, n)$$

În formula (15) $\Delta u_i(k)$ este un increment/decrement, o creștere (sau o diminuare) a

variabilei de comandă, care poate fi calculată cu ajutorul formulei de calcul:

$$\Delta u_i(k) = \frac{\sum_{i=1}^n \mu_i(x_i) \cdot u_i(k)}{\sum_{i=1}^n \mu_i(x_i)} \quad (16)$$

$$(i = 1, 2, \dots, n)$$

Formula de calcul (16) este expresia matematică a defuzificării prin metoda centrelor de greutate, sau a ponderării variabilelor de comandă $u_i(k)$, cu ajutorul gradului funcției de apartenență $\mu_i(x_i)$.

Termenii K_{ui} sunt coeficienți de calcul experimental, în mod normal furnizați de către expert (vezi tabloul 1):

Tabloul 1

dx_i / x_i	VS	S	O	L	VL
$-L$	K_{ui}^{11}	K_{ui}^{12}	0	K_{ui}^{14}	K_{ui}^{15}
$-S$	K_{ui}^{21}	K_{ui}^{22}	0	K_{ui}^{24}	K_{ui}^{25}
Z	K_{ui}^{31}	K_{ui}^{32}	0	K_{ui}^{34}	K_{ui}^{35}
S	K_{ui}^{41}	K_{ui}^{42}	0	K_{ui}^{44}	K_{ui}^{45}
L	K_{ui}^{51}	K_{ui}^{52}	0	K_{ui}^{54}	K_{ui}^{55}

Combinând cunoștințele din tabloul 1 cu relațiile de calcul (15) - (16), rezultă un set de reguli, bazate pe cunoștințe fuzzy, utile în controlul sistemelor complexe și anume:

< Dacă $(x_i(k)$ este χ_i) și $(dx_i(k)$ este $d\chi_i)$,

Atunci noua regulă de control este:

$$u_i(k+1) = u_i(k) + K_{ui} \Delta u_i(k) > \quad (17)$$

$$(i = 1, 2, \dots, n)$$

Subliniem faptul că numărul de reguli de control fuzzy este $25 \cdot n$ (n este numărul de variabile de stare, iar 25 reprezintă numărul de valori K_{ui}).

Teorema de compatibilitate între modelul matematic (de simulare) și modelul (de control) fuzzy.

Modelul matematic de simulare a procesului (controlat) și modelul fuzzy (de control) nu sunt apriori, în mod automat, compatibile. Compatibilitatea este probată cu ajutorul unei teoreme de compatibilitate.

Teorema de compatibilitate: Modelul de control fuzzy (14)-(16) este compatibil cu modelul matematic de simulare a procesului controlat (5), dacă utilizând un motor inferențial, se pot infera pornind de la regulile euristice fuzzy (17) - noile variabile de control capabile să readucă variabilele de stare în interiorul intervalelor de (sub)optimalitate; aceasta aserțiune se transpune în următoarea condiție:

$$x_{i \min} - \Omega_i(k) \leq \sum_{j=0}^{k-1} A_i^{k-j-1} B_i u_j + \sum_{i=0}^{k-2} \sum_{j=0}^{k-l-2} A_i^{k-j-2} K_{ui} \Delta u_i \leq x_{i \max} + \Omega_i(k)$$

$$\text{unde: } \Omega_i(k) = A_i^k x_{i0} + \sum_{j=0}^{k-1} A_i^{k-j-1} (f_{ij} + v_{ij})$$

Demonstrația teoremei este dată în Anexa.

Algoritm de control fuzzy

Utilizând modelul elaborat anterior, se poate construi următorul algoritm de control fuzzy:

Pasul 1: Rezolvă problema de simulare numerică a procesului controlat, descrisă cu ajutorul modelului cu timp discret (5) și fie $x_i(k)$ valorile obținute prin simulare (pentru $u_j(k)$ dat, ($i = 1, 2, \dots, n$); ($j = 1, 2, \dots, m$); $k \in [k_0, k_f]$;

Pasul 2: La fiecare moment de timp k **verifică** utilizând modelul cu evenimente discrete (7)-(10), dacă variabilele de stare x_i ($i = 1, 2, \dots, n$), obținute la pasul 1, aparțin sau nu intervalului $[x_{i \min}, x_{i \max}]$:

- dacă **nu**, mergi la pasul 3!
- dacă **da**, mergi la pasul 5!

Pasul 3: Utilizând modelul de control fuzzy (14)-(16), **sintetizează** noi valori pentru variabilele de comandă, capabile să readucă variabilele de stare în intervalele de suboptimalitate;

Pasul 4: Întorcete la **Pasul 1** și resimulează sistemul controlat, utilizând noile variabile de comandă, obținute la **Pasul 3**;

Pasul 5: Examinează evoluția variabilelor de stare simulate $x_i(k)$ ($i = 1, 2, \dots, n$), obținute la **Pasul 1**, care aparțin intervalelor de suboptimalitate $[x_{i \min}, x_{i \max}]$, dar sunt foarte aproape de limite:

- dacă:

$$\left(x_i(k+1) < x_i(k) \right) \wedge \left(|x_i(k+1) - x_{i \min}| < \varepsilon_1 \right) \vee \left(x_i(k+1) > x_i(k) \right) \wedge \left(|x_{i \max} - x_i(k+1)| < \varepsilon_2 \right),$$

mergi la **Pasul 3**!

- dacă **NU**, mergi la **Pasul 6**!

Pasul 6: Verifică dacă $k < k_f$:

- dacă **da**, mergi la **Pasul 1**!
- dacă **nu**, mergi la **Pasul 7**!

Pasul 7: Imprimă sau trasează graficele mărimilor de stare $x_i(k)$ ($i = 1, 2, \dots, n$), obținute cu noile mărimi de comandă, de la **Pasul 3**:

Pasul 8: Stop!

3. Structura sistemului de control fuzzy

3.1 Structura regulatorului fuzzy

Structura regulatorului fuzzy este prezentată în figura 3:

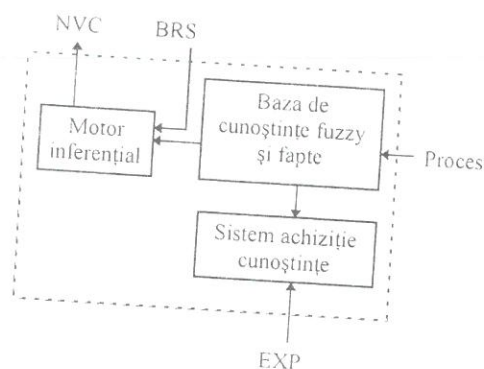


Figura 3. Structura regulatorului fuzzy

Această structură include un sistem de achiziționare a cunoștințelor (SAC), o bază de cunoștințe fuzzy și fapte (BCFF), și un motor inferențial (ENG) și lucrează sub controlul unui (expert system)shell. Ținând seama de modelul de control fuzzy (11)-(17), legea de reglare poate fi definită astfel:

$$u_i(k+1) \leftarrow u_i(k) + K_{ui} \cdot \Delta u_i(k) \quad (i=1, 2, \dots, n)$$

3.2 Simulatorul procesului controlat

Structura simulatorului este redată în figura 4:

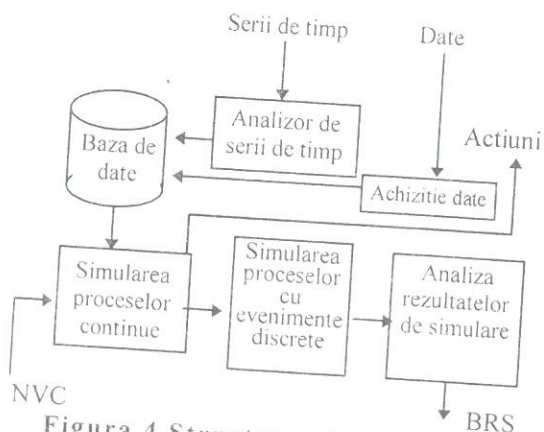


Figura 4. Structura simulatorului procesului controlat

Această structură include o bază de date (BD), un modul de simulare a proceselor continue și/sau discrete (SPCD) și un modul de simulare a proceselor cu evenimente discrete (PED), și nu analizor al rezultatelor de simulare (ARS).

3.3. Structura "blackboard"

Structura modulului "blackboard" este redată în figura 5.

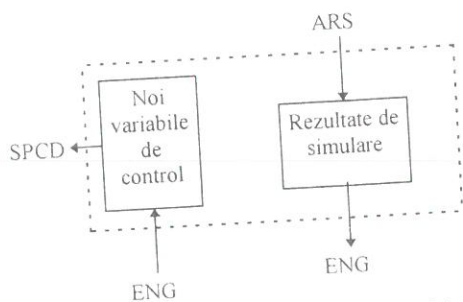


Figure 5. Structura modulului "blackboard"

Ea include baza rezultatelor de simulare (BBS), baza rezultatelor obținute din prelucrarea cunoștințelor (regulilor) fuzzy (BRF) și noile variabile de control (NVC). Rolul său este acela de a asigura legătura între simulatorul de proces și regulatorul fuzzy, stocând rezultatele primite de la simulator și regulatorul fuzzy, și de a le alimenta cu

datele și informațiile necesare.

3.4 Arhitectura sistemului de control fuzzy

Schema bloc a sistemului de control fuzzy este prezentată în figura 6.

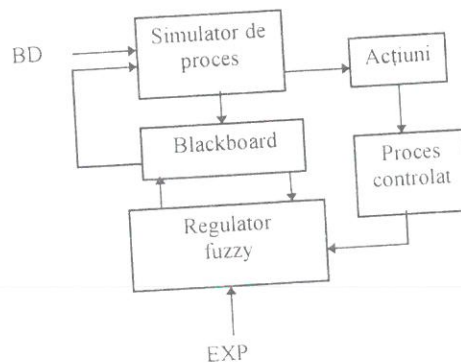


Figure 6. Schema bloc a sistemului de control fuzzy

Suntem acum în situația de a crea arhitectura sistemului de control fuzzy așa cum este ea redată în figura 7. Simulatorul de proces simulează atât procesele continue și/sau discrete, cât și pe cele cu evenimente discrete și alertează regulatorul fuzzy, atunci când variabile(le) de stare ies din intervalele

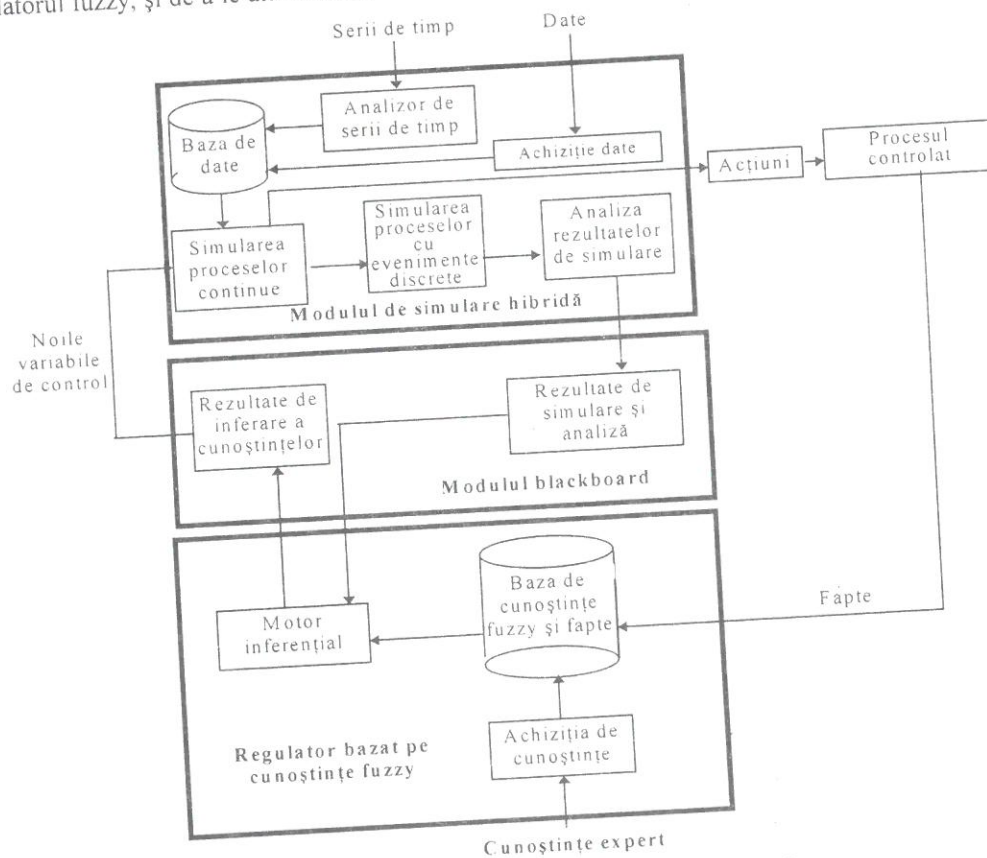


Figure 7. Arhitectura sistemului de control fuzzy

de suboptimalitate, furnizate de către expert. Regulatorul fuzzy procesează cunoștințele (sub forma de reguli euristice fuzzy) și sintetizează noile valori ale variabilelor de comandă, capabile să readucă mărimile de stare în interiorul intervalelor de (sub)optimalitate.

4. Aplicație

Aplicația se referă la un sistem hidrologic complex, compus dintr-o rețea de lacuri interconectate prin canale și șenale. Problema care se pune este aceea de a simula evoluția unor variabile de stare ale sistemului: nivelul apei în lacuri (H_j), adâncimea apei în lacuri (HH_j), volumul apei acumulate în lacuri (V_j) și factorul de împospătare a apei (W_j) și de a controla evoluția factorului de împospătare a apei (W_j), care trebuie să rămână în intervalul $[W_{jmin}, W_{jmax}]$ dat de expert ($j = 1, 2, \dots, n$; în cazul nostru: $j=10$). Se precizează că variabilele de control sunt debitele de intrare a apei (Q_{ij}), care pot fi comandate prin parametrii geometrici ai canalelor de apă (a_{ij}, b_{ij}) și prin factorul de rugozitate (ρ_{ij}).

Modelul hibrid (matematico-euristic) al sistemului/ procesului hidrologic este compus din:

(a) Modelul numeric de simulare a proceselor continue (discretizat), care include câte 4 ecuații de stare, asociate cu fiecare lac, câte una pentru H_j , HH_j , V_j și W_j :

$$H_j(k+1) = H_j(k) + p(k) - e(k) + B_i \cdot Q_{ij}(k) - B_i \cdot Q_{ie}(k) + B_i \cdot u_i(k),$$

$$HH_j(k) = H_j(k) + H_{jmed}(k),$$

($H_{jmed}(k)$ este adâncimea medie a lacului j)

$$V_j(k) = A_j H_j(k),$$

$$W_j(k) = \frac{I_{ij}(k)}{I_{ij}(k) + I_{ie}(k)}$$

unde:

$$Q_{ij}(k) = \rho_{ij} A_{ij} (2g)^{1/2} \text{sign } x_{ij} (|x_{ij}|)^{1/2}$$

- debitul de intrare al apei

$$Q_{ie}(k) = \rho_{ie} A_{ie} (2g)^{1/2} \text{sign } x_{ie} (|x_{ie}|)^{1/2}$$

- debitul de ieșire al apei

$$x_{ij} = H_i - H_j$$

$$x_{ie} = H_j - H_e$$

Modelul de mai sus derivă din legile mecanicii fluidelor și are la bază ipoteza de calcul că apa circulă între două lacuri (prin canale) datorită diferenței de nivel, în ambele sensuri, după cum: $H_i >$

H_j , $H_i < H_j$ sau staționează dacă $H_i = H_j$. Modelul include factorul de rugozitate ρ_{ij} / ρ_{ie} a canalelor de apă și/sau șenalelor, a cărui expresie matematică este:

$$\rho_{ij} = \frac{K_{ij}}{f_{ij}} \cdot \frac{a_{ij} \cdot l_{ij}}{2a_{ij} + l_{ij}}$$

unde: f_i - reprezintă gradul de acoperire cu vegetație a suprafeței canalului [%], K_{ij} - este un factor de ajustare a modelului, a_{ij} - este adâncimea canalului/ șenalului care leagă lacurile i și j [m], l_{ij} - lățimea aceluiași canal/șenal [m].

(b) Modelul de simulare cu evenimente discrete este construit cu ajutorul stărilor cu evenimente discrete ($W_j(k)$), definite astfel:

$$W_j(k) = \begin{cases} 1, & \text{daca } W_i \in [0.2, 0.5], \\ 0, & \text{daca } W_i \notin [0.2, 0.5]. \end{cases}$$

și cu ajutorul ecuațiilor diferențiale booleene:

$$Y(k+1) = W_1(k) \wedge W_2(k) \wedge \dots \wedge W_n(k);$$

$$Y(k+1) = \begin{cases} 1, & \text{procesul de simulare continua} \\ 0, & \text{este alertat regulatorul fuzzy, care} \\ & \text{va sintetiza noile variabile de control} \\ & (Q_{ij}(k), \text{ respectiv: } a_{ij}, b_{ij} \text{ si } f_{ij}). \end{cases}$$

($i=1, 2, \dots, n$)

(c) Modelul de control fuzzy

Variabilele de stare controlate fiind $W_j(k)$, $i=1, 2, \dots, n$, pentru a pune la lucru regulatorul fuzzy, trebuie materializați parametrii modelului de control fuzzy din tabloul 1, respectiv: $x_{imin} = 0.2$, $x_{imax} = 0.5$, $\varepsilon = 0.1$, $\delta = 0.3$. Expresia matematică a funcțiilor de apartenență $\mu_i(W_i(k))$ este:

$$\mu_i(W_i(k)) = \begin{cases} 1, & \text{if } W_i \in [0.2, 0.5], \\ 0, & \text{if } W_i \in [0, 0.1] \text{ or } W_i \in (0.8, 1], \\ 10W_i - 1, & \text{if } W_i \in [0.1, 0.2] \\ -\frac{10}{3}W_i + \frac{8}{3}, & \text{if } W_i \in (0.5, 0.8] \end{cases}$$

Noile variabile de control $f_i(k)$ se calculează astfel:

$f_i(k+1) = f_i(k) + Kf_i \Delta f_i(k)$, unde:

$$\Delta f_i(k) = \frac{\sum_{i=1}^6 \mu_i(W_i) \cdot f_i(k)}{\sum_{i=1}^6 \mu_i(W_i)}$$

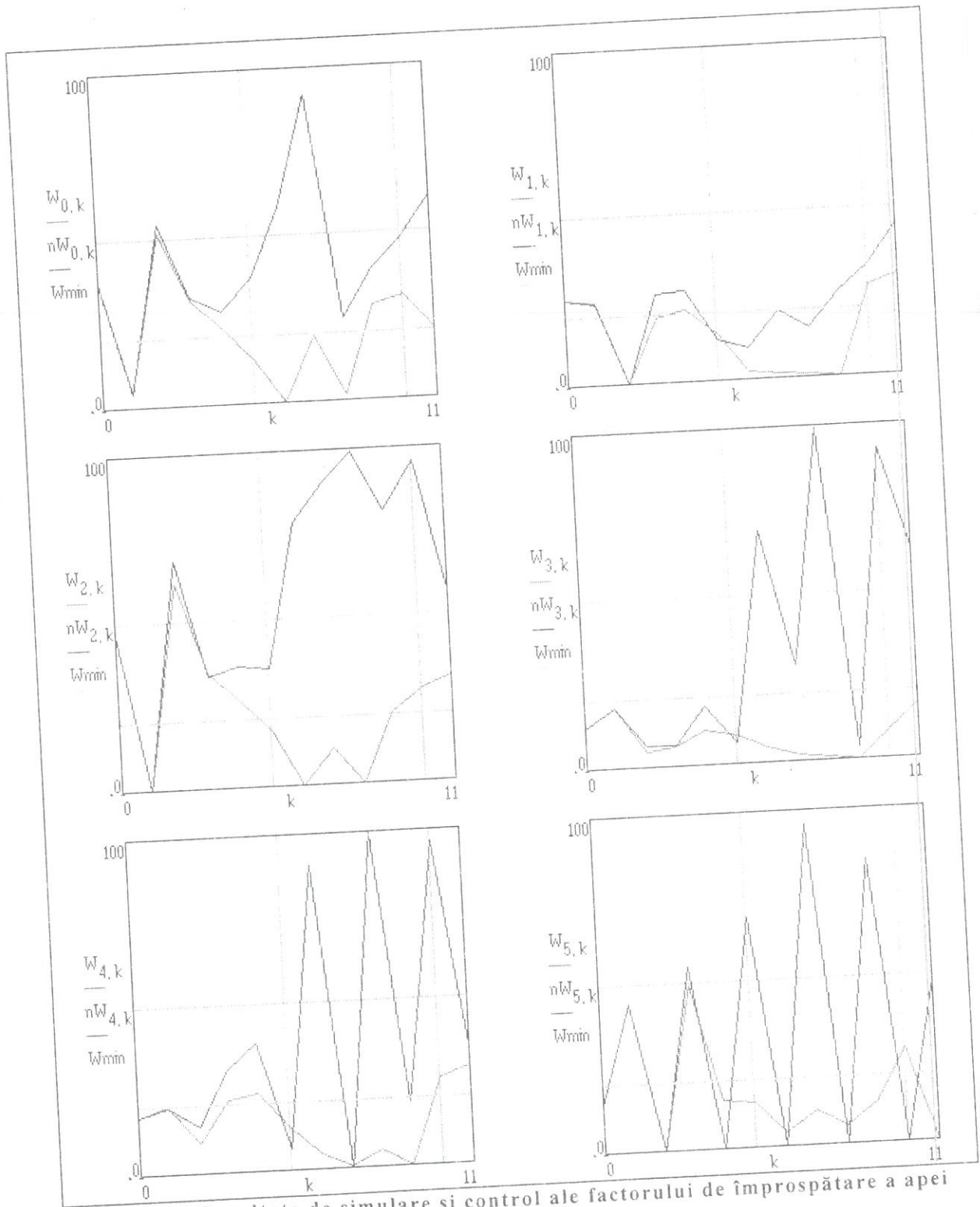


Figura 8. Rezultate de simulare și control ale factorului de împrăștiere a apei într-un sistem hidrologic complex:

- $W_{ij}(k)$, reprezintă graficele variabilelor de stare în regim natural (necontrolat);
- $nW_{ij}(k)$, reprezintă graficele variabilelor de stare în regim controlat (utilizând acțiunile bazate pe măsurimile de comandă sintetizate de regulatorul fuzzy)

Coefficienții euristici Kf_i sunt dați de către expert (vezi matricea de mai jos):

Tabloul 2

dW_i, W_i	$V'S$	S	O	L	VL
$-L$	-0.5	-0.35	0	0.35	0.5
$-S$	-0.35	-0.25	0	0.25	0.35
Z	-0.15	0	0	0	0.15
S	-0.35	-0.25	0	0.25	0.35
L	-0.5	-0.35	0	0.35	0.5

Regulile euristice de control, utilizate în evaluarea noilor variabile de control f_i (sunt în total 25 de reguli euristice de control fuzzy) au forma:

$\langle \text{Daca } (W_i(k) = S) \wedge (dW_i(k) = -L) ,$

Atunci legea de control este:

$$f_i(k+1) = f_i(k) - 0.35 \Delta f_i(k) \rangle$$

(și încă 24 de reguli de control fuzzy similare).

Rezultate de simulare și control fuzzy au fost obținute utilizând modelul de simulare a procesului hidrologic și modelul fuzzy al mărimii de stare $W_j(k)$, algoritmul de control fuzzy descris în Cap.2, limbajul de simulare Mathcad sub Windows, precum și un (expert system) shell (au fost utilizate două versiuni, una în TurboProlog, alta utilizând limbajul Clips). Experimentele de simulare și control au fost realizate cu ajutorul unui calculator personal PC486 (compatibil IBM), utilizând date și cunoștințe fuzzy, achiziționate de la experți (hidrologi și ecologi) și fapte din procesul hidrologic. Rezultatele obținute prin simulare sunt prezentate în figura 8 și ele arată că în sistemul natural (necontrolat) factorul de înprospătare a apei era inferior limitei $W_{\text{min}} = 0.1$ (adică 10%) $j=0,1, 2, \dots, 5$ pentru o lungă perioadă de timp ($k=3 \div 7$).

După aplicarea acțiunilor rezultate din evaluarea noilor comenzi sintetizate de regulatorul fuzzy, factorul de înprospătare a apei a crescut spectaculos (a se vedea noile curbe $nW_j(k)$).

5. Concluzii

Este prezentat un sistem de control fuzzy bazat pe un model hibrid de simulare și control conceput să prelucreze cunoștințe preluate de la experți (sub forma unui set de reguli euristice de control fuzzy), utilizând un algoritm de control fuzzy. Piesa centrală a sistemului de control fuzzy este regulatorul fuzzy, compus din sistemul de achiziționare a cunoștințelor, baza de reguli euristice fuzzy și fapte și motorul inferențial. Sistemul de control fuzzy include, pe lângă regulatorul fuzzy, simulatorul procesului și modulul blackboard.

Aplicația tratată și prezentată în lucrare, referitoare la un sistem hidrologic complex, a arătat că rezultatele de simulare și control obținute sunt credibile și utile în practică, pentru ecologi și hidrologi.

Elementele de noutate prezentate în această lucrare sunt: modelul hibrid de simulare și control (în special modelul de control fuzzy), algoritmul de control hibrid (fuzzy), teorema de compatibilitate între modelul matematic de simulare și modelul de control fuzzy. Dar elementul de noutate cel mai interesant este indiscutabil *regulatorul fuzzy*, care are numeroase alte aplicații.

Mulțumiri

Autorul este îndatorat d-rei *Adriana Zamfir* și d-lui *Marius Mincă* pentru contribuția lor la implementarea pe calculator și tehnoredactarea computerizată.

Bibliografie

1. GUPTA M, KISZKA J, TROJAN G. Multivariable Structure of Fuzzy Control Systems. In: IEEE Transactions on Systems, Man and Cybernetics, 1990, 20/2, pp. 419-435.
2. JUTTNER TH., UNBEHAUEN H.: An Expert System Application in the Field of Adaptive Control. In: Proceedings of the third European Control Conference (Isidori, A. et all eds). Universita di Roma "La Sapienza", Roma, 1995, pp. 1607-1612.
3. LIU K, LEWIS FL. Implement Variable Structure Control via Fuzzy Logic. In: Proceedings of the third European Control Conference (Isidori, A. et all eds), Universita di Roma "La Sapienza", Roma, 1995, pp 803-808.
4. SHEURING R, WEHLAN H. Control of Discrete Event Systems by Means of the Boolean Differential Calculus. In: Discrete Event System: Modelling and Control (Balemi S. et all eds), Birkhauser, Basel, 1993, pp 79-93.
5. SHOURESHI A. R. & ALL, Intelligent Control Systems. In: 13th IFAC Triennial World Congress, San Francisco, USA, 1996, pp. 101-106.
6. STANCIULESCU F.: Control of Large Scale System Using a Knowledge-Based Controller. In: Proceedings IFAC Symposium on Design Method of Control Systems (Franke D., Kraus F eds), ETH, Zurich, 1991, pp. 783-788.

7. STANCIULESCU F.: The Control of Complex Systems by Using a Knowledge Processing Technique. In: Proceedings of the second European Control Conference (Nieuwenhuis JW, et all. eds.), University of Groningen, Groningen, 1993, pp. 1898-1903.
8. STANCIULESCU F.: A Hybrid Intelligent Control System Using a Knowledge-Based Controller. In: Proceedings of the third European Control Conference (Isidori A, et all eds.), Universita di Roma "La Sapienza", Roma, 1995, pp. 1613-1618.
9. STANCIULESCU F.: Hybrid Modelling of Large Scale and Complex Systems Under Uncertainty Conditions, 1996 (in manuscris).
10. TANAKA K, SUGENO M.: Stability Analysis and Design of Fuzzy Control Systems. In: Fuzzy Sets and Systems, 1992, 45, pp. 135-156.
11. WONHAM W.M., RAMADGE P.J.: Modular Feedback Logic for Discrete Event Systems, Systems Control Group Report no.8614, University of Toronto, 1984.
12. ZADEH A.L., Fuzzy Control: Issues, Contention and Perspectives In: 13th IFAC Triennial World Congress, San Francisco, USA, 1996, pp. 35-38.

Anexă

Demonstrația teoremei de compatibilitate

Teorema de compatibilitate stabilește condiții necesare și suficiente, astfel încât modelul de simulare al sistemului controlat și modelul de control fuzzy să fie compatibile. Pentru a demonstra aceasta teoremă se pornește de la modelul de simulare cu timp discret al procesului controlat (5), reprezentând ecuația de stare standard a acestui sistem/proces și se presupune că la momentul de timp k , $x_i(k)$ iese din intervalul de (sub)optimalitate, într-un sens sau altul. S-a arătat că utilizând principiile controlului (reglării) bazat(ă) pe regulatorul fuzzy, așa cum au fost ele sintetizate în algoritmul de control fuzzy (Cap.2), se poate determina o nouă variabilă de control, adăugând valorii $u_i(k)$ valoarea $K_{ui}\Delta u_i(k)$. Astfel, noua variabilă de control este:

$$u_i(k+1) = u_i(k) + K_{ui} \cdot \Delta u_i(k),$$

unde: K_{ui} este dat de către expert, iar $\Delta u_i(k)$ este calculat cu ajutorul formulei (16); rezultă expresia variabilei de stare x_i la momentul de timp $k+1$:

$$x_i(k+1) = A_i x_i(k) + B_i \left\{ u_i(k) + K_{ui} \frac{\sum_{j=1, j \neq i}^n \mu_j(x_j) u_j(k)}{\sum_{j=1, j \neq i}^n \mu_j(x_j)} \right\} + \quad (A1)$$

$$+ f_i(x_i(k)) + \sum_{j=1, j \neq i}^n g_{ij}(x_j(k))$$

Se pune acum condiția ca restricția de apartenență a mărimii de stare $x_i(k+1)$ la intervalul $[x_{imin}, x_{imax}]$, să fie îndeplinită:

$$x_{imin} \leq A_i x_i(k) + B_i \left\{ u_i(k) + K_{ui} \frac{\sum_{j=1, j \neq i}^n \mu_j(x_j) u_j(k)}{\sum_{j=1, j \neq i}^n \mu_j(x_j)} \right\} + \quad (A2)$$

$$f_i(x_i(k)) + \sum_{j=1, j \neq i}^n g_{ij}(x_j(k)) \leq x_{imax}$$

După unele calcule elementare, din relația (A2) rezultă:

$$\begin{aligned} B_i^{-1}(x_{imin} - \omega_i) &\leq \\ &\leq u_i(k) + K_{ui} \frac{\sum_{j=1, j \neq i}^n \mu_j(x_j) u_j(k)}{\sum_{j=1, j \neq i}^n \mu_j(x_j)} \leq \quad (A3) \\ &\leq B_i^{-1}(x_{imax} - \omega_i), \end{aligned}$$

unde:

$$\omega_i(k) = A_i x_i(k) + f_i(x_i(k)) + \sum_{j=1, j \neq i}^n g_{ij}(x_j(k)) \quad (A4)$$

De subliniat faptul că x_{imin} și x_{imax} sunt date de către expert și că ω_i este calculabil, deoarece valoarea lui $x_i(k)$ este ea însăși calculabilă recursiv, pornind de la $x_i(0)$ și $u_i(0)$ și construind șirul de valori:

$$\{x_i(0), \{x_i(1)\}, \{x_i(2)\}, \dots, \{x_i(k)\}\}.$$

Acum se poate enunța teorema de compatibilitate, așa cum a fost ea dată în Cap.2.

Remarcă: Teorema de compatibilitate poate fi privită ca bază teoretică a algoritmului de control hibrid/fuzzy, dat în Cap.2, deoarece ea asigură condițiile pentru a avea o soluție (furnizată de regulatorul fuzzy), capabilă să readucă variabilele de stare în intervalele de (sub)optimalitate, atunci când acestea ies din intervale (condițiile (A3) și (A4)).