

# ARHITECTURI RECURENTE NEUROFUZZY PENTRU CONDUCEREA PROCESELOR NELINIARE

Conf. dr. ing. Nicolae Constantin

Universitatea Politehnica București  
E-mail: [nicu@cib.pub.ro](mailto:nicu@cib.pub.ro)

s.l. dr. ing. Monica Drăgoicea

Universitatea Politehnica București  
E-mail: [ma-dragoicea@ics.pub.ro](mailto:ma-dragoicea@ics.pub.ro)

**Rezumat:** În această lucrare, se prezintă o nouă arhitectură de rețea recurentă neuro - fuzzy, pentru a construi modele de predicție pe un orizont extins, în vederea proiectării unor structuri de conducere predictivă, a proceselor neliniare.

În fiecare regiune din domeniul de operare, se identifică un model liniar local, de ordin redus. Modelul global se obține prin interpolarea modelelor locale. În acest fel, se folosesc atât cunoștințele asupra procesului, care sunt utilizate pentru împărțirea în regiuni de operare fuzzy și pentru inițializarea ponderilor rețelei, cât și datele intrare-ieșire, utilizate pentru antrenare, în vederea minimizării erorii de predicție pe un orizont extins. Prin antrenare, se modifică funcțiile de apartenență a regiunilor de operare și sunt determinate modelele liniare locale. Legea de comandă utilizează funcțiile de apartenență, obținute în procesul de învățare.

Exemplificarea metodei propuse se face considerând un sistem de reglare a nivelului într-un rezervor conic.

**Cuvinte cheie:** rețele neuro - fuzzy, modelarea proceselor neliniare, conducere predictivă.

## 1. Introducere

În cadrul activităților de conducere, supervizare și monitorizare a proceselor, sunt solicitate, adesea, modele precise ale proceselor. Modelele proceselor pot fi împărțite în două categorii: modele determinate prin scrierea ecuațiilor ce guvernează desfășurarea procesului și modele empirice.

Construirea modelelor din prima categorie este dificilă și, în plus, și mare consumatoare de timp, în special pentru cazul proceselor complexe și al celor insuficient cunoscute.

Pentru aceste situații, se pot construi modele empirice, pe baza datelor intrare-ieșire din proces. Rețelele neurale au demonstrat o bună capacitate de aproximare a funcțiilor neliniare și au fost aplicate cu succes în modelarea proceselor [6], [7].

O rețea neurală poate învăța comportarea procesului, pe baza datelor intrare-ieșire din proces, utilizând un algoritm de învățare supervizat. În urma procesului de antrenare, rezultă un model de tip „cutie neagră”. Limitările acestui tip de model sunt date de dificultatea de interpretare a modelului rezultat și de robustețea scăzută pentru situațiile în care datele intrare-ieșire din proces nu sunt prezente în setul de antrenare.

O modalitate de a îmbunătăți robustețea modelului și de a deschide „cutia neagră” se bazează pe utilizarea combinată a cunoștințelor apriori asupra procesului și, respectiv, a datelor intrare-ieșire din proces. Cunoștințele asupra procesului pot fi utilizate, spre exemplu, pentru descompunerea domeniului de operare într-un număr de regiuni astfel încât, pentru fiecare regiune, comportarea locală a procesului să poată fi descrisă printr-un model liniar, de ordin redus.

Un proces neliniar poate fi liniarizat local, în jurul unui punct de funcționare, iar modelul obținut se consideră valid într-o întregă regiune din jurul acestui punct. Delimitarea regiunilor de operare este, adesea, imprecisă și există suprapuneri între diferitele regiuni astfel încât, definirea acestora se poate realiza într-o manieră fuzzy, conducând la o modelare fuzzy.

Pentru fiecare intrare a modelului, sunt asociate câteva mulțimi fuzzy, cu definirea corespunzătoare a funcțiilor de apartenență. Prin combinarea acestor intrări fuzzy, spațiul de intrare este împărțit în regiuni fuzzy. În fiecare regiune, este utilizat un model liniar local. Modelul global este obținut prin defuzificare utilizând, spre exemplu, metoda centrelor de greutate, care, în esență, realizează interpolarea ieșirilor modelelor locale. Se realizează, astfel, descompunerea unor procese complexe în subprocese de complexitate redusă.

Raționamentul fuzzy are abilitatea de a trata o informație imprecisă și incertă, în timp ce rețelele neurale pot fi identificate utilizând măsurători ale semnalelor de intrare și ieșire din proces. Rețelele neuro-fuzzy combină avantajele raționamentului fuzzy și ale rețelelor neurale. Cunoștințele asupra procesului pot fi, de asemenea, incluse în rețele neuro-fuzzy atât în cadrul partiționării fuzzy, cât și în cadrul modelelor locale, de ordin redus. Multe din structurile pentru modelare neuro-fuzzy, propuse în literatură, reprezintă modele de predicție cu un pas în care ieșirea curentă a procesului este utilizată ca intrare a rețelei pentru a predicta ieșirea la următorul pas.

În unele structuri de conducere a proceselor, este necesară o predicție pe un orizont extins.

Modelele neliniare, de predicție multipas, pot fi realizate utilizând rețele neurale dinamice.

În această lucrare, se prezintă o arhitectură neurofuzzy recurentă, care permite construirea unui model global neliniar, de predicție multipas, rezultat din conjuncția fuzzy a modelelor dinamice locale. Structura asigură predicții corespunzătoare, pe un orizont extins de timp. Prin antrenarea rețelei, se determină funcțiile de apartenență ale regimurilor locale și modelele asociate acestora. Obiectivul antrenării este acela de a minimiza eroarea de predicție pe un orizont extins. Rețeaua corect antrenată poate furniza cu precizie predicțiile pe un orizont extins ale ieșirii. Pe baza modelelor liniare locale, conținute în rețeaua recurentă neurofuzzy, se pot proiecta legi locale, de comandă liniară predictivă. Aceste legi locale pot fi combinate prin defuzzificare pentru a realiza o lege de comandă neliniară predictivă, pe orizont extins.

Organizarea lucrării este următoarea. Secțiunea 2 descrie modelarea fuzzy a proceselor neliniare. Se prezintă tipul de arhitectură de rețea recurentă neuro-fuzzy și strategiile de antrenare. În secțiunea 3, se prezintă proiectarea legii de comandă predictivă.

Secțiunea 4 este dedicată aplicării arhitecturii propuse de rețeaua recurentă neuro-fuzzy, pentru reglarea nivelului într-un rezervor conic. În încheiere, sunt prezentate concluziile lucrării.

## 2. Modelarea proceselor neliniare, apelând la rețele neurofuzzy

Domeniul de operare globală a unui proces neliniar este împărțit în mai multe regiuni locale de operare. În cadrul fiecărei regiuni, comportarea procesului este reprezentată printr-un model liniar de tip ARX - AutoRegressive with eXogenous inputs.

Definirea regiunilor de operare este realizată utilizând mulțimi fuzzy. Modelul dinamic fuzzy al procesului neliniar poate fi descris sub forma

Ri: IF condiție de operare  $i$  THEN

$$\hat{y}_i(t) = \sum_{j=1}^{ny} a_{ij} \hat{y}_i(t-j) + \sum_{j=1}^{nu} b_{ij} u(t-j) \quad (i=1, 2, \dots, nr) \quad (1)$$

Ieșirea modelului final se obține prin defuzzificare, prin metoda centrelor de greutate.

$$\hat{y}(t) = \left( \sum_{i=1}^{nr} \mu_i \hat{y}_i(t) \right) / \left( \sum_{i=1}^{nr} \mu_i \right) \quad (2)$$

În modelul de mai sus,  $\hat{y}$  reprezintă ieșirea procesului,  $u$  intrarea,  $\hat{y}_i$  reprezintă predicția ieșirii procesului pentru regiunea de operare  $i$ ,  $nr$  numărul regiunilor de operare fuzzy, nu și  $ny$  sunt întârzierile intrării și ieșirii,  $\mu_i$  funcția de apartenență pentru modelul  $i$ ,  $a_{ij}$  și  $b_{ij}$  - parametrii modelului local (ponderile stratului funcție),  $t$  - timpul discret.

Acest model este un model de predicție pe un orizont extins, întrucât, pentru predicție, se utilizează ieșirile anterioare ale modelului, și nu ieșirile anterioare ale procesului.

Modelele locale liniare pot fi transformate în forma incrementală.

$$\hat{y}(t) = (a_1 + 1)y(t-1) + (a_2 - a_1)y(t-2) + \dots + (a_{ny} - a_{ny-1})y(t-ny) + a_{ny}y(t-ny-1) + b_1 \Delta u(t-1) + b_2 \Delta u(t-2) + \dots + b_{nu} \Delta u(t-nu) \quad (3)$$

unde  $\Delta u(t-1) = u(t-1) - u(t-2)$ . Avantajul acestui model este eliminarea erorii staționare.

Modalitatea fuzzy, de reprezentare a regiunilor de operare, poate trata informația imprecisă și incertă. Modelarea fuzzy este potrivită pentru clase de procese a căror desfășurare implică mai multe regiuni de operare. Ca exemplu, sunt procesele de tip batch, care prezintă dinamici diferite, la diferite stadii de încărcare. Alt exemplu îl oferă procesele care produc ieșiri cu diferite grade de calitate.

În interiorul unei regiuni de operare, dinamica procesului poate fi reprezentată de un model simplu, de exemplu, un model liniar de ordin redus. În diferite regiuni de operare, modelele locale pot fi foarte diferite. Prin modelarea fuzzy se poate dezvolta un model global, în termenii unor modele locale simple.

Modelele de predicție fuzzy pe orizont extins pot fi realizate utilizând rețele recurente neuro-fuzzy. Arhitectura propusă în această lucrare reprezintă o modificare a rețelei structurale neurofuzzy, de tip feedforward.

Arhitectura rețelei conține cinci straturi: strat de intrare, strat de fuzzificare, strat regulă, strat funcție, strat de defuzzificare. Intrările în stratul de fuzzificare sunt variabilele din proces, utilizate pentru definirea regiunilor de operare fuzzy.

Fiecare din aceste variabile este transformată în câteva mulțimi fuzzy, în stratul de fuzzificare, unde fiecare neuron corespunde la o mulțime fuzzy, cu funcția de apartenență dată de ieșirea neuronului. Funcțiile de apartenență corespunzătoare pot fi obținute prin acordarea ponderilor din stratul de fuzzificare. Fiecare neuron din stratul regulă corespunde unei regiuni de operare fuzzy, a procesului. Intrările sunt mulțimile fuzzy care determină acea regiune de operare. Ieșirea este produsul dintre intrări și funcția de apartenență corespunzătoare. Neuronii din acest strat implementează o relație de intersecție de tip fuzzy.

Construcția stratului regulă se bazează pe unele cunoștințe asupra procesului: cunoașterea numărului de regiuni de operare și modul de stabilire a fiecărei regiuni.

Neuronii din stratul funcție implementează modelele liniare locale, în regiunile de operare fuzzy. Fiecare neuron este liniar și corespunde unei anumite regiuni de operare. Ieșirea este suma dintre intrările ponderate și o polarizare, ce reprezintă termenul constant al unui model local. Ponderile din stratul funcție reprezintă parametrii modelului.

Stratul defuzzificare efectuează operația de defuzzificare și furnizează ieșirea finală a rețelei. Intrările într-un neuron al acestui strat sunt funcțiile de apartenență ale regiunilor de operare fuzzy și ieșirile modelelor locale, în aceste regiuni.

Dintre metodele de antrenare a rețelelor recurente neuro-fuzzy, pot fi amintite următoarele: metode de tip backpropagation, metode de gradient conjugat, metode de optimizare Levenberg–Marquardt.

Funcția obiectiv în procesul de antrenare poate fi definită sub forma:

$$J = \frac{1}{N} \sum_{t=1}^N (\hat{y}(t) - y(t))^2 + \lambda \|w\|^2 \quad (4)$$

unde  $N$  reprezintă numărul datelor,  $\hat{y}$  predicția rețelei,  $y(t)$  valoarea țintă,  $t$  - timp discret,  $w$  este vectorul ponderilor rețelei,  $\lambda$  - parametru de regularizare.

Setul de date, pentru construirea rețelei recurente neuro-fuzzy, este împărțit într-un set de antrenare și un set de testare. Pentru modele liniare, minimizarea din (4) conduce la formula de regresie de tip ridge. Tehnicile de regularizare sunt larg utilizate pentru construirea modelelor statistice și au fost dezvoltate tehnici precum ridge regression, principal component regression, și partial least squares regression.

În algoritmul de antrenare de tip Levenberg–Marquardt, actualizarea ponderilor rețelei se face conform relației:

$$\Delta w(k+1) = -\eta \left( \frac{1}{N} \sum_{t=1}^N \frac{\partial \hat{y}(t)}{\partial w(k)} \left( \frac{\partial \hat{y}(t)}{\partial w(k)} \right)^T + \delta I \right)^{-1} \frac{\partial J}{\partial w(k)} \quad (5)$$

O valoare mare pentru  $\delta$  produce un pas redus, în direcția gradientului, în timp ce o valoare redusă produce un pas de căutare similar metodei Gauss-Newton. Ca și în alte metode de antrenare pe bază de gradient, antrenarea este încheiată atunci când eroarea de gradient se încadrează într-o anumită toleranță. Antrenarea se poate încheia și pe baza unui criteriu de cross validare.

Datorită recurenței prezente în cadrul rețelei, calculul gradientului necesită o metoda de tip „backpropagation through time”. Fenomenul de over-fitting este evitat prin utilizarea unui mecanism de oprire, pe bază de cross validare.

Punctul de oprire al antrenării este cel pentru care eroarea de testare este minimă. Pe durata antrenării, erorile de antrenare și de testare sunt permanent monitorizate. Validarea unui model bazat pe o rețea de tip neuro-fuzzy poate fi făcută pe baza unor teste de corelație, de ordin superior.

Dacă testele de corelație sunt îndeplinite, atunci reziduurile modelului reprezintă o secvență aleatoare, și nu pot fi predictate din intrările modelului. Dacă testele nu sunt trecute, înseamnă că modelul nu este adecvat. Se pot considera diferite structuri ale modelelor, cu un număr diferit de regiuni de operare locale și diferite ordine ale modelelor locale.

O manieră eficientă de construire a rețelei neuro-fuzzy este aceea de a începe cu un model simplu, cu un număr redus de regiuni de operare, și de a crește gradat complexitatea modelului.

### 3. Conducere predictivă utilizând modelul furnizat de rețeaua neurofuzzy

Legea de comandă pentru regulatorul global este obținută prin combinarea ieșirilor modelelor locale, într-o formă asemănătoare cu formula de defuzificare, pe baza centrelor de greutate. Modelele locale de tip ARX, conținute în rețeaua neurofuzzy, pot fi utilizate pentru a proiecta lege de comandă predictivă de tip generalizat Generalized Predictive Controllers (GPC) [3].

Avantajul acestei metode este acela că legea de comandă predictivă este construită pe baza unor legi de comandă predictivă, proiectate pe baza unor modele liniare locale, și care au soluții analitice. Astfel, se evită procedurile numerice de căutare și incertitudinile privind găsirea optimului global.

Regulatele liniare de tip GPC sunt obținute considerând modele liniare locale de forma:

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) \quad (6)$$

unde A și B sunt polinoame în operatorul de întârziere  $q^{-1}$ :

$$A(q^{-1}) = 1 + a_1q^{-1} + \dots + a_{ny}q^{-ny} \quad (7)$$

$$B(q^{-1}) = 1 + b_1q^{-1} + \dots + b_{nu}q^{-nu} \quad (8)$$

Pentru aflarea predictorului peste j-pași, se consideră următoarea ecuație diofantică:

$$1 = E_j(q^{-1})A(q^{-1})\Delta + q^{-j}F_j(q^{-1}) \quad (9)$$

unde  $E_j(q^{-1})$  și  $F_j(q^{-1})$  sunt polinoame în operatorul  $q^{-1}$ , iar  $\Delta$  este operatorul de diferențiere  $1 - q^{-1}$ .

Înmulțind ambii membri din (6) cu  $E_j(q^{-1})\Delta q^j$  se obține:

$$E_j(q^{-1})A(q^{-1})\Delta y(t+j) = E_j(q^{-1})B(q^{-1})\Delta u(t+j-1) \quad (10)$$

Înlocuind pe  $E_j(q^{-1})A(q^{-1})\Delta$  din (9), avem:

$$y(t+j) = E_j(q^{-1})B(q^{-1})\Delta u(t+j-1) + F_j(q^{-1})y(t) \quad (11)$$

Astfel, predicția ieșirii peste j pași este dată de:

$$\hat{y}(t+j|t) = G_j(q^{-1})\Delta u(t+j-1) + F_j(q^{-1})y(t) \quad (12)$$

unde  $G_j(q^{-1}) = E_j(q^{-1})B(q^{-1})$ . Ecuația diofantică poate fi rezolvată în mod recursiv [3]. Utilizând scrierea vectorială și notând cu N orizontul de predicție, se obțin următoarele relații:

$$\hat{Y} = GU + F \quad (13)$$

unde

$$\hat{Y} = [\hat{y}(t+1) \hat{y}(t+2) \dots \hat{y}(t+N)]^T \quad U = [\Delta u(t) \Delta u(t+1) \dots \Delta u(t+N-1)]^T \quad F = [f(t+1) f(t+2) \dots f(t+N)]^T$$

Notăm cu  $G$  matricea inferior triunghiulară, de dimensiune  $N \times N$ :

$$G = \begin{bmatrix} g_0 & 0 & \cdots & 0 \\ g_1 & g_0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ g_{N-1} & g_{N-2} & \cdots & g_0 \end{bmatrix} \quad (14)$$

Presupunând că evoluția referinței este cunoscută [ $w(t+j)$ ;  $j = 1, 2, \dots$ ], funcția obiectiv poate fi scrisă sub forma:

$$J = (\hat{Y} - W)^T (\hat{Y} - W) + \lambda U^T U \quad (15)$$

Prin minimizarea fără restricții a funcției obiectiv, se obține legea de comandă:

$$U = (G^T G + \lambda I)^{-1} G^T (W - F) \quad (16)$$

Deoarece primul element din  $U$  este  $\Delta u(t)$ , comanda curentă este dată de:

$$u(t) = u(t-1) + g^T (W - F) \quad (17)$$

unde  $g^T$  este prima linie din matricea  $(G^T G + \lambda I)^{-1} G^T$ . Legea de comandă conține acțiune integrală și determină anularea erorii în regim staționar.

După ce au fost calculate toate ieșirile reguletoarelor GPC, ele sunt combinate, utilizând metoda centrelor de greutate, pentru a obține legea globală de comandă:

$$u = \left( \sum_{i=1}^{nr} \mu_i u_i \right) / \left( \sum_{i=1}^{nr} \mu_i \right) \quad (18)$$

unde  $\mu_i$  reprezintă funcția de apartenență pentru regiunea de operare  $i$ , învățată pe durata antrenării rețelei,  $u_i$  sunt ieșirile reguletoarelor locale de tip GPC calculate anterior iar  $nr$  este numărul regiunilor de operare.

#### 4. Studiu de caz. Reglarea nivelului într-un rezervor conic

Pentru exemplificarea metodei propuse, se consideră un sistem de reglare a nivelului într-un rezervor conic. Scrierea ecuațiilor ce descriu funcționarea procesului permite obținerea următorului model:

$$\frac{dh}{dt} = \frac{Q_i - k\sqrt{h}}{\pi \left( r^2 + \frac{2rh}{\tan \theta} + \frac{h^2}{(\tan \theta)^2} \right)} \quad (19)$$

unde semnificația parametrilor este următoarea:  $h$  - nivelul lichidului din rezervor,  $Q_i$  - debitul de intrare,  $r$  - raza bazei rezervorului,  $\theta$  - unghiul de înclinare a pereților rezervorului față de orizontală. Între nivelul lichidului din rezervor și debitul de intrare există o dependență neliniară. Amplificarea statică depinde de nivelul din rezervor: la valori ridicate ale nivelului corespund amplificări mari. Datorită formei conice a rezervorului, constanta de timp a procesului crește o dată cu creșterea nivelului de lichid. Caracteristicile statice și dinamice ale procesului se modifică în funcție de condițiile de operare. Datele intrare-ieșire din proces au fost obținute prin simulare, considerând următoarele valori  $r = 15$  cm,  $k = 32.78$ ,  $\theta = 60$ . Domeniul de operare se consideră împărțit în trei regiuni, nivel mic, mediu, mare. Efectele zgomotului de măsură au fost simulate prin introducerea unui zgomot de tip  $N(0,0.6)$ , la ieșirea procesului.

După antrenare, modelul furnizat de rețeaua neurofuzzy este:

IF nivel este mic THEN  $\hat{y}(t) = -0.1314\hat{y}(t-1) - 0.0675\hat{y}(t-2) + 0.027u(t-1) - 0.0028u(t-2) + 1.237$

IF nivel este mediu THEN  $\hat{y}(t) = -0.1639\hat{y}(t-1) + 0.3881\hat{y}(t-2) + 0.0231u(t-1) + 0.0285u(t-2) - 1.487$

IF nivel este mare THEN  $\hat{y}(t) = -0.1730\hat{y}(t-1) + 0.7123\hat{y}(t-2) + 0.0241u(t-1) + 0.0285u(t-2) - 0.723$ .

Pentru comparație, a fost identificat și un model liniar:

$$y(t) = 0.4622\hat{y}(t-1) + 0.5841\hat{y}(t-2) + 0.0039u(t-1) + 0.0085u(t-2) - 0.379$$

Generarea datelor pentru antrenare, testare, validare, este realizată prin suprapunerea unui semnal aleator, peste mărimea de intrare, în timp ce celelalte mărimi sunt menținute constante. Au fost generate trei seturi de date, pentru antrenare, testare și validare. Valorile ieșirii au fost contaminate de zgomot aleator. Pe baza caracteristicilor procesului, regiunea de operare a fost împărțită în trei regiuni locale, având nivel mic, mediu și mare.

Ponderile pentru stratul funcție au fost inițializate cu numere aleatoare, în intervalul (-0.2, 0.2).

Au fost proiectate trei regulatoare de tip GPC. Valorile pentru orizonturile de comandă și predicție au fost alese 1 și respectiv 7. Aceleași valori pentru regulatorul GPC, proiectat pe baza modelului liniar. În figura 1, se prezintă performanțele de urmărire pentru un domeniu de operare 0-30 cm. Răspunsul regulatorului neurofuzzy este satisfăcător pentru toate regiunile de operare. Pentru studierea comportării la perturbații, s-a considerat că debitul de ieșire este redus la jumătate pentru intervalul 1000-3000 secunde. Și în acest caz, legea de comandă pe baza rețelei neurofuzzy produce performanțe superioare, așa cum rezultă din figura 2.

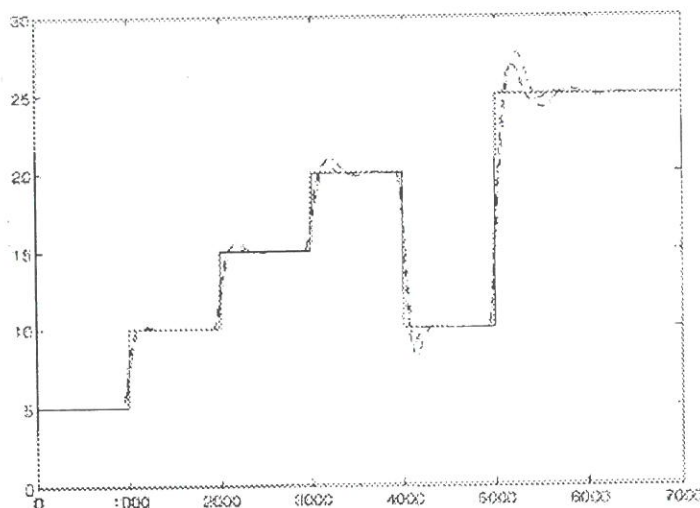


Figura 1. Performanțele de urmărire referință

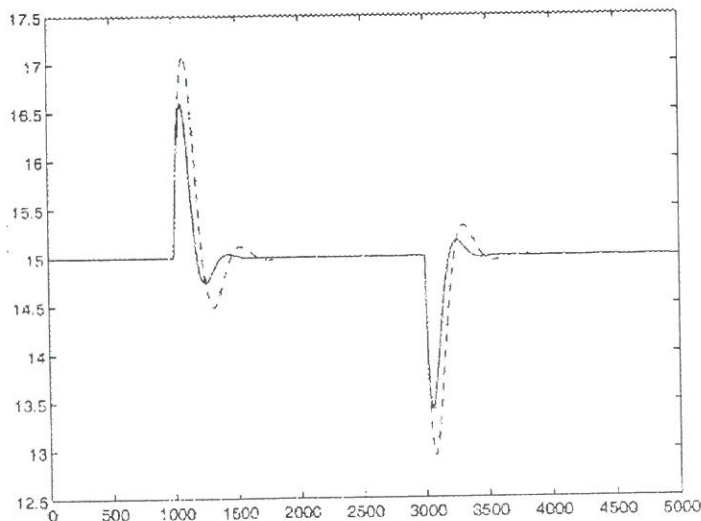


Figura 2. Comportarea la perturbații

Inițial, au fost alese modele locale de ordinul doi. Dacă eroarea nu se încadrează în limitele acceptate, se crește ordinul modelului și/sau a regiunilor fuzzy de operare. Rețeaua a fost antrenată utilizând algoritmul Levenberg–Marquardt, în combinație cu tehnici de regularizare, și un criteriu de terminare, utilizând metode de cross-validare.

## 5. Concluzii

În această lucrare, arhitecturile de tip neuro-fuzzy sunt utilizate pentru a construi modele de predicție, pe orizont extins, pentru procese neliniare. Modelele rezultate sunt mai ușor de interpretat decât modelele bazate pe rețele neurale. Ponderile din stratul de fuzzificare reprezintă funcțiile de apartenență ale regiunilor fuzzy, iar ponderile din stratul funcție determină modelele locale în aceste regiuni.

Informația asupra procesului poate fi utilizată pentru inițializarea rețelei: numărul regiunilor de operare, definiția lor, funcțiile de apartenență.

Modelul realizat pe baza rețelelor neuro-fuzzy constituie o bază pentru un nou tip de regulator neliniar, compus din mai multe regulatoare locale liniare.

Valoarea comenzii este calculată prin combinarea comenzilor locale, pe baza funcțiilor de apartenență. Un regulator predictiv, proiectat pe bază de model realizat cu rețea neuro-fuzzy, poate fi obținut prin combinarea unor regulatoare predictive locale, bazate pe modele liniare locale, care admit soluții analitice. Se evită, astfel, dezavantajele prezente în conducerea predictivă neliniară, metodele de optimizare și incertitudinea privind convergența la un optim global.

## Bibliografie

1. **ASTROM, K. J., B. WITTENMARK:** Adaptive Control, 2nd ed. Reading, MA: Addison-Wesley, 1989.
2. **BROWN, M., C. J. HARRIS:** Neurofuzzy Adaptive Modelling and Control, Englewood Cliffs, NJ: Prentice-Hall, 1994.
3. **CLARKE, D. W., C. MOHTADI, P. S. TUFFS:** Generalized Predictive Control, Parts 1 and 2. În: Automatica, vol. 23, 1987, pp. 859–875.
4. **CONSTANTIN, N., M. DRAGOICEA:** Regularization Effect in Modelling based on Local Model Network. În: Proc. 11<sup>th</sup> Int. Symp. On Modelling, and System's Identif., 2001, pp. 95-100.
5. **DUMITRACHE, I., N. CONSTANTIN, M. DRAGOICEA:** Rețele neurale. Identificarea și conducerea proceselor, Ed. MatrixRom, București, 1999, 320 p.
6. **GIROSI, F., T. POGGIO:** Networks and the Best Approximation Property. În: Biol. Cybern., vol. 63, 1990, pp. 169–179.
7. **MORRIS, A. J., G. A. MONTAGUE, M. J. WILLIS:** Artificial Neural Networks: Studies in Process Modelling and Control. În: Trans. IChemE, vol. 72, 1994, pt. A, pp. 3–19.
8. **NIE, J., D. A. LINKENS:** Learning Control Using Fuzzified Self-organizing Radial Basis Function Networks. În: IEEE Trans. Fuzzy Syst., vol. 1, 1993, pp. 280–287.
9. **TAKAGI, T., M. SUGENO:** Fuzzy Identification of Systems and Its Application to Modeling and Control. În: IEEE Trans. Syst., Man, Cybern., vol. SMC-15, 1985, pp. 116–132.
10. **WANG, L. X.:** Adaptive Fuzzy Systems and Control: Design and Stability Analysis, Englewood Cliffs, NJ: Prentice-Hall, 1994.
11. **ZHANG, J., A. J. MORRIS:** On-line Process Fault Diagnosis Using Fuzzy Neural Networks. În: Intell. Syst. Eng., vol. 3, 1994, pp. 37–47.