

Raport de Cercetare

Grant: CNCSIS 517 Tema 2

Autori: Sl. Dr. Ing. Georgeta Budura, Sl. Dr. Ing. Corina Botoca

Universitatea: "Politehnica" Timișoara

IDENTIFICAREA ȘI COMPENSAREA NELINIARITĂȚILOR NEDORITE ÎN CANALELE DE TELECOMUNICAȚII

CAPITOLUL I

INTRODUCERE

1.1 Aspecte privind necesitatea modelării neliniare

Domeniul telecomunicațiilor a cunoscut în ultimele decenii o evoluție fără precedent în cercetarea științifică și tehnologică relevând probleme noi complexe pentru care tehnicile liniare au eșuat în a furniza soluții acceptabile.

Un exemplu în acest sens este oferit de încercarea de a modela sau controla fenomene neliniare cu metode de procesare liniară.

În acest context, analiza sistemelor neliniare s-a dezvoltat major, încercându-se închegarea unei teorii care să ofere un model matematic de abordare general, capabil să trateze unitar o clasă cât mai largă de tipuri de neliniarități. Cu toate acestea, nu putem vorbi la ora actuală de o teorie unitară în ceea ce privește sistemele neliniare. Există câteva modele intens utilizate. Dezvoltarea acestora s-a făcut în special în strânsă legătură cu aplicațiile care le-au generat. Ulterior însă s-a realizat aplicarea cu succes a acestora și în alte domenii, ceea ce a contribuit la evidențierea calităților modelului considerat, dar și a unor deficiențe. S-au scris numeroase cărți și articole raportând rezultate științifice din domenii de cercetare variate: biologie, chimie, medicină, telecomunicații. Sistemele neliniare s-au dovedit a manifesta fenomene complexe, surprinzătoare și de cele mai multe ori imposibil de prezis pentru un cercetător învățat a manevra doar tehnici liniare. Cele mai frecvente aplicații din domeniul telecomunicațiilor care necesită utilizarea modelelor neliniare sunt:

- Studiul sistemelor care prezintă neliniarități de tip saturație în legătura intrare-ieșire. În această categorie sunt incluse: - legăturile de transmisie prin sateliți care conțin amplificatoare de semnal ce lucrează în regim de saturație.
- Canalele de comunicații ce operează la viteze mai mari de 48000 biți/sec, realizate pe cablu coaxial prezintă de asemenea o comportare neliniară de tip saturație. S-a demonstrat experimental că probabilitatea de apariție a erorii în aceste sisteme de transmisie a datelor este datorată în mare măsură distorsiunilor neliniare ce nu pot fi înlăturate de egalizoarele liniare. Numeroase cercetări au fost dedicate implementării egalizoarelor de canal bazate pe structuri neliniare.
- Problema anulării ecoului în telecomunicații a dezvoltat necesitatea utilizării modelelor neliniare și în acest domeniu. Un număr mare de aplicații abordează reducerea adaptivă a ecoului cu ajutorul structurilor neliniare.

Sunt cunoscute la ora actuală soluțiile oferite de egalizoarele neliniare pe canalele de comunicații de mare viteză, utilizarea structurilor neliniare în eliminarea ecoului acustic și electric, a eliminării zgomotului, în prelucrarea imaginii, în modelarea legăturilor prin sateliți precum și în caracterizarea dispozitivelor semiconductoare. Cu toate acestea domeniul este într-o continuă dinamică datorită apariției și dezvoltării rețelelor moderne de telecomunicații, a apariției de numeroase echipamente cum sunt cele legate de telefonia mobilă. La ora actuală se caută soluții în ceea ce privește eliminarea neliniarităților nedorite pe liniile de comunicații de mare viteză cum sunt legăturile radio numerice, a ecoului generat într-o comunicație fără fir sau a celui introdus de dispozitivul „hands-free” utilizat tot mai mult de către posesorii de telefoane mobile. Se încearcă de asemenea găsirea unor soluții în ceea ce privește modelarea și predicția semnalului pentru eliminarea interferențelor intersimbol în comunicațiile de mare viteză.

Principala problemă a introducerii structurilor de prelucrare neliniară o constituie complexitatea implementării. Acest dezavantaj a devenit mai puțin sesizabil odată cu dezvoltarea

tehnologiilor VLSI care au permis nu numai posibilitatea prelucrării în timp real dar și un preț de cost scăzut al acestor filtre.

Pasul natural spre implementarea structurilor de prelucrare neliniară de mare complexitate l-a constituit abordarea acelor filtre neliniare a căror structură se bazează pe o relație liniară în raport cu parametrii. Introducerea neliniarităților în modul de operare al unui astfel de filtru este reprezentată de transformările neliniare care apar la nivelul semnalului de intrare în calculul răspunsului unui astfel de filtru.

În cadrul lucrării de față se vor aborda două tipuri de filtre neliniare a căror structură se bazează pe o relație liniară în raport cu parametrii și anume; filtrele Volterra și filtrele bazate pe rețele neuronale de tip RBF (Radial Basis Functions). Ambele tipuri de filtre cunosc la ora actuală o largă utilizare în domeniul telecomunicațiilor.

1.2 Filtre Volterra

1.2.1 Generalități

Conform celor prezentate mai sus, în evoluția tehnicilor de filtrare neliniară, ceea ce a lipsit o lungă perioadă de timp a fost o teorie care să ocupe un loc central, raportat la cele 2 caracteristici importante, generalitate și aplicabilitate și care să servească ca punct de plecare atât pentru studiul matematic cât și pentru dezvoltarea tehnicilor ingineresti.

Începutul anilor '70 a constituit punctul de cristalizare a tuturor acestor încercări în ceea ce avea să se constituie drept teoria sistemelor neliniare. În formularea acestei teorii s-au avut în vedere aspecte legate de modelarea cu ajutorul ecuațiilor diferențiale, reprezentarea cu ajutorul transformatorilor precum și descrierea cu ajutorul operatorilor matematici a fenomenelor neliniare.

Deasemenea s-a avut permanent în vedere posibilitatea extinderii tehnicilor liniare în domeniul sistemelor neliniare.

O contribuție importantă privind modelarea fenomenelor neliniare a constituit-o reprezentarea acestora cu ajutorul seriilor Volterra și aparține lui Fréchet. Tot el a fost cel care a creat cadrul teoretic ce permite reprezentarea sistemelor neliniare fără memorie, de natură nepolinomială, prin modele polinomiale bazate pe funcționale, pentru un anumit domeniu al semnalului de intrare.

Potrivit importanței bibliografii a lui Schetzen [M.Schetzen, 89], Wiener a fost primul care a aplicat modelul Volterra în analiza unui sistem neliniar în anul 1942.

Metodele privind măsurarea nucleelor Volterra atașate unui sistem neliniar au fost publicate de către Schetzen în 1965 și s-au dovedit deosebit de dificile datorită cuplărilor între nuclee. Acesta a constituit unul din principalele motive care l-au determinat pe Wiener să abordeze o reprezentare ortogonală a sistemelor neliniare cu memorie precum și dezvoltarea unor metode pentru măsurarea nucleelor Wiener.

O importantă parte în ceea ce privește modelarea prin intermediul modelelor Volterra și Wiener o constituie exactitatea măsurării coeficienților sau a nucleelor modelului. Introducerea modelului Wiener a constituit un important pas legat de măsurarea nucleului. Ortogonalitatea funcționalelor Wiener în cazul unui semnal de intrare gaussian a permis determinarea nucleelor Wiener asociate sistemului neliniar, prin utilizarea tehnicilor bazate pe intercorelația dintre semnalul de intrare și cel de ieșire al sistemului.

Mai târziu, Schetzen a generalizat teoria Wiener extinzând-o în cazul intrărilor gaussiene colorate și elaborând metode de măsurare ale nucleelor în cazul unor astfel de semnale de intrare. Schetzen a fost cel care a introdus și dezvoltat teoria sistemului Volterra invers de un anumit ordin și ulterior a publicat o carte privind aplicarea teoriilor Volterra și Wiener în cazul sistemelor neliniare.

Modelele Volterra și Wiener au fost aplicate în numeroase domenii cum ar fi:

- Modelarea sistemelor biologice;
- Studiul interacțiunilor dintre valurile oceanice și platformele maritime;
- Legăturile digitale prin sateliți, în care amplificatoarele de semnal operează în apropierea punctului de saturație, prezentând caracteristici puternic neliniare;
- Canalele de comunicații de înaltă viteză (peste 4800bps) care prezintă neliniarități de tip saturație. S-a demonstrat experimental că probabilitatea de apariție a erorii în aceste sisteme de transmisie a datelor este datorată în mare măsură distorsiunilor neliniare, distorsiuni ce nu pot fi înlăturate de egalizoare liniare.
- Compensarea amplificatoarelor de putere;
- Compensarea neliniarității difuzoarelor.

Modelul Volterra prezintă totuși o complexitate sporită, după cum se va evidenția în lucrarea de față, în special datorită numărului mare de calcule, care crește rapid odată cu creșterea ordinului neliniarității și cu dimensiunea memoriei utilizate. Astfel complexitatea modelului mărește timpul

necesar estimării nucleului și timpul necesar realizării unor simulări utilizând acest model. Afectate de această complexitate sunt de asemenea viteza și complexitatea hardware necesare implementării și compensării sistemului.

1.2.2 Operatori Volterra

Un sistem neliniar continuu în timp, fără memorie, poate fi descris cu anumite restricții, prin intermediul unei serii Taylor.

În cazul sistemelor neliniare invariante în timp, cu memorie, legătura ieșire-intrare se poate exprima generalizând dezvoltarea în serie Taylor prin aplicarea ei unei funcții de mai multe variabile. Relația intrare ieșire în cazul acestor sisteme, cunoscută sub numele de serie Volterra este de forma [M.Schetzen, 89]:

$y(t) = h_0 + \int_{-\infty}^{+\infty} h_1(\tau_1)x(t-\tau_1)d\tau_1 + \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} h_2(\tau_1, \tau_2)x(t-\tau_1)x(t-\tau_2)d\tau_1 d\tau_2 + \dots$ $+ \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} h_p(\tau_1, \dots, \tau_p)x(t-\tau_1)\dots x(t-\tau_p)d\tau_1 \dots d\tau_p$	(1.1)
--	-------

unde $x(t)$ și $y(t)$ reprezintă semnalul de intrare, respectiv cel de ieșire la momentul t iar $h_p(\tau_1, \dots, \tau_p)$ este nucleul Volterra de ordin p .

Așa după cum rezultă din relația 1.1, sistemul neliniar este complet caracterizat de funcțiile multidimensionale $h_p(\tau_1, \dots, \tau_p)$, numite nuclee Volterra. Nucleul de ordin zero, h_0 , este o constantă, nucleul de ordinul întâi, $h_1(\tau_1)$, reprezintă răspunsul la impuls al sistemului, iar nucleele de ordin $p \geq 2$ sunt funcții simetrice în raport cu argumentele lor. Astfel, cele $p!$ permutări posibile ale variabilelor τ_1, \dots, τ_p nu modifică nucleul $h_p(\tau_1, \dots, \tau_p)$.

Se obișnuiește să se noteze integrala p -dimensională din relația 1.1 cu $H_p[x(t)]$, unde H_p este numit operator Volterra de ordin p . Relația 1.1 devine în acest caz:

$y(t) = h_0 + \sum_{p=1}^{\infty} H_p[x(t)]$	(1.2)
--	-------

Fréchet, care s-a ocupat cu studiul funcționalelor continue, a dat o fundamentare matematică riguroasă seriilor Volterra arătând că orice funcțională continuă poate fi reprezentată printr-o serie de funcționale de ordin întreg. Această serie este convergentă uniform pe orice set compact de funcții continue. O funcțională de ordin întreg este echivalentă cu o funcțională Volterra.

Rezultatul obținut de Fréchet se datorează generalizării teoremei lui Weierstrass care afirmă că orice funcție continuă poate fi reprezentată printr-o serie polinomială care converge uniform pe orice interval închis, adică setul de funcții $\{f_n(x) = x^n\}$ este complet [M.Fréchet, 10].

Seriile Volterra au fost aplicate pentru prima dată în studiul sistemelor neliniare de către Norbert Wiener. Pentru un sistem cauzal limitele inferioare ale integralelor din relația 1.1 sunt puse pe zero, iar limitele superioare au o valoare finită, reprezentând memoria sistemului.

Pentru reprezentarea sistemelor reale se utilizează în general serii Volterra trunchiate:

$y(t) = h_0 + \sum_{p=1}^N H_p[x(t)]$	(1.3)
---------------------------------------	-------

În relația de mai sus N este numit, în general, ordinul sau gradul seriei.

Relația 1.1 poate fi interpretată ca o extensie a reprezentării integrale a operatorilor liniari, acest caz corespunzând primului termen al seriei.

Spre deosebire de cazul sistemelor liniare, complet caracterizate de funcția răspuns la impuls, în cazul sistemelor neliniare, reprezentate prin serii Volterra trunchiate, funcția răspuns la impuls nu oferă o caracterizare completă a acestora. Într-adevăr, răspunsul unui astfel de sistem la semnalul de intrare :

$x(t) = A\delta(t)$	(1.4)
---------------------	-------

este:

$h(t) = h_0 + Ah_1(t) + A^2h_2(t, t) + \dots + A^p h_p(t, \dots, t) \dots$	(1.5)
--	-------

După cum se poate observa din relația de mai sus răspunsul la impuls este determinat de valorile nucleelor în puncte situate pe diagonala nucleelor, adică la acele momente de timp pentru care: $t_1 = t_2 = \dots = t_p$. Sistemul neliniar se poate reprezenta prin operatorii neliniari omogeni H_p de nucleu $h_p(t_1, \dots, t_p)$ ca în figura 1.1.

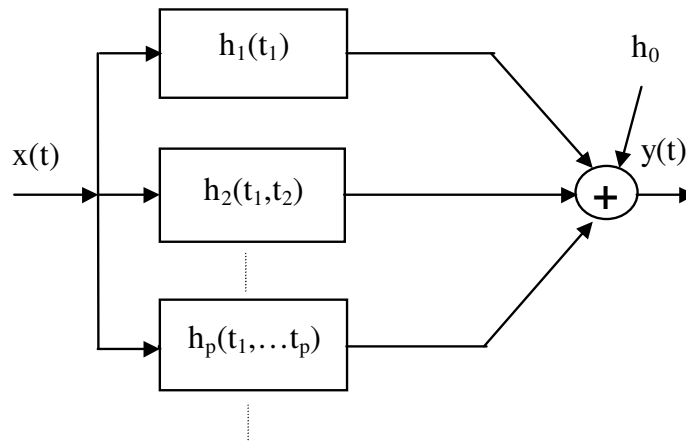


Figura 1.1 Reprezentarea unui sistem neliniar cu ajutorul operatorilor omogeni H_p

Un caz particular de operator neliniar se obține când semnalul de ieșire este exprimabil în serie de puteri:

$y(t) = \sum_{p=1}^{\infty} \alpha_p x^p(t)$	(1.6)
--	-------

În acest caz nucleele Volterra se pot scrie sub forma simplificată:

$h_p(t_1, \dots, t_p) = \alpha_p \delta(t_1) \times \dots \times \delta(t_p)$	(1.7)
---	-------

și sunt caracteristice unor operatori neliniari fără memorie. În relația 1.7 operatorul notat "X" reprezintă produsul direct.

O altă categorie de operatori neliniari se poate evidenția dacă considerăm cazul sistemelor neliniare ce pot fi modelate prin legarea în cascadă a două sau trei sisteme așa după cum se indică în figura 1.2.

Cele trei sisteme indicate în figura 1.2 trebuie să fie separabile.

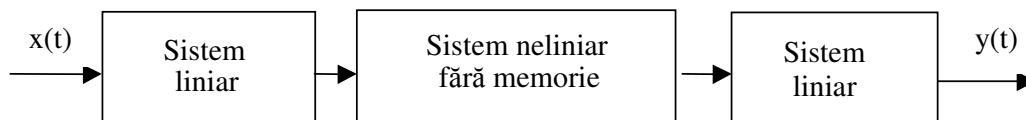


Figura 1.2 Structura unui sistem neliniar ce se poate reprezenta prin operatori liniari și neliniari separabili

Având în vedere cele prezentate mai sus se poate face o clasificare a operatorilor neliniari în trei categorii [38]:

- operatori neliniari fără memorie
- operatori neliniari separabili
- operatori neliniari neseparabili

Se va prezenta în continuare cea de-a doua categorie de operatori întrucât aceasta își găsește numeroase aplicații în modelarea sistemelor fizice. Pentru caracterizarea acestui tip de operatori M.Schetzen a dezvoltat teoria Volterra din punctul de vedere al operatorilor p -liniari [M.Schetzen, 89].

1.2.3 Operatori p-liniari

M. Schetzen și-a construit teoria de abordare a sistemelor neliniare cu ajutorul operatorilor p-liniari. Pentru simplitate se va prezenta în continuare cazul operatorului de ordinul al II-lea, bilinear ($p=2$).

Fie $x(t)$ un semnal de intrare de forma:

$$x(t) = \sum_{n=1}^N c_n x_n(t) \quad (1.8)$$

Un operator bilinear T_2 se definește ca fiind un operator al cărui răspuns la o combinație liniară de semnale de intrare (vezi rel.1.8) este o operație biliniară asupra fiecărui semnal de intrare, așa după cum se vede din relația 1.9.

$$y(t) = T_2[x(t)] = T_2 \left[\sum_{n=1}^N c_n x_n(t) \right] = \sum_{m=1}^N \sum_{n=1}^N T_2 \{c_m x_m(t) c_n x_n(t)\} = \sum_{m=1}^N \sum_{n=1}^N c_m c_n T_2 \{x_m(t), x_n(t)\} \quad (1.9)$$

Operatorul $T_2\{\}$ este numit bilinear întrucât este linear în raport cu x_m pentru un anumit x_n și este linear în raport cu x_n pentru un anumit x_m dat.

Cel mai simplu sistem de ordinul al doilea îl constituie sistemul a cărui relație ieșire-intrare este exprimabilă prin relația: $y(t) = x^2(t)$. Răspunsul acestui sistem la semnalul din relația 1.8 este dat de relația 1.10:

$$y(t) = \sum_{m=1}^N \sum_{n=1}^N c_m c_n x_m(t) x_n(t) \quad (1.10)$$

Expresia operatorului bilinear este în acest caz:

$$T_2 \{x_m(t), x_n(t)\} = x_m(t) x_n(t) \quad (1.11)$$

În cazul în care sistemul de ordinul al II-lea este și invariant în timp, $T_2[\]$ devine $H_2[\]$, numit operator Volterra de ordinul al II-lea.

$$y(t) = H_2[x(t)] \quad (1.12)$$

Descrierea funcțională a operatorului $H_2\{\}$ se poate obține în mod analog celei pentru operatorul linear și anume considerând pentru semnalul de intrare aproximarea:

$$x(t) = \lim_{\Delta \rightarrow 0} x_{\Delta}(t) \quad (1.13)$$

unde:

$$x_{\Delta}(t) = \sum_{k\Delta=-\infty}^{\infty} \Delta x(k\Delta) \delta(t - k\Delta) \quad (1.14)$$

Răspunsul sistemului Volterra de ordinul al II-lea este în acest caz:

$$y_{\Delta}(t) = H_2[x_{\Delta}(t)] = \sum_{m\Delta=-\infty}^{\infty} \sum_{n\Delta=-\infty}^{\infty} \Delta \cdot x(m\Delta) \Delta \cdot x(n\Delta) H_2 \{ \delta(t - m\Delta), \delta(t - n\Delta) \} \quad (1.15)$$

unde $H_2\{\}$ este operatorul bilinear invariant în timp. Funcția de timp corespunzătoare fiecărei operații biliniare este dată de relația:

$$h_2(t - m\Delta, t - n\Delta, \Delta) = H_2 \{ \delta(t - m\Delta), \delta(t - n\Delta) \} \quad (1.16)$$

Ținând cont de relația 1.16, y_{Δ} se poate exprima sub forma:

$$y_{\Delta}(t) = \sum_{m\Delta=-\infty}^{\infty} \sum_{n\Delta=-\infty}^{\infty} \Delta x(m\Delta) \Delta x(n\Delta) h_2(t - m\Delta, t - n\Delta, \Delta) \quad (1.17)$$

Când $\Delta \rightarrow 0$, $y_{\Delta}(t) \rightarrow y(t)$ și relația 1.17 devine:

$y(t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x(\sigma_1)x(\sigma_2)h_2(t - \sigma_1, t - \sigma_2)d\sigma_1d\sigma_2$	(1.18)
--	--------

unde s-au folosit notațiile:

$h_2(t - \sigma_1, t - \sigma_2) = \lim_{\Delta \rightarrow 0} h_2(t - m\Delta, t - n\Delta, \Delta) = H_2\{\delta(t - \sigma_1), \delta(t - \sigma_2)\}$	(1.19)
---	--------

Relația 1.19 constituie reprezentarea funcțională a unui operator Volterra. Făcând înlocuirile $\tau_1 = t - \sigma_1$ și $\tau_2 = t - \sigma_2$ se poate scrie relația sub forma unei integrale de convoluție bidimensionale:

$y(t) = H_2[x(t)] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h_2(\tau_1, \tau_2)x(t - \tau_1)x(t - \tau_2)d\tau_1d\tau_2$	(1.20)
--	--------

unde $h_2(\tau_1, \tau_2)$ reprezintă nucleul Volterra de ordinul al II-lea. Pentru un sistem de ordinul al II-lea se poate determina întotdeauna $h_2(\tau_1, \tau_2)$ ca o funcție simetrică în raport cu argumentele sale. Importanța existenței nucleului simetric pentru un sistem dat constă în faptul că el este unic.

Ca o ilustrare a celor de mai sus se determină în continuare nucleul Volterra de ordinul al-II-lea al sistemului din figura 1.3.

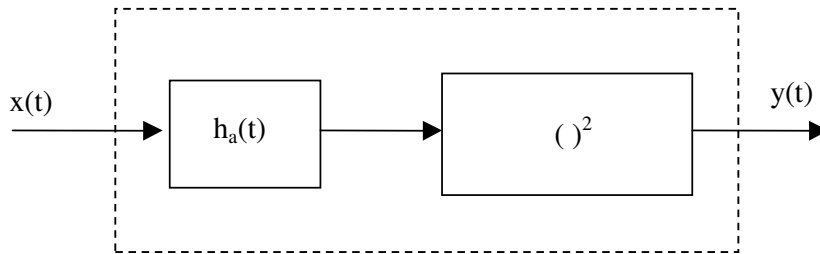


Figura 1.3 Sistem de ordinul al-II-lea

Răspunsul sistemului $y(t)$ e dat de relația:

$y(t) = \left[\int_{-\infty}^{\infty} h_a(\tau)x(t - \tau)d\tau \right]^2 = \left[\int_{-\infty}^{\infty} h_a(\tau_1)x(t - \tau_1)d\tau_1 \right] \left[\int_{-\infty}^{\infty} h_a(\tau_2)x(t - \tau_2)d\tau_2 \right] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h_a(\tau_1)h_a(\tau_2)x(t - \tau_1)x(t - \tau_2)d\tau_1d\tau_2$	(1.21)
--	--------

Comparând rezultatul dat de 1.21 cu cel general exprimat de relația 1.20 se obține:

$h_2(\tau_1, \tau_2) = h_a(\tau_1)h_a(\tau_2)$	(1.22)
--	--------

De exemplu, în cazul în care $h_a(t)$ este de forma:

$h_a(t) = Ee^{-at}\sigma(t)$	(1.23)
------------------------------	--------

nucleul Volterra de ordinul al-II-lea al sistemului neliniar devine:

$h_2(\tau_1, \tau_2) = E^2 e^{-a(\tau_1 + \tau_2)}\sigma(\tau_1)\sigma(\tau_2)$	(1.24)
---	--------

Așa după cum se observă din relația 2.24, $h_2(\tau_1, \tau_2)$ e diferit de zero numai în primul cadran al semiplanului $\tau_1 - \tau_2$ caracteristică specifică oricărui nucleu Volterra de ordinul al-II-lea cauzal.

Interpretarea fizică a nucleului atașat operatorului Volterra de ordinul al-II-lea

Nucleul Volterra ordinul al-II-lea $h_2(\tau_1, \tau_2)$ caracterizează operatorul Volterra $H_2[\cdot]$, întrucât permite determinarea răspunsului sistemului oricare ar fi semnalul de intrare. Se pune problema găsirii unei semnificații fizice funcției $h_2(\tau_1, \tau_2)$. Așa după cum se știe $h_1(\tau)$, nucleul de ordinul întâi atașat operatorului $H_1[\cdot]$, reprezintă răspunsul la impuls al acestui sistem.

Răspunsul la impuls al sistemului de ordinul al-II-lea poate fi dedus imediat pe baza relației 2.16 și este:

$$y(t) = H_2[\delta(t)] = h_2(t, t) \quad (1.25)$$

Relația 1.25 subliniază imposibilitatea caracterizării sistemului de ordinul al-II-lea prin răspunsul său la impuls.

Funcției $h_2(\tau_1, \tau_2)$ i se poate găsi o interpretare și anume aceea de răspuns al sistemului bilinear Volterra, caracterizat de operatorul, $H_2\{ \}$, la un semnal de intrare constând în două impulsuri Dirac, la momente de timp diferite. Pentru a demonstra această afirmație se consideră răspunsul $y(t)$ al unui sistem Volterra de ordinul al-II-lea la o sumă de semnale:

$$y(t) = H_2[x_1(t) + x_2(t)] = H_2[x_1(t)] + H_2\{x_1(t), x_2(t)\} + H_2[x_2(t)] \quad (1.26)$$

unde $H_2[]$ este operatorul Volterra caracterizat de 1.20, iar $H_2\{ \}$ este operatorul bilinear având reprezentarea funcțională:

$$H_2\{x_1(t), x_2(t)\} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h_2(\tau_1, \tau_2) x_1(t - \tau_1) x_2(t - \tau_2) d\tau_1 d\tau_2 \quad (1.27)$$

Dacă cele două semnale de intrare sunt: $x_1(t) = \delta(t - T_1)$; $x_2(t) = \delta(t - T_2)$, răspunsul sistemului bilinear devine:

$$H_2\{\delta(t - T_1), \delta(t - T_2)\} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h_2(\tau_1, \tau_2) \delta(t - T_1 - \tau_1) \delta(t - T_2 - \tau_2) d\tau_1 d\tau_2 = h_2(t - T_1, t - T_2) \quad (1.28)$$

ezultă că relația 1.28 oferă deasemenea posibilitatea măsurării nucleului Volterra $h_2(\tau_1, \tau_2)$, utilizând un sistem cu răspunsul $H_2\{x_1(t), x_2(t)\}$, la intrarea căruia se aduc semnalele $x_1(t) = \delta(t - T_1)$ și $x_2(t) = \delta(t - T_2)$.

Schema de măsură reprezentată în figura 1.4 are la bază relația existentă între operatorul bilinear Volterra de ordinul al-II-lea și operatorul Volterra de ordinul al-II-lea:

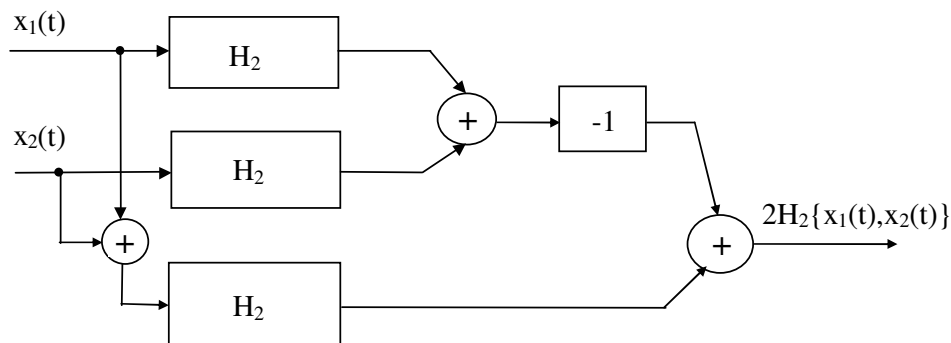


Figura 1.4 Schema bloc pentru sinteza operatorului Volterra bilinear conform cu rel. 1.29.

$$2H_2\{x_1(t), x_2(t)\} = H_2[x_1(t) + x_2(t)] - H_2[x_1(t)] - H_2[x_2(t)] \quad (1.29)$$

Conform schemei de sinteză a operatorului bilinear, dacă la intrare se aduc semnalele $x_1(t) = \delta(t - T_1)$ și $x_2(t) = \delta(t - T_2)$, răspunsul sistemului bilinear va fi:

$g_2(t) = 2H_2\{\delta(t - T_1), \delta(t - T_2)\} = 2h_2(t - T_1, t - T_2)$	(1.30)
--	--------

Funcția $g_2(t)$ reprezintă răspunsul la impuls bidimensional al sistemului de ordinul al-II-lea. Valoarea sa este dublul valorii nucleului de-a lungul unei linii înclinată la 45° în planul $\tau_1 - \tau_2$, așa după cum e indicat în figura 1.5.

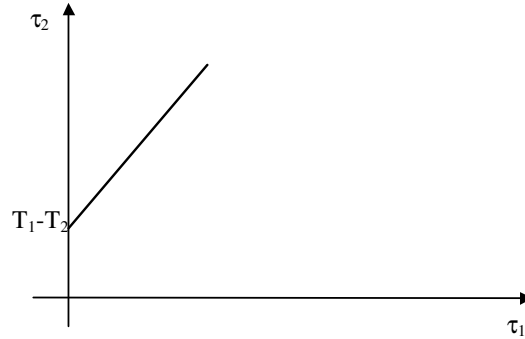


Figura 2.5 Graficul funcției $g_2(t)$

Pentru sistemul de ordinul al-II-lea, cu schema din figura 2.3, avem:

$g_2(t) = \begin{cases} E^2 e^{-a(T_1+T_2)} e^{-2at} & pt \quad t \geq \max(T_1, T_2) \\ 0 & \text{în rest} \end{cases}$	(1.31)
--	--------

Folosind schema din figura 1.4 nucleul de ordinul al-II-lea se poate determina măsurând valorile $g_2(t)$, obținute pentru diferite valori T_1, T_2 .

În mod asemănător introducerii operatorului de ordinul al-II-lea se poate proceda pentru a defini operatorul p -liniar:

$y_p(t) = H_p[x(t)]$	(1.32)
----------------------	--------

a cărei descriere funcțională are forma dată de relația 1.33:

$H_p[x(t)] = \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} h_p(\tau_1, \dots, \tau_p) x(t - \tau_1) \dots x(t - \tau_p) d\tau_1 \dots d\tau_p$	(1.33)
--	--------

În relația 1.33 $h_p(\tau_1, \dots, \tau_p)$ este o funcție simetrică în raport cu argumentele sale.

1.2.4 Transformata Fourier a nucleului de ordinul al-II-lea

Se definește transformata Fourier bidimensională a nucleului $h_2(\tau_1, \tau_2)$, sau simplu transformata nucleului, ca fiind:

$H_2(\omega_1, \omega_2) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} h_2(\tau_1, \tau_2) e^{-j(\omega_1\tau_1 + \omega_2\tau_2)} d\tau_1 d\tau_2$	(1.34)
---	--------

Condiția de existență a transformatei Fourier bidimensionale este asigurată de relația:

$\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} h_2(\tau_1, \tau_2) d\tau_1 d\tau_2 < \infty$	(1.35)
--	--------

Transformata inversă având ca rezultat $h_2(\tau_1, \tau_2)$ se deduce cu ajutorul relației:

$h_2(\tau_1, \tau_2) = \frac{1}{(2\pi)^2} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} H_2(\omega_1, \omega_2) e^{j(\omega_1\tau_1 + \omega_2\tau_2)} d\omega_1 d\omega_2$	(1.36)
---	--------

Utilizarea Transformatei în determinarea răspunsului unui sistem de ordinul al-II-lea la un semnal sinusoidal

Fie: $x(t) = A \cos \omega_0 t = \frac{1}{2} A e^{j\omega_0 t} + \frac{1}{2} A e^{-j\omega_0 t} = x_a(t) + x_b(t)$, cu proprietatea că: $x_b(t) = x_a^*(t)$.

Răspunsul sistemului de ordinul al-II-lea se calculează pornind de la relația:

$y(t) = H_2[x(t)] = H_2[x_a(t) + x_b(t)] = H_2[x_a(t)] + H_2[x_b(t)] + H_2\{x_a(t), x_b(t)\} + H_2\{x_b(t), x_a(t)\}$	(1.37)
---	--------

Întrucât $h_2(\tau_1, \tau_2)$ este o funcție simetrică în raport cu argumentele sale, avem: $H_2\{x_a(t), x_b(t)\} = H_2\{x_b(t), x_a(t)\}$. Analizând fiecare termen din 2.37 obținem:

$H_2[x_a(t)] = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} h_2(\tau_1, \tau_2) x_a(t - \tau_1) x_a(t - \tau_2) d\tau_1 d\tau_2 = \frac{A^2}{4} H_2(\omega_0, \omega_0) e^{2j\omega_0 t}$	(1.38)
--	--------

În mod similar se poate deduce:

$H_2[x_b(t)] = \frac{A^2}{4} H_2(-\omega_0, -\omega_0) e^{-2j\omega_0 t}$	(1.39)
---	--------

Pentru termenul al-III-lea respectiv al-IV-lea relațiile sunt:

$H_2\{x_a(t), x_b(t)\} = \frac{A^2}{4} H_2(\omega_0, -\omega_0)$	(1.40)
$H_2\{x_b(t), x_a(t)\} = \frac{A^2}{4} H_2(-\omega_0, \omega_0)$	

Însumând cei 4 termeni dați de relațiile 1.38, 1.39 și 1.40 obținem:

$y(t) = 2 \left(\frac{A}{2} \right)^2 \operatorname{Re} \{ H_2(\omega_0, \omega_0) e^{2j\omega_0 t} \} + 2 \frac{A^2}{2} \operatorname{Re} \{ H_2(\omega_0, -\omega_0) \}$	(1.41)
---	--------

După cum se poate observa din relația 1.41, răspunsul constă dintr-o constantă dată de cel de-al II-lea termen al relației și un semnal sinusoidal de frecvență $2\omega_0$ și amplitudine: $\frac{A^2}{2} |H_2(\omega_0, \omega_0)|$.

Utilizarea Transformatei în determinarea răspunsului unui sistem de ordinul al-II-lea la un semnal aperiodic

Răspunsul unui sistem Volterra de ordinul al-II-lea la un semnal de intrare oarecare e dat de relația:

$y_2(t) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} h_2(\tau_1, \tau_2) x(t - \tau_1) x(t - \tau_2) d\tau_1 d\tau_2$	(1.42)
--	--------

Se dorește în cele ce urmează stabilirea unei legături între transformata Fourier a răspunsului $Y(\omega)$, transformata Fourier a semnalului $X(\omega)$ și transformata nucleului $H_2(\omega_1, \omega_2)$. Se introduce în acest scop funcția $y_{(2)}(t_1, t_2)$ conform relației:

$y_{(2)}(t_1, t_2) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} h_2(\tau_1, \tau_2) x(t_1 - \tau_1) x(t_2 - \tau_2) d\tau_1 d\tau_2$	(1.43)
---	--------

Substituind în ecuația de mai sus $t_1 = t_2 = t$, obținem: $y_{(2)}(t, t) = y(t)$. Transformata Fourier bidimensională a lui $y_{(2)}(t_1, t_2)$ este:

$Y_{(2)}(\omega_1, \omega_2) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} y_{(2)}(t_1, t_2) e^{-j(\omega_1 t_1 + \omega_2 t_2)} dt_1 dt_2 =$ $= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} h_2(\tau_1, \tau_2) x(t_1 - \tau_1) x(t_2 - \tau_2) e^{-j(\omega_1 t_1 + \omega_2 t_2)} d\tau_1 d\tau_2 dt_1 dt_2$	(1.44)
---	--------

Făcând schimbarea de variabilă: $\sigma_1 = t_1 - \tau_1$ și $\sigma_2 = t_2 - \tau_2$ se obține în final:

$Y_{(2)}(\omega_1, \omega_2) = H_2(\omega_1, \omega_2) X(\omega_1) X(\omega_2)$	(1.44')
---	---------

Pentru determinarea lui $y(t)$ se aplică transformarea inversă:

$y(t) = y_{(2)}(t, t) = \frac{1}{(2\pi)^2} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} Y_{(2)}(\omega_1, \omega_2) e^{j(\omega_1 + \omega_2)t} d\omega_1 d\omega_2$	(1.45)
---	--------

Substituind: $\omega = \omega_1 + \omega_2$, obținem în final:

$y(t) = \frac{1}{(2\pi)^2} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} Y_{(2)}(\omega_1, \omega - \omega_1) e^{j\omega t} d\omega d\omega_1$	(1.46)
--	--------

Analizând 1.46, rezultă că $y(t)$ se poate exprima ca transformata Fourier inversă a funcției:

$Y(\omega) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} Y_{(2)}(\omega_1, \omega - \omega_1) d\omega_1$	(1.47)
--	--------

Un rol important în sinteza operatorilor de ordinul al-II-lea îl joacă transformata Laplace bidimensională a nucleului de ordinul al-II-lea definită de relația:

$H_2(s_1, s_2) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} h_2(\tau_1, \tau_2) e^{-(s_1 \tau_1 + s_2 \tau_2)} d\tau_1 d\tau_2$	(1.48)
--	--------

unde: $s_1 = \sigma_1 + j\omega_1$ și $s_2 = \sigma_2 + j\omega_2$

1.2.5 Transformata Fourier a nucleului de ordin p

În cadrul paragrafului 1.2.3. s-a introdus transformata Fourier bidimensională, atașată nucleului Volterra de ordinul al-II-lea. În continuare se generalizează noțiunea de transformată Fourier în cazul funcțiilor de mai multe variabile. Fie nucleul de ordin p , $h_p(\tau_1, \dots, \tau_p)$, atașat operatorului Volterra de același ordin așa cum se indică în relația 1.33, o funcție de p variabile care satisface relația:

$\int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} h_p(\tau_1, \dots, \tau_p) d\tau_1 \dots d\tau_p < \infty$	(1.49)
---	--------

Aplicând transformarea Fourier funcției $h_p(\tau_1, \dots, \tau_p)$, în raport cu fiecare din variabilele τ_1, \dots, τ_p , pe rând, se ajunge în final la:

$H_p(\omega_1, \dots, \omega_p) = \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} h_p(\tau_1, \dots, \tau_p) e^{-j(\omega_1 \tau_1 + \dots + \omega_p \tau_p)} d\tau_1 \dots d\tau_p$	(1.50)
---	--------

Funcția $H_p(\omega_1, \dots, \omega_p)$ este numită transformata Fourier a nucleului de ordin p și este simetrică în raport cu argumentele sale, dacă $h_p(\tau_1, \dots, \tau_p)$ este o funcție simetrică. Transformarea Fourier inversă se definește pe baza relației:

$h_p(\tau_1, \dots, \tau_p) = \frac{1}{(2\pi)^p} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} H_p(j\omega_1, \dots, j\omega_p) e^{j(\omega_1\tau_1 + \dots + \omega_p\tau_p)} d\omega_1 \dots d\omega_p$	(1.51)
---	--------

Utilizarea transformatei în determinarea răspunsului unui sistem de ordin p la un semnal de intrare sinusoidal

Determinarea răspunsului unui sistem de ordin p la un semnal de intrare sinusoidal se va exemplifica, din motive de simplificare a calculului, pentru cazul $p=3$. Fie și în acest caz semnalul de intrare de forma: $x(t) = A \cos \omega_0 t = \frac{1}{2} A e^{j\omega_0 t} + \frac{1}{2} A e^{-j\omega_0 t} = x_a(t) + x_b(t)$, cu proprietatea că: $x_b(t) = x_a^*(t)$. Răspunsul sistemului de ordinul al-III-lea la acest semnal de intrare este dat de:

$y_3(t) = H_3[x_a(t) + x_b(t)]$	(1.52)
---------------------------------	--------

și conduce la relația:

$y_3(t) = H_3[x_a(t)] + H_3[x_b(t)] + 3H_3\{x_a(t), x_a(t), x_b(t)\} + 3H_3\{x_a(t), x_b(t), x_b(t)\}$	(1.53)
--	--------

unde:

$y_3(t) = H_3[x_a(t)] = \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} h_3(\tau_1, \tau_2, \tau_3) d\tau_1 d\tau_2 d\tau_3 = \left(\frac{A}{2}\right)^3 H_3(j\omega_0, j\omega_0, j\omega_0) e^{3j\omega_0 t}$	(1.54)
---	--------

$3H_3\{x_a(t), x_a(t), x_b(t)\} = 3 \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} h_3(\tau_1, \tau_2, \tau_3) x_a(t-\tau_1) x_a(t-\tau_2) x_b(t-\tau_3) d\tau_1 d\tau_2 d\tau_3$ $= 3 \left(\frac{A}{2}\right)^3 H_3(j\omega_0, j\omega_0, -j\omega_0) e^{j\omega_0 t}$	(1.55)
---	--------

$3H_3\{x_a(t), x_b(t), x_b(t)\} = 3 \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} h_3(\tau_1, \tau_2, \tau_3) x_a(t-\tau_1) x_b(t-\tau_2) x_b(t-\tau_3) d\tau_1 d\tau_2 d\tau_3$ $= 3 \left(\frac{A}{2}\right)^3 H_3(j\omega_0, -j\omega_0, -j\omega_0) e^{j\omega_0 t}$	(1.56)
--	--------

$H_3[x_b(t)] = \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} h_3(\tau_1, \tau_2, \tau_3) x_b(t-\tau_1) x_b(t-\tau_2) x_b(t-\tau_3) d\tau_1 d\tau_2 d\tau_3$ $= \left(\frac{A}{2}\right)^3 H_3(-j\omega_0, -j\omega_0, -j\omega_0) e^{-3j\omega_0 t}$	(1.57)
--	--------

Însumând termenii dați de relațiile: 1.54, 1.55, 1.56 și 1.57 se obține:

$y_3(t) = 2 \left(\frac{A}{2}\right)^3 \operatorname{Re}\{H_3(j\omega_0, j\omega_0, j\omega_0) e^{3j\omega_0 t}\} + 3 \cdot 2 \cdot \left(\frac{A}{2}\right)^3 \operatorname{Re}\{H_3(j\omega_0, j\omega_0, -j\omega_0) e^{j\omega_0 t}\}$	(1.58)
--	--------

Din relația 1.58 se poate observa că semnalul de ieșire conține armonicile întâia și a treia a semnalului de intrare.

În mod asemănător se poate deduce expresia răspunsului la semnal sinusoidal a unui filtru Volterra caracterizat de operatorul de ordin IV.

$y_4(t) = H_4[A \cos \omega_0 t] = 2 \left(\frac{A}{2}\right)^4 \operatorname{Re}\{H_4(j\omega_0, j\omega_0, j\omega_0, j\omega_0) e^{j4\omega_0 t}\}$ $+ 4 \cdot 2 \cdot \left(\frac{A}{2}\right)^4 \operatorname{Re}\{H_4(j\omega_0, j\omega_0, j\omega_0, -j\omega_0) e^{2j\omega_0 t}\}$ $+ 6 \left(\frac{A}{2}\right)^4 H_4(j\omega_0, j\omega_0, -j\omega_0, -j\omega_0)$	(1.59)
---	--------

1.2.6 Filtre Volterra numerice

Modelul Volterra discret a fost introdus la începutul anilor '80 [W.J.Rugh,81] și interesul manifestat față de el a devenit de atunci din ce în ce mai mare.

Prin analogie cu cazul continuu, un sistem neliniar, invariant în timp, cu memorie este descris de o serie Volterra discretă dată de relația 2.60:

$y[n] = h_0 + \sum_{p=1}^{\infty} H_p[x[n]]$	(1.60)
--	--------

obținută prin eșantionarea ecuațiilor 1.1 și 1.2, unde $y[n]$ și $x[n]$ rezultă din eșantionarea lui $y(t)$ respectiv $x(t)$ cu pasul de eșantionare normalizat: $T=1$ și :

$H_p[x[n]] = \sum_{m_1=0}^{\infty} \dots \sum_{m_p=0}^{\infty} h_p[m_1, \dots, m_p] x[n-m_1] \dots x[n-m_p]$	(1.61)
--	--------

dacă variabilele m_1, \dots, m_p au un suport strict cauzal.

Termenul constant h_0 este un termen de polarizare, $h_1[m_1]$ reprezintă răspunsul la impuls al unui filtru IIR și $h_p[m_1, \dots, m_p]$ poate fi considerat ca răspunsul sistemului la impulsul p -dimensional.

Ca și în cazul liniar se pot construi modelele recursive de ordin finit care conțin termeni de ieșire întârziată și se pot obține dezvoltări polinomiale recursive. Acest lucru poate fi realizat prin alegerea limitei de însumare finită (N) în 1.60. În acest caz $h_1[m_1]$ reprezintă răspunsul la impuls al unui filtru FIR, iar efectul nelinierității este caracterizat la ieșire de valori prezente și trecute ale semnalului de intrare.

În plus, s-a demonstrat că un sistem discret, cauzal și invariant în timp, cu memorie finită, având proprietatea de a răspunde la variații mici ale semnalului de intrare prin variații mici ale semnalului de ieșire poate fi aproximat uniform, în concordanță cu teorema Stone-Weierstrass, pentru un set de semnale de intrare uniform mărginit, de către o serie Volterra finită, de forma [39]:

$y[n] = h_0 + \sum_{p=0}^N H_p[x[n]]$	(1.62)
---------------------------------------	--------

unde:

$H_p[x[n]] = \sum_{m_1=0}^{M-1} \dots \sum_{m_p=0}^{M-1} h_p[m_1, \dots, m_p] x[n-m_1] \dots x[n-m_p]$	(1.63)
--	--------

Modelele nerecursive au fost deasemenea intens studiate în literatură datorită simplității lor.

1.2.7 Filtrul Volterra discret de ordinul al-II-lea (FV2)

Ca și în cazul continuu se va prezenta în cele ce urmează filtrul Volterra discret de ordinul al-II-lea, el constituind un prototip în cadrul filtrelor Volterra numerice. Relația ce caracterizează filtrul în domeniul timp este:

$y[n] = h_0 + \sum_{k_1=0}^{N-1} h_1[k_1] x[n-k_1] + \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{N-1} h_2[k_1, k_2] x[n-k_1] x[n-k_2]$	(1.64)
---	--------

unde $h_1[k_1]$, reprezintă coeficienții nucleului liniar de filtrare, iar $h_2[k_1, k_2]$, reprezintă coeficienții nucleului pătratic de filtrare. În cele ce urmează nucleul pătratic va fi considerat o funcție simetrică în raport cu argumentele, adică: $h_2[k_1, k_2] = h_2[k_2, k_1]$.

Transformata Fourier a nucleului de ordinul al-II-lea se definește cu relația:

$$H_2(\Omega_1, \Omega_2) = \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{N-1} h_2[k_1, k_2] e^{-j\Omega_1 k_1} e^{-j\Omega_2 k_2} \quad (1.65)$$

Transformata nucleului, $H_2(\Omega_1, \Omega_2)$ își găsește numeroase aplicații în determinarea răspunsului FV2 la diferite semnale de intrare.

Răspunsul în frecvență al FV2 la un semnal de intrare determinist

Pentru determinarea răspunsului în frecvență, $Y(\Omega)$, al unui FV2 este necesară mai întâi găsirea răspunsului bispectral al nucleului pătratic la un semnal de intrare cu transformata $X(\Omega)$. Fie $y_2[n]$, contribuția termenului pătratic la răspunsul în timp al filtrului:

$$y_2[n] = \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{N-1} h_2[k_1, k_2] x[n-k_1] x[n-k_2] \quad (1.66)$$

Ca și în cazul continuu, se definește semnalul teoretic:

$$y_{(2)}[n_1, n_2] = \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{N-1} h_2[k_1, k_2] x[n_1-k_1] x[n_2-k_2] \quad (1.67)$$

Transformata Fourier bidimensională a lui $y_{(2)}[n_1, n_2]$ e dată de relația [40]:

$$Y_{(2)}(\Omega_1, \Omega_2) = \sum_{n_1=0}^{\infty} \sum_{n_2=0}^{\infty} \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{N-1} h_2[k_1, k_2] x[n_1-k_1] x[n_2-k_2] e^{-j\Omega_1 n_1} e^{-j\Omega_2 n_2} \quad (1.68)$$

făcând schimbarea de variabilă: $u_1 = n_1 - k_1$ și $u_2 = n_2 - k_2$, se obține în final:

$$Y_{(2)}(\Omega_1, \Omega_2) = H_2(\Omega_1, \Omega_2) X(\Omega_1) X(\Omega_2) \quad (1.69)$$

$Y_{(2)}(\Omega_1, \Omega_2)$ se numește răspunsul bispectral al nucleului pătratic la excitația $X(\Omega)$.

Ținând cont de relația: $y_2[n] = y_{(2)}[n, n]$ se poate deduce:

$$\begin{aligned} y_2[n] &= \frac{1}{(2\pi)^2} \iint_{2\pi} H_2(\Omega_1, \Omega_2) X(\Omega_1) X(\Omega_2) e^{j\Omega_1 n} e^{j\Omega_2 n} d\Omega_1 d\Omega_2 = \\ &= \frac{1}{(2\pi)^2} \iint_{2\pi} H_2(\Omega_1, \Omega_2) X(\Omega_1) X(\Omega_2) e^{j(\Omega_1 + \Omega_2)n} d\Omega_1 d\Omega_2 \end{aligned} \quad (1.70)$$

Analizând relația de mai se observă că $y_2[n]$, reprezintă transformata Fourier inversă a funcției:

$$Y_2(\Omega) = \frac{1}{2\pi} \int_{2\pi} Y_{(2)}(\Omega_1, \Omega - \Omega_1) d\Omega_1 \quad (1.71)$$

Relația 1.71 realizează o combinare a frecvențelor întrucât funcția $Y_{(2)}$ e integrată în lungul drepte de ecuație: $\Omega = \Omega_1 + \Omega_2$.

Răspunsul în frecvență al nucleului liniar fiind:

$$H_1(\Omega) = \sum_{k_1=0}^{N-1} h_1[k_1] e^{-j\Omega k_1} \quad (1.72)$$

rezultă răspunsul în frecvență al FV2:

$$Y(\Omega) = H_1(\Omega) X(\Omega) + \frac{1}{2\pi} \int_{2\pi} Y_{(2)}(\Omega_1, \Omega - \Omega_1) d\Omega_1 \quad (1.73)$$

Răspunsul în frecvență al FV2 la un semnal de intrare aleator

Acest paragraf va trata cazul particular al semnalului de intrare având distribuție de probabilitate gaussiană. Cazul general necesită calcule deosebit de laborioase fiind obiectul de studiu al unor lucrări recente.

Asupra semnalului de intrare se fac următoarele ipoteze: se consideră că acesta e ergodic având medie nulă și momente statistice finite pentru $k \leq 4$. Momentul statistic de ordin k e dat de relația:

$m_k = \int_{-\infty}^{+\infty} x^k p(x) dx$	(1.74)
--	--------

Se definește funcția de autocorelație temporală a variabilei aleatoare x conform cu relația de mai jos:

$r_{xx}[k_1 - k_2] = \lim_{N \rightarrow \infty} \frac{1}{2N + 1} \sum_{n=-N}^N x[n - k_1] x[n - k_2]$	(1.75)
--	--------

Pentru ca semnalul $y[n]$, definit de relația 1.64 să aibe media nulă e necesar ca:

$h_0 = - \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{N-1} h_2[k_1, k_2] r_{xx}[k_1 - k_2]$	(1.76)
---	--------

Relația 1.64 devine în acest caz:

$y[n] = \sum_{k_1=0}^{N-1} h_1[k_1] x[n - k_1] + \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{N-1} h_2[k_1, k_2] \{x[n - k_1] x[n - k_2] - r_{xx}[k_1 - k_2]\}$	(1.77)
---	--------

Funcția de autocorelație a lui $y[n]$ este:

$r_{yy}[k] = \lim_{N \rightarrow \infty} \frac{1}{2N + 1} \sum_{n=-N}^N y[n + k] y[n]$	(1.78)
--	--------

Relația de mai sus se poate scrie:

$r_{yy}[k] = \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{N-1} h_1[k_1] h_1[k_2] r_{xx}[k_1 - k_2 + k] - 2 \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{N-1} \sum_{l_1=0}^{N-1} \sum_{l_2=0}^{N-1} h_2[k_1, k_2] h_2[l_1, l_2] r_{xx}[k_1 - k_2 + k] r_{xx}[l_1 - l_2 + k]$	(1.79)
---	--------

În deducerea relației de mai sus s-a ținut cont de proprietatea variabilelor aleatoare cu densitate de probabilitate gaussiană referitoare la medierea statistică a produsului conținând un număr impar, respectiv par de variabile aleatoare, precum și de faptul că $x[n]$ fiind un semnal ergodic, medierile statistice sunt egale cu cele temporale, corespunzătoare. În cazul relației 2.79 s-au folosit relațiile de mai jos:

$\overline{x_1 x_2 x_3} = 0$ $\overline{x_1 x_2 x_3 x_4} = \overline{x_1 x_2} + \overline{x_3 x_4} + \overline{x_1 x_3} + \overline{x_2 x_4} + \overline{x_1 x_4} + \overline{x_2 x_3}$	(1.80)
--	--------

În relația 1.80 prin bară s-a notat operatorul de mediere temporală.

Densitățile spectrale de putere pentru semnalele $x[n]$ și $y[n]$ se definesc în mod uzual:

$\Phi_{xx}(\Omega) = \sum_{k=-\infty}^{\infty} r_{xx}[k] e^{-j\Omega k}$ $\Phi_{yy}(\Omega) = \sum_{k=-\infty}^{\infty} r_{yy}[k] e^{-j\Omega k}$	(1.81)
--	--------

Printr-o procedură similară celei prezentate în cazul determinării răspunsului FV2 la un semnal de intrare determinist, relația dintre $\Phi_{xx}(\Omega)$ și $\Phi_{yy}(\Omega)$ devine:

$$\Phi_{yy}(\Omega) = |H_1(\Omega_1)|^2 \Phi_{xx}(\Omega) + 2 \frac{1}{2\pi} \int_{-\pi}^{\pi} \Phi_{(2)yy}(\Omega_1, \Omega - \Omega_1) d\Omega_1 \quad (1.82)$$

unde s-au folosit notațiile:

$$\Phi_{(2)yy}(\Omega_1, \Omega_2) = |H_2(\Omega_1, \Omega_2)|^2 \Phi_{xx}(\Omega_1) \Phi_{xx}(\Omega_2) \quad (1.83)$$

Densitatea de putere interspectrală este definită ca fiind:

$$\Phi_{xy}(\Omega) = \sum_{k=-\infty}^{\infty} r_{yx}[k] e^{-jk\Omega} \quad (1.84)$$

unde:

$$r_{yx}[k] = \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{n=-N}^N y[n] x[n+k] \quad (1.85)$$

Efectuând calculele se obține:

$$\begin{aligned} r_{yx}[k] &= \sum_{k_1=0}^{N-1} h_1[k_1] r_{xx}[k+k_1] \\ \Phi_{xy}(\Omega) &= H_1(\Omega) \Phi_{xx}(\Omega) \end{aligned} \quad (1.86)$$

Această relație similară celei obținute în cazul filtrării liniare se datorește proprietăți variabilelor aleatoare gaussiene specificată în 1.80.

Densitatea de putere inter-bispectrală, se va defini conform relației:

$$S_{xy}(\Omega_1, \Omega_2) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} t_{yx}[k_1, k_2] e^{-j(k_1\Omega_1 + k_2\Omega_2)} \quad (1.87)$$

unde:

$$t_{yx}[k_1, k_2] = \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{n=-N}^N y[n] x[n+k_1] x[n+k_2] \quad (1.88)$$

reprezintă intercorelația între secvența $y[n]$ și $x[n+k_1] x[n+k_2]$.

$S_{xy}(\Omega_1, \Omega_2)$ devine:

$$S_{xy}(\Omega_1, \Omega_2) = 2H_2(\Omega_1, \Omega_2) \Phi_{xx}(\Omega_1) \Phi_{xx}(\Omega_2) \quad (1.89)$$

Relațiile 1.86 și 1.87 dovedesc faptul că în cazul semnalului de intrare gaussian, funcțiile de transfer $H_1(\Omega)$, respectiv $H_2(\Omega_1, \Omega_2)$, pot fi cunoscute în măsura în care se cunosc $\Phi_{xy}(\Omega)$, $S_{xy}(\Omega_1, \Omega_2)$ și $\Phi_{xx}(\Omega)$. Această proprietate facilitează proiectarea filtrului optimal Volterra.

Analizând relațiile 1.62 și 1.63 și făcând presupunerea că nucleele $h_p[m_1, \dots, m_p]$ sunt funcții simetrice în raport cu argumentele m_i , rezultă că ieșirea filtrului Volterra discret constă din convoluții multidimensionale între coeficienții filtrului și semnalul de intrare.

Ca o consecință a acestui fapt este posibil și adeseori util să interpretăm filtrul în domeniul frecvență, utilizând transformata Z. De exemplu, dacă notăm $y_p[n]$, termenul de ordin p din relația 1.62:

$$y_p[n] = \sum_{S_p} h_p[m_1, \dots, m_p] x[n-m_1] \dots x[n-m_p] \quad (1.90)$$

Transformata Z a lui y_p poate fi exprimată cu ajutorul lui $X(z)$ și a lui $H_p(z_1, \dots, z_p)$:

$Y_p(z) = F[H_p(z_1, \dots, z_p)]X(z_1) \cdots X(z_p)$	(1.91)
--	--------

unde:

$H_p(z_1, \dots, z_p) = \sum_{S_p} h_p(m_1, \dots, m_p) z_1^{m_1} \cdots z_p^{m_p}$	(1.92)
---	--------

și reprezintă transformata Z p -dimensională a nucleului h_p , iar F este operația prin care transformata Z p -dimensională este redusă la o transformată Z unidimensională. Operația poate fi interpretată ca un combinator de frecvențe și a fost prezentată anterior pentru cazul $p = 2$.

1.2.8 Proprietăți ale filtrelor Volterra.

A. Proprietăți structurale ale nucleelor

Dezvoltarea în serie Volterra prezintă proprietăți importante atât în timp continuu cât și în timp discret. Se prezintă în continuare unele dintre aceste proprietăți pentru dezvoltarea în timp discret.

- Așa după cum s-a specificat în prezentarea filtrului Volterra în timp continuu, nucleele Volterra sunt funcții simetrice în raport cu argumentele și în consecință toate cele $p!$ permutări posibile ale argumentelor m_1, \dots, m_p păstrează nucleul nemodificat. Această proprietate a nucleelor Volterra reduce mult complexitatea calculului în cadrul dezvoltării în serie Volterra. Complexitatea calculului este dată de lungimea memoriei utilizate precum și de ordinul filtrului. Astfel nucleul de ordin p al modelului conține, în cazul unei ferestre de filtrare de lungime N , N^p coeficienți. Numărul acestora poate fi redus datorită proprietății de simetrie la C_N^p .
- În ceea ce privește identificarea nucleelor Volterra s-a arătat deasemenea că răspunsul la impuls al sistemului de un anumit ordin nu e suficient pentru a identifica toate elementele nucleului corespunzător. Pentru identificarea nucleului de ordin p e necesar un semnal de intrare constând din produsul a p impulsuri unitate plasate corespunzător pe suportul filtrului.
- Nucleele Volterra reprezintă o extensie directă a conceptului de răspuns la impuls din teoria sistemelor liniare în cazul sistemelor multidimensionale. Această observație permite atribuirea unei semnificații fizice nucleelor Volterra; ele caracterizează sistemul neliniar fiind analogul multidimensional al răspunsului la impuls.
- O altă caracteristică impusă uneori în modelarea și realizabilitatea filtrelor Volterra o constituie proprietatea nucleelor de a fi separabile. Această proprietate se poate exprima astfel:

$h_p(m_1, \dots, m_p) = \prod_{j=1}^p h_j(m_j)$	(1.93)
---	--------

B. Proprietăți structurale ale relației intrare - ieșire în cazul filtrelor Volterra

Aceste proprietăți sunt deosebit de importante pentru a caracteriza comportarea neliniară a filtrelor Volterra.

- Prima proprietate se referă la liniaritatea relației intrare-ieșire în raport cu nucleele, privite ca și coeficienți ai filtrului, așa după cum se vede din relațiile 1.62 și 1.63. În expresia semnalului de ieșire, neliniaritatea se reflectă în produsele multiple între valori ale semnalului de intrare întârziate, în timp ce, relația e liniară în raport cu coeficienții filtrului. Această proprietate specifică permite extinderea diverselor concepte și proceduri aplicate în cazul filtrării liniare și sistemelor neliniare. Este vorba de principiul proiecției ortogonale, teoria filtrării optimale precum și a implementării algoritmilor de filtrare adaptivă.
- A doua proprietate pornește de la observația făcută asupra relațiilor 1.61, 1.63, și anume că neliniaritatea sistemului se reflectă în nucleele multidimensionale care apar în relația intrare-ieșire și a căror dimensiune (ordin) e dată de numărul de factori în produsele între valorile întârziate ale semnalului de intrare. Astfel relația intrare-ieșire poate fi privită ca o sumă de convoluții multidimensionale. Comparând termenul de ordinul p al dezvoltării, dat de relația 1.63, cu expresia unei convoluții p -dimensionale :

$$\omega[n_1, \dots, n_p] = \sum_{m_1=0}^{M-1} \dots \sum_{m_p=0}^{M-1} h_p[m_1, \dots, m_p] v[n_1 - m_1, \dots, n_p - m_p] \quad (1.94)$$

se poate observa că, pentru : $n_1 = \dots = n_p = n$, se obține:

$$y[n] = \omega[n, \dots, n] \quad (1.95)$$

Relația 1.95 arată că ieșirea unui sistem neliniar este dată de particularizarea convoluției multidimensionale la elementele diagonalei principale ale suportului filtrului multidimensional corespunzător.

Cea de-a doua proprietate permite deasemenea o interpretare în domeniul frecvență a filtrelor Volterra. În acest scop, pornind de la expresia răspunsului în frecvență al unui sistem p -dimensional, liniar, se poate deduce comportarea în frecvență a filtrului caracterizat de un nucleu de ordin p .

Răspunsul sistemului p -dimensional, liniar, caracterizat de funcția $h_p(m_1, \dots, m_p)$, la semnalul $v(n_1, \dots, n_p) = e^{j\Omega_1 n_1} \dots e^{j\Omega_p n_p}$ este:

$$w(n_1, \dots, n_p) = H_p(e^{j\Omega_1}, \dots, e^{j\Omega_p}) e^{j\Omega_1 n_1} \dots e^{j\Omega_p n_p} \quad (1.96)$$

unde :

$$H_p(e^{j\Omega_1}, \dots, e^{j\Omega_p}) = \sum_{m_1=0}^{\infty} \dots \sum_{m_p=0}^{\infty} h_p[m_1, \dots, m_p] e^{-j\Omega_1 m_1} \dots e^{-j\Omega_p m_p} \quad (1.97)$$

Ieșirea filtrului Volterra caracterizat de nucleul de ordin p se poate obține pe baza relației 1.95 presupunând:

$$v[n_1, \dots, n_p] = x[n_1] \dots x[n_p] \quad ; \quad n_1 = \dots = n_p = n \quad (1.98)$$

Se obține astfel:

$$y[n] = w[n, \dots, n] = H_p(e^{j\Omega_1}, \dots, e^{j\Omega_p}) e^{j(\Omega_1 + \dots + \Omega_p)n} \quad (1.99)$$

Relația 1.99 arată prezența în semnalul de ieșire a frecvenței $\Omega_1 + \dots + \Omega_p$, neprezentă în semnalul de intrare.

Dacă semnalul de intrare este un semnal de timp continuu, având spectrul $X(\omega)$, se poate deduce pentru semnalul de ieșire expresia:

$$y(t) = w(t, \dots, t) = \frac{1}{(2\pi)^p} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} H_p(e^{j\omega_1}, \dots, e^{j\omega_p}) X(\omega_1) \dots X(\omega_p) e^{j(\omega_1 + \dots + \omega_p)t} d\omega_1 \dots d\omega_p \quad (1.100)$$

Relația 1.100 indică prezența în semnalul de ieșire a unei componente având frecvența $\omega_1 + \dots + \omega_p$. În cazul în care semnalul de la intrare este o sinusoidă în semnalul de ieșire apar componente având frecvența multiplii sau submultiplii ai frecvenței semnalului de intrare. Dacă semnalul de intrare este o combinație liniară de sinusoidă, în semnalul de ieșire apar termeni de intermodulație.

Aceste considerații generale confirmă una din particularitățile ce atestă comportarea neliniară a filtrelor Volterra și anume: apariția în semnalul de ieșire a unor componente având frecvența diferită de cea a semnalului de intrare.

1.3 Rețele neuronale

1.3.1 Generalități

Rețelele moderne de comunicații au un număr de sute de mii de noduri, cu diferite tipuri de surse, diferite tipuri de trafic, care deserveșc un număr variabil, foarte mare de utilizatori. Principala caracteristică a traficului de comunicații vehiculat este că aproape toți parametrii săi variază, respectiv numărul utilizatorilor, topologia rețelei, ratele de transfer ale informației, lățimea de bandă necesară. Banda limitată de frecvențe restricționează dezvoltarea sistemelor de telecomunicații. În plus emițătorul, receptorul, cât și canalul de comunicație introduc diferite tipuri de perturbații asupra semnalului util.

În acest context, introducerea unor tehnici de control inteligente, eficiente, capabile să se adapteze necesităților comunicațiilor moderne este vitală. Acestea trebuie să elimine perturbațiile

introduse la toate nivelele sistemului, să asigure flexibilitatea și fiabilitatea rețelei de comunicații, utilizarea eficientă a canalului disponibil, administrarea optimă a resurselor disponibile în raport cu diferite medii de propagare, aflate într-o permanentă dinamică.

Între toate tehnicile inteligente RN se remarcă prin caracteristicile lor: adaptabilitatea, flexibilitatea, fiabilitatea și viteza extraordinară de procesare. S-au raportat viteze de procesare de Terra operații pe secundă pentru un chip de 1 cm² [L.Chua, T.Roska].

Utilizarea RN se recomandă în special în probleme care:

- nu pot fi modelate prin metode clasice datorită cantității mari de date ce trebuie manipulate sau complexității, respectiv în probleme pentru care tehnicile convenționale n-au soluții;
- implică procese aleatorii;
- nu necesită explicarea deciziilor, deoarece RN sunt ca și o "cutie neagră", imposibilitatea de a da răspunsuri legate de modul în care au găsit soluția fiind una dintre carențele lor;

Rețelele neuronale sunt sisteme neliniare formate dintr-un număr mare de procesoare elementare, relativ simple numite neuroni (unități sau perceptrone) care operează în paralel. Procesoarele interacționează între ele prin intermediul conexiunilor: excitatorii și inhibitorii, cărora le sunt asociate ponderi. Învățarea se realizează prin modificarea ponderilor conform unei reguli de învățare.

Un model al elementului de bază al unei RN este reprezentat în Fig.1.6.

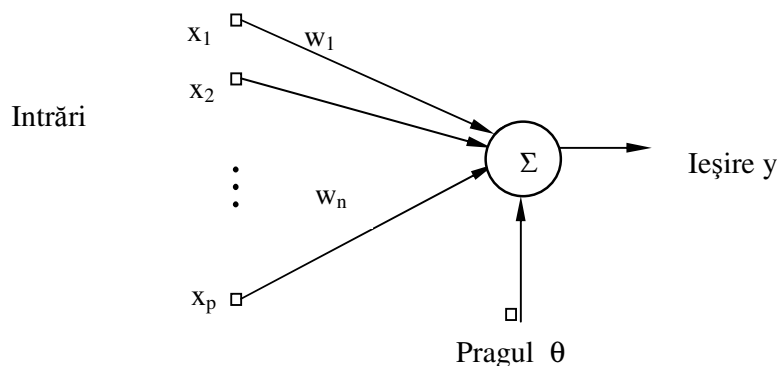


Fig. 1.6 Modelul unui neuron

Intrările x_i sunt înmulțite cu ponderile w_i corespunzătoare și combinate printr-un operator (uzual adunare), formând intrarea netă, net_i . Intrarea netă este comparată cu pragul θ și trecută prin funcția de activare pentru a genera ieșirea. Deci ieșirea neuronului este dată de relația:

$y = f \left(\sum_{i=1}^p w_i x_i - \theta \right)$	(1.106)
--	---------

unde:

- w_i care multiplică intrarea x_i se numește ponderea interconexiunii;
- x_i este intrarea i ;
- θ este o constantă care se numește prag și trebuie sustras din suma produselor dintre intrări și ponderi;
- funcția $f(\cdot)$ este funcția de activare;
- y este ieșirea;

Funcția de activare poate fi o funcție deterministă, uzual neliniară, sau probabilistică. În cele ce urmează se prezintă câteva exemple:

- funcția Heaviside (cunoscută în rețelele neuronale și sub numele de hard-limitatoare): unipolară $\sigma(t)$ (treaptă unitate) sau bipolară $\text{sgn}(t)$;

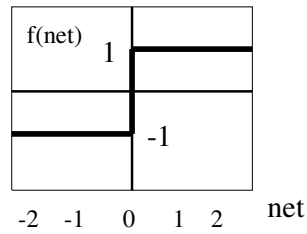


Fig. 1.7 Funcția Heaviside

$$y_i(t+1) = \text{sgn}(net_i(t)) \begin{cases} 1 & \text{dacă } net_i(t) \geq 0 \\ -1 & \text{dacă } net_i(t) < 0 \end{cases} \quad (1.107)$$

Uneori intrarea netă trebuie să depășească o valoare numită prag pentru determinarea unei noi activări:

$$y_i(t+1) = f_i\left(\sum_j w_{ij} a_j(t) - \theta_i(t)\right) \quad (1.108)$$

O funcție semiliniară unipolară poate fi definită printr-o relație de forma:

$$y_i(t+1) = f(net) = \begin{cases} 0, & \text{pentru } net(t) \leq -\theta \\ \frac{net(t) + \theta}{2\theta}, & \text{pentru } -\theta < net < \theta \\ 1, & \text{pentru } net(t) \geq \theta \end{cases} \quad (1.109)$$

Pentru varianta bipolară se poate utiliza relația :

$$f_b(net) = 2f(net) - 1 \quad (1.110)$$

Adesea funcția de activare trebuie să fie o funcție neliniară, nedescrescătoare, ca de exemplu:

$$y_i(t+1) = \begin{cases} 1 & \text{dacă } net_i(t) > \theta_i \\ net_i(t) & \text{dacă } net_i(t) = \theta_i \\ 0 & \text{în rest} \end{cases} \quad (1.111)$$

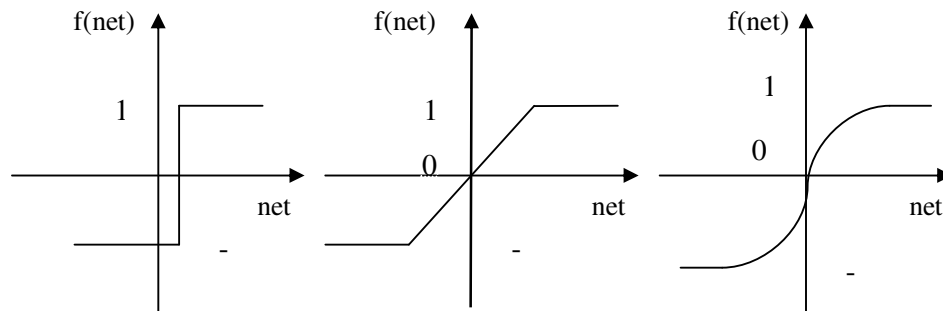


Fig.1.8 Funcții de activare pentru neuron

a) comparatoare bipolară cu prag; b) liniară cu saturație; c) sigmoidală bipolară

Uzual funcția de activare este o funcție neliniară.. Un exemplu este funcția logistică numită și sigmoidă :

$$y_i(t+1) = \frac{1}{1 + e^{-\beta \cdot net_i(t)}} \quad (1.112)$$

Unde β este o constantă aparținând intervalului de valori (0,1).
Există rețele care utilizează funcții de activare probabilistice. Probabilitatea ca neuronul să fie activ este :

$p(y_i(t) \rightarrow 1) = \frac{1}{1 + e^{-\frac{net_i(t)}{T}}}$	(1.113)
---	---------

unde T este o constantă, numită temperatură. Aceasta dă panta curbei de probabilitate.

Rețelele cu funcții neliniare sunt uzual utilizate, performanțele lor fiind superioare celor cu funcții liniare sau semiliniare. Deși majoritatea funcțiilor de activare sunt monotone, există și funcții nonmonotone, care conduc la performanțe foarte bune în special în memoriile asociative. Fig.1.9 prezintă câteva astfel de funcții:

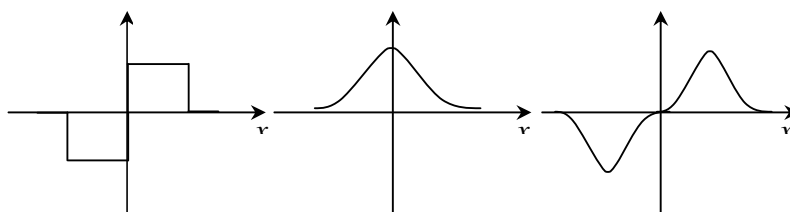


Fig.1.9 Funcții de activare nonmonotone

O pondere pozitivă reprezintă o intrare excitatoare. O pondere negativă reprezintă o intrare inhibitoare. Ponderile w_i sunt modificate în timpul procesului de antrenament, în funcție de experiența RN, în conformitate cu o relație numită regulă de învățare. Câteva dintre regulile de învățare uzuale sunt:

1) Regula lui Hebb

Conform acestei reguli ponderea în pasul k+1 se modifică proporțional cu produsul dintre intrarea și ieșirea neuronului :

$\Delta w_{ij} = \eta \cdot y_i \cdot x_j$	(1.114)
--	---------

unde:

- Δw_{ij} este variația vectorului pondere w_{ij} de la neuronul j către neuronul i din pasul k în pasul (k+1), dată de relația: $w_{ij}[k+1] = w_{ij}[k] + \Delta w_{ij}$;
- y_i este ieșirea neuronului i;
- x_j este intrarea în neuronul j;
- η este o constantă de care depinde viteza de învățare, $\eta \in (0,1)$;

Această regulă de învățare este fără control (nesupravegheată) deoarece nu utilizează răspunsul dorit.

2) Regula perceptronului :

Este o regulă supervizată pentru că în calculul variației ponderii se utilizează răspunsul dorit notat cu d_i :

$\Delta w_{ij} = \eta \cdot [d_i - \text{sgn}(w_i^T x)] \cdot x_j$	(1.115)
--	---------

unde x este vectorul intrărilor în neuronul j $x = [x_1 \ x_2 \ \dots \ x_j \ \dots \ x_n]$

3) Regula Delta (sau regula Widrow-Hoff)

Denumirea de Delta este dată de diferența dintre ieșirea curentă și răspunsul dorit :

$\Delta w_{ij} = \eta \cdot [d_i - y_i] \cdot x_j$	(1.116)
--	---------

Regula este cunoscută și sub denumirea autorilor săi , ca regula Widrow și Hoff.

4) Regula Delta generalizată

Modificările în ponderi se calculează cu :

$$\Delta w_{ij} = \eta \cdot [d_i - y_i] \cdot f'(net_i) x_j \quad (1.117)$$

unde f' este derivata funcției de activare, deci este valabilă doar pentru funcții de activare continue .

5) Regula de învățare a corelației

Este o variantă a regulii lui Hebb:

$$\Delta w_{ij} = \eta \cdot d_i \cdot x_j \quad (1.118)$$

6) Regula de învățare de tip competitiv

În RN competitive se modifică de obicei doar ponderile neuronului câștigător. Există numeroase reguli de acest tip, exemple fiind următoarele două relații:

$$\Delta w_{mj} = \eta \cdot (y_j \cdot w_{mj}) \quad (1.119)$$

unde m este neuronul câștigător și w_{mj} este vectorul său pondere.

O altă regulă de tip competitiv este regula produsului ("outstar" a lui Grossberg):

$$\Delta w_{ij} = \eta \cdot (d_i \cdot w_{ij}) \quad (1.120)$$

Există multe alte reguli de învățare, dezvoltate pentru a ameliora performanțele RN. În fiecare aplicație este adecvată o anumită funcție de activare, care poate fi determinată doar prin metoda încercării.

1.3.2 Algoritmi de antrenare

Configurarea interconexiunilor unei rețele neuronale trebuie făcută astfel încât aplicarea unui set de intrări să genereze un set de ieșiri dorite. Există RN cu ponderi fixe și cu ponderi adaptabile. Deci în determinarea ponderilor interconexiunilor există două direcții distincte:

- fixarea explicită, utilizând informație apriori referitoare la particularitățile și eventual restricțiile la care este supusă aplicația considerată. Astfel de considerații conduc la sisteme specializate, de dimensiuni reduse, ușor de manipulat ;
- determinarea ponderilor prin antrenare, generând rețelei modele de învățat și lăsând-o să-și modifice ponderile conform unei reguli de învățare, în mod iterativ ;

O condiție esențială este ca algoritmul de antrenare să fie convergent, adică la un moment dat ponderile să rămână constante, indiferent de intrările aplicate.

Se poate face o clasificare a RN în funcție de modalitățile de învățare:

RN cu învățare supravegheată (cu control)

Fig.1.10 prezintă schema de principiu a unei RN învățare supravegheată. Se generează rețelei un set de perechi de modele de intrare-modele de ieșire dorite, cu ajutorul cărora se calculează mărimile de eroare în funcție de diferența dintre valoarea reală curentă a ieșirii și cea dorită și se ajustează parametrii rețelei. Răspunsurile dorite pot fi furnizate din exterior sau de către sistemul global care încorporează RN.

$$e[n] = d[n] - y[n] \quad (1.121)$$

Exemple sunt RN care utilizează: regula Delta, algoritmul retropropagării erorii și variantele sale, cuantizarea vectorială cu control, pentru rețelele cu propagarea informației "spre înainte " respectiv algoritmul retropropagării erorii în timp și învățarea în timp real pentru RN recurente.

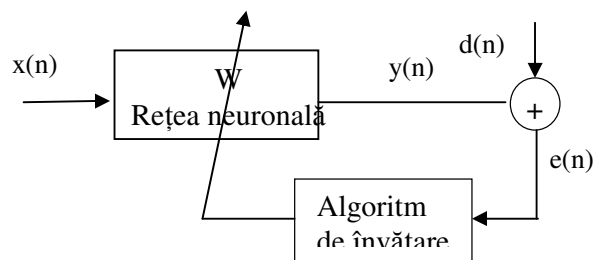


Fig.1.10 RN cu învățare supravegheată

RN cu învățare nesupravegheată (fără control)

În Fig.1.11 este reprezentată schema de principiu a unei RN învățare fără control. RN extrage singură caracteristicile esențiale ale modelelor de intrare, formează reprezentări interne distincte ale acestora și realizează gruparea modelelor pe baza unui criteriu de similaritate. RN utilizează un gen de competiție între neuronii elementari, care are ca efect modificarea ponderilor conexiunilor neuronului care a câștigat întrecerea, eventual și a ponderilor neuronilor învecinați, restul ponderilor interconexiunilor rămânând neafectate.

În unele modele apare un parametru numit conștiință, care intră în funcție când unul dintre neuroni câștigă prea des competiția.

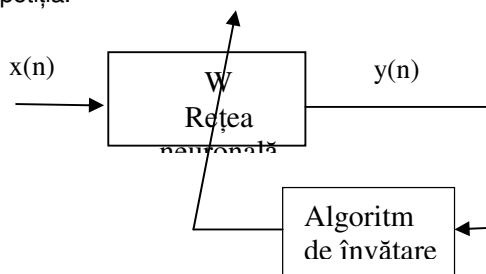


Fig.1.11 RN cu învățare nesupravegheată

Reprezentative pentru această categorie sunt rețelele neuronale auto-organizatoare Kohonen, RN cu cuantizare vectorială, RN pentru analiza componentelor principale. Aceste rețele pot încorpora și un mecanism de control care să permită o rafinare ulterioară a parametrilor.

RN cu învățare cu critic

RN cu învățare cu critic, numite și cu recompensă și pedeapsă, au schema de principiu din Fig.1.12. Rețeaua nu beneficiază de un semnal dorit (ca în învățarea supravegheată), ci, de unul care apreciază cât de bine funcționează sistemul. Algoritmii aparținând acestei categorii se bazează pe observațiile experimentelor cu animale și funcționează după următorul principiu: dacă acțiunea unui sistem capabil să învețe are un efect favorabil, această acțiune este încurajată, în caz contrar este inhibată.

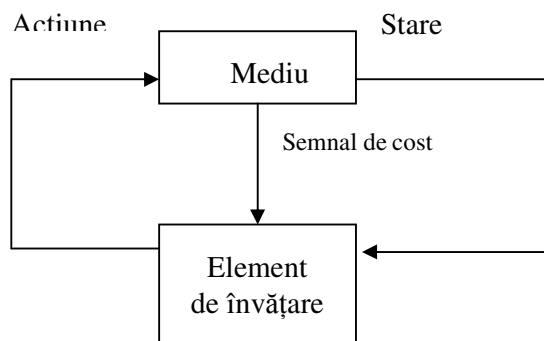


Fig.1.12 RN cu învățare cu critic

1.3.3 Direcții de aplicabilitate a RN în telecomunicații

Datorită potențialului extraordinar de procesare, adaptabilității, flexibilității, și a vitezei extraordinare de procesare aplicațiile RN în telecomunicații sunt în plină expansiune, în toate direcțiile posibile de dezvoltare, dintre care menționăm:

Modelarea și identificarea de sisteme neliniare:

Abilitatea RN de a reprezenta orice funcție, atât reală cât și complexă, liniară cât și neliniară constituie motivul aplicării lor în modelarea și identificarea sistemelor neliniare. Această proprietate este cunoscută sub numele de "aproximare universală" [S.Haykin, 95].

Există aplicații ale RN în modelarea canalelor de comunicații neliniare [M.Ibnkahla, 98], [R.Parisi, 97], modelarea amplificatoarelor neliniare, cum ar fi tuburile cu undă progresivă [M.Ibnkahla,97] și amplificatoarele în stare solidă [M.Ibnkahla,98], proiectarea de emițătoare și

receptoare [U.Mitra, 95], , respectiv în estimarea direcției semnalului și formarea acestuia [A.K.Y.Lai, 02], în proiectarea antenelor adaptive [A.K.Y.Lai, 02] Se urmărește astfel minimizarea perturbațiilor care pot fi introduse în transmisie, determinate de neliniaritatea și fluctuațiile parametrilor canalelor de comunicație, reflexii, efecte de interferență intersimbol și interferență cu canalele adiacente. Utilizarea antenelor adaptive cu RN permite transmisii la viteze mai mari în condițiile unei capacități de recepție superioare. Metodele convenționale, bazate pe algebra liniară, care implică adesea calcule complexe, cu inversări de matrici, nu fac față cerințelor de transmisie în timp real. Ele necesită în plus antene calibrate cu caracteristici uniforme și sunt sensibile la defectele de fabricație și alte incertitudini fizice. RN depășesc toate aceste probleme oferind soluții robuste, precise și în timp real.

Receptoarele convenționale ale semnalelor codate cu diviziune multiplă a accesului (CDMA code division multiple acces) au eșuat în reconstrucția informației atunci când numărul semnalelor transmise a fost mare. În plus aceste receptoare necesită cunoașterea codurilor tuturor utilizatorilor care interferează [M.Varanasi, 90]. Receptoarele de semnale CDMA cu RN elimină atât zgomotul inerent canalului cât și interferența datorată accesului multiplu [X.M.Gao, 97] și reconstruiesc semnalele transmise fără a necesita cunoașterea codurilor de împrăștiere [U.Mitra, 95]. Implementarea VLSI analogică conferă receptoarelor adaptive cu RN viteza de calcul necesară comunicațiilor moderne de debit ridicat [J.Choi, 90].

Egalizarea de canal

Abordând egalizarea de canal ca o problemă de clasificare neliniară, RN generează curbe neliniare de separare a semnalelor, inclusiv complexe, astfel încât realizează cu succes egalizarea și detecția semnalelor transmise prin canale complexe, profund neliniare și variabile în timp [S.Chen, 94]. Performanțele egalizatoarelor neuronale raportate de literatura de specialitate [G.De.Vecciana, 92], [M.Ibnkahla, 97], [A.Hussain, 97], [A.Guntsch, 98], [C.Botoca,00] [N.Miclău,04], sunt superioare egalizatoarelor convenționale atât din punctul de vedere al preciziei cât și al vitezei de procesare.

Utilizarea RN la recepție permite eliminarea atât a distorsiunilor liniare cât și celor neliniare introduse de canalul de comunicație în caracteristica de fază și cea de amplitudine a semnalului de comunicație. Între efectele nedorite compensate menționăm: interferența intersimbol, interferența cu canalele adiacente, zgomotul aditiv, fluctuațiile de fază și amplitudine ale semnalelor transmise datorate condițiilor meteo de propagare.

Eliminarea ecoului

Pentru eliminarea ecoului, respectiv pentru eliminarea zgomotului de fundal și a reverberațiilor, tehnicile convenționale au găsit soluții acceptabile. Suprapunerea acestor două efecte nedorite, mai evidente în cazul transmisiilor fără fir, în condițiile transmisiilor multiple, constituie o problemă în mare parte nerezolvată însă până în prezent. RN sunt o bună alternativă datorită potențialului lor deosebit de procesare, în timp real [M.Moonen, 97].

Codarea, decodarea și codurile corectoare de erori

Această direcție necesită o viteză și o capacitate de procesare deosebite. Adaptabilitatea și flexibilitatea RN le permite să funcționeze eficient în situații complexe, acolo unde presupunerile simplificatoare ale tehnicilor de codare și decodare nu sunt satisfăcute [M.Ibnkahla, 97]

Procesarea de imagini

- Compresia imaginilor este importantă pentru multe aplicații din telecomunicații, ca de exemplu detecția prin satelit la distanță, comunicațiile multimedia, televiziunea digitală, Internetul. RN sunt o alternativă de succes la procesarea tradițională datorită factorului de compresie ridicat realizat, a calității imaginilor reproduse, vitezei de procesare în timp real [S.B.Zahir, 00]. Literatura de specialitate consemnează rezultate remarcabile în această direcție, de la procesările simple de imagini statice, la cele complexe, de imagini video, în mișcare. Menționăm: extragerea de contur, filtrarea, sortarea de obiecte în funcție de orientare sau dimensiune, detecția de minime și maxime într-o imagine pe scara nuanțelor de gri, mărirea și micșorarea de imagine, segmentarea imaginilor și recunoașterea texturilor în imaginile transmise prin satelit, recunoașterea caracterelor scrise pentru automatizarea serviciilor poștale, eliminarea zgârieturilor sau a altor perturbații în faxuri și copiatoare, recunoașterea bancnotelor și blocarea copiatoarelor inteligente pentru evitarea fraudelor.

În ceea ce privește detecția mișcării noi, realizările sunt în plină expansiune în:

- extragerea unui anumit model din imagine (detecția gurii și a ochilor pe imaginile faciale în mișcare, detecția mânerelor de ușă din imagini pentru orientarea orbilor într-un mediu necunoscut),
- reconstrucția de obiecte tridimensionale prin interpolare și aproximare, rotația obiectelor tridimensionale;
- analiza și detecția mișcării;
- detecție de ținte multiple și urmărire;
- controlul roboților la distanță, într-un mediu necunoscut;

Optimizarea traficului de comunicații

Rețelele neuronale fac posibilă rezolvarea problemelor complexe ale comunicațiilor moderne. Ele oferă soluții de mare viteză în:

- caracterizarea și predicția traficului de comunicații [N.Leitao, 95] (estimarea calității serviciilor);
- controlul comutării, controlul fluxului și congestiei [S.Jagannathan, 01], [I.Reijin, 97];
- dirijarea dinamică a traficului de comunicații [E.Nordstrom,95];
- asistarea proiectării de rețea de comunicații;
- atribuirea de canal în rețelele de comunicații mobile ;

Capacitatea RN de a învăța din exemple este un avantaj esențial pentru managementul adaptiv al resurselor. Pentru controlul rapid și exact al traficului este necesară adesea caracterizarea traficului și predicția sa. O RN este capabilă să învețe funcția distribuție de probabilitate a traficului, să selecteze parametri semnificativi statistici și să estimeze valorile previzibile ale lor. Caracterizarea, clasificarea și predicția traficului reprezintă o aplicație directă a RN. Strategiile de control cu RN ale comutării au evoluat în general ca o soluție la problemele de optimizare. Simulările au demonstrat beneficiile utilizării RN în rutarea dinamică prin îmbunătățirea semnificativă a performanțelor în special în condiții de trafic greu.

Analiză și service on-line

Companiile care furnizează servicii de telefonie, bănci, companii de asigurare, oficii de vânzare de bilete on-line utilizează un distribuitor de apel automat. Sistemul este orientat spre oferta de servicii, având un grup de agenți atașați fiecărui serviciu. Sistemul distribuie apelurile sosite spre servicii, respectiv înspre agenți. Utilizarea unei RN pentru monitorizarea on-line a apelurilor cât și pentru analiza off-line duce la îmbunătățirea performanțelor întregului sistem de comunicații și la creșterea parametrilor de calitate ai serviciilor [T.Clarkson, 02].

Detecția fraudelor în telefonia mobilă este o problemă majoră în comunicațiile prezentului, ce implică supravegherea tuturor apelurilor, deci manipularea unei baze de date enorme. Problema implică atât o clasificare statistică cât și o predicție temporală. Un detector pe scară largă de fraudă telefonică, produs de firma NORTEL, este în prezent utilizată cu deosebit succes [T.Clarkson, 02].

Produse de consum

În condițiile comunicațiilor moderne produsele de uz personal necesită capacitatea funcționării la mari viteze, în condițiile unui preț scăzut și a unui consum energetic redus. Adaptabilitatea datorată utilizării unei RN ar duce la creșterea performanțelor, fără a fi necesară creșterea de putere [M.lbnkahla,04].

CAPITOLUL II

Egalizoare

2.1 Introducere

Canalele de comunicație introduc diferite perturbații, ca de exemplu zgomot aditiv, distorsiuni neliniare, atenuări, interferență intersimbol, interferență cu canalele adiacente, având caracteristici de fază și amplitudine variabile în timp în funcție de condițiile meteo de propagare. Amplificatoarele care lucrează în mod uzual în apropierea punctului de saturație introduc neliniarități fără memorie, care combinate cu efectele filtrelor de transmisie și recepție devin neliniarități cu memorie.

Tehnica de înlăturare a efectelor nedorite ale neliniarităților canalelor de comunicații, ale amplificatoarelor precum și ale condițiilor de propagare este egalizarea de canal. În cazul unei dispersii mari a valorilor datelor de intrare, egalizoarele liniare se confruntă cu problema instabilității numerice. Deoarece în general canalele de comunicații sunt variante în timp egalizoarele trebuie să fie adaptive pentru a urmări variațiile în timp ale răspunsului în frecvență al canalului. În cazul distorsiunilor neliniare, generate de canalele variante în timp, egalizoarele liniare nu au performanțe bune. Singura soluție în cazul distorsiunilor neliniare și severe ale canalelor de comunicație o reprezintă egalizoarele neliniare. Datorită caracterului lor neliniar și a adaptabilității, rețelele neuronale reprezintă un bun candidat, ele fiind capabile să modeleze funcții neliniare complexe intrare-ieșire.

Abordând problema egalizării ca o problemă de clasificare, rețelele neuronale pot genera regiuni de decizie arbitrare cu o mare precizie. Studiile realizate în ultimul deceniu au stabilit

superioritatea egalizoarelor neuronale comparativ cu a celor tradiționale, în special în condițiile distorsiunilor neliniare mari și a semnalelor rapid variabile în timp. Implementarea prin intermediul rețelelor neuronale oferă și avantajul unei viteze de procesare deosebite, în timp real, absolut necesară pentru comunicațiile viitorului.

Deoarece comunicațiile viitorului se vor baza pe transmisiile mai eficiente din punct de vedere spectral, ca de exemplu semnalele modulate în cuadratură, cercetarea noastră a vizat în mod special aceste semnale. Semnalele modulate în cuadratură, având anvelopă variabilă, sunt mai puternic afectate atât în fază, cât și în amplitudine, de neliniaritățile introduse în transmisie. Pentru a elimina distorsiunile semnalelor modulate în fază și în cuadratură sunt necesare egalizoare complexe. Egalizoarele complexe sunt extensii directe ale celor reale, obținute prin înlocuirea parametrilor relevanți, ca de exemplu, intrările, ieșirile, ponderile, pragurile și/sau funcțiile de activare, cu mărimi complexe [D.Jianping,02].

S-au studiat diferite tipuri de egalizoare neliniare, variante clasice, variante hibride care încorporează și tehnica neliniară a rețelelor neuronale și variante care folosesc exclusiv rețele neuronale. Egalizoarele hibride sunt uzual combinații între un filtru liniar transversal și diferite rețele neuronale, cu sau fără reacție inversă. Filtrul liniar transversal elimină distorsiunile liniare, ca de exemplu interferența intersimbol, astfel încât rețeaua neuronală se poate ocupa exclusiv de distorsiunile neliniare.

Pentru implementarea rețelelor neuronale s-au abordat diferite arhitecturi: rețele neuronale multistrat [S.Bouchired,99],[M.Ibnkahla, 97], rețele neuronale pe bază de funcții radiale [I.Cha, 95], [S.Chen,94], [N.Miclău, 04], rețele neuronale cu legături funcționale [A.Hussain, 97], [J.C.Patra, 99], rețele neuronale celulare [R.Perfetti, 97], rețele neuronale recurente [G.Kehriotis, 94], [R.Parisi,97], [S.Haykin, 00]. Pentru antrenarea egalizoarelor neuronale s-au studiat diferiți algoritmi: algoritmi de tip gradient de minimizare a erorii și algoritmi competitivi.

2. 2 Problema egalizării

Fig. 1 prezintă poziția relativă a unui egalizor într-un sistem de comunicație. Ieșirea canalului de comunicații $y(n)$ este afectată de un zgomot aditiv $w(n)$ (în cel mai simplu caz modelat ca un proces Gaussian aditiv).

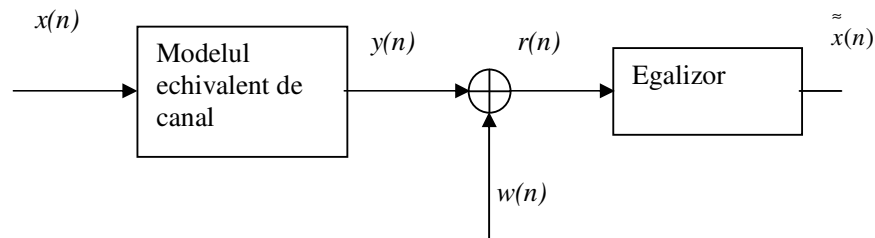


Fig.2.1 Poziția egalizorului în canalul de comunicații

În proiectarea unui egalizor trebuie luați în considerare următorii factori: tipul canalului, zgomotul introdus de receptor în special datorită amplificării și interferența intersimbol;

Canalul de comunicație poate fi fix sau variabil în timp, de fază minimă sau nu. Un canal este de fază minimă dacă zerourile funcției de transfer $H(z)$ sunt conținute toate în interiorul cercului unitate în planul z .

Canalul de comunicație poate introduce atât de distorsiuni liniare cât și distorsiuni neliniare ale semnalelor. Interferența intersimbol determină în special distorsiuni liniare, în timp ce amplificatoarele, convertoarele și mediul de propagare introduc distorsiuni neliniare.

Egalizorul trebuie să realizeze cel mai bun compromis între eliminarea interferenței intersimbol și amplificarea zgomotului introdus de receptor.

Forma convențională a egalizorului se bazează pe teoria filtrului adaptiv liniar, ale cărui ponderi sunt adecvat ajustate de un algoritm ce minimizează eroarea dintre simbolurile de ieșire și cele primite. Uzual se folosește algoritmul erorii medii pătratice minime (Least Mean Square) . Alți algoritmi sunt RLS (Recursive least square) , fast RLS , square root RLS , gradient RLS. Egalizoarele liniare nu au performanțe bune în cazul canalelor cu nuluri spectrale adânci, din moment ce ele plasează un câștig mare la aceste frecvențe, în consecință amplifică zgomotul aditiv. Dacă canalul h este modelat ca un filtru liniar, ieșirea y este dată de convoluția dintre x și h , adică $y = x \otimes h$. Dacă canalul este neliniar lucrurile se complică. În absența zgomotului aditiv și în cazul canalelor de fază minimă, egalizorul liniar operează ca și un sistem invers canalului, furnizând o transmisie fără distorsiuni. Atunci când zgomotul este prezent și canalul nu este de fază minimă, utilizarea modelului invers nu este suficientă.

Amplificarea zgomotului poate fi evitată prin utilizarea tehnicilor neliniare. Diferența dintre egalizoarele liniare și cele neliniare este că cele din urmă utilizează semnalul reconstruit de ieșire într-o cale de reacție inversă pentru adaptarea egalizorului.

Un model clasic de egalizor neliniar este egalizorul cu decizie pe baza reacției inverse (decision feedback equaliser DFE) . Acest egalizor este utilizat în special atunci când interferența intersimbol, este mare, având la o aceeași complexitate performanțe superioare filtrelor liniare transversale. Un alt exemplu este egalizorul care realizează estimarea secvenței de intrare pe baza asemănării maxime. (maximum likelihood sequence estimation).

Dacă în plus în aceste egalizoare se încorporează și o rețea neuronală , performanțele obținute sunt semnificativ îmbunătățite.

Literatura de specialitate consemnează mai multe variante de egalizoare adaptive neuronale implementate cu:

- RN multistrat de tip spre înainte, antrenate cu un algoritm de tip gradient descendent, ca de exemplu algoritmul retropropagării erorii;
- rețele neuronale cu legături funcționale;
- rețele neuronale cu funcții de bază radiale;
- rețele neuronale celulare;
- rețele neuronale recurente

Pentru a elimina distorsiunile semnalelor complexe, ca de exemplu semnalele modulate în fază și în cuadratură (MAQ) sunt necesare egalizoare complexe. Egalizoarele complexe sunt extensii directe ale celor reale, obținute prin înlocuirea parametrilor relevanți, ca de exemplu intrările, ieșirile, ponderile, pragurile și/sau funcțiile de activare, cu mărimi complexe. Structurile ce vor fi prezentate în cele ce urmează combină un filtru liniar transversal cu rețele neuronale de diferite tipuri, într-o arhitectură cu sau fără reacție inversă. Filtrul liniar transversal elimină distorsiunile liniare, ca de exemplu interferența intersimbol, astfel încât rețeaua neuronală se poate concentra asupra eliminării distorsiunilor neliniare.

2.3 Egalizarea ca o problemă de clasificare

Un punct de vedere alternativ la ideea filtrării inverse este abordarea tehnicii de egalizare ca o problemă de clasificare a modelelor. Obiectivul egalizării este în acest caz separarea semnalelor recepționate în spațiul modelelor. În situațiile reale, când zgomotul este prezent în semnalul recepționat și canalul de comunicație, variabil în timp, nu are fază minimă, o clasificare optimă este neliniară. Datorită proprietății lor de aproximatoare universale RN sunt capabile să proceseze regiuni de decizie neliniare, cu o precizie remarcabilă. Viteza deosebită a RN datorată procesării paralele este încă un argument pentru utilizare a lor în implementarea egalizoarelor adaptive.

În cazul unei transmisii bipolare de date, eșantioanele recepționate, afectate de interferența intersimbol și de zgomot, ar trebui clasificate ca +1 sau -1. Funcția egalizorului este în acest caz să asocieze fiecare eșantion recepționat cu regiunea de decizie corectă.

În cazul unor semnale complexe clasificarea trebuie realizată în spațiul complex. Astfel pentru semnalele modulate în cuadratură 4 QAM , constelația intrărilor posibile este:

$x(n) = x_R + jx_I = \begin{cases} x^{(1)} = 1 + j \\ x^{(2)} = -1 + j \\ x^{(3)} = 1 - j \\ x^{(4)} = -1 - j \end{cases}$	(2.1)
--	-------

Din punctul de vedere al unei RN, egalizorul trebuie să clasifice semnalul recepționat într-una dintre cele patru clase posibile:

$P_{m,d} = \bigcup_{1 \leq l \leq 4} P_{m,d}(l)$	(2.2)
--	-------

sau:

$P_{m,d}(l) = \{y(n) x(n-d) = x^{(l)}\}$ $1 \leq l \leq 4$	(2.3)
--	-------

2. 4 Un model al problemei de egalizare

Fie un semnal $x(n)$ aplicat la intrarea unui canal liniar dispersiv cu răspuns finit la impuls (FIR) (Fig.2.2), care poate fi modelat cu un filtru FIR de ordinul k , cu valori reale și nenule ale coeficienților a_0, a_1, \dots, a_k . Semnalul aleator de intrare generează ieșirea $y(n)$:

$y(n) = \sum_{i=0}^k a_i \cdot x(n-i)$	(2.4)
--	-------

Fie $w(n)$ un zgomot aditiv aplicat pentru generarea semnalului observat $r(n)$. În majoritatea studiilor $w(n)$ este modelat ca un zgomot alb.

Problema egalizării constă în determinarea unei estimări cât mai exacte a semnalului de intrare $x(n)$, utilizând informația reprezentată de semnalul recepționat, $r(n)$ și semnalul întârziat $x(n-d)$. În cel mai simplu caz, cel liniar, un egalizor poate fi implementat ca un filtru liniar transversal. Fie M ordinul filtrului, vectorul ponderilor $c = [c_0 \ c_1 \ c_2 \ \dots \ c_M]$, d întârzierea și semnalul recepționat $r = [r(n) \ r(n-1) \ \dots \ r(n-M)]$. Trebuie determinată coeficienții c_i astfel încât să fie minimizată funcția de eroare:

$J(n) = \frac{1}{2} E \left[x(n-d) - \sum_{k=0}^M c_k x(n-k) \right]^2$	(2.5)
--	-------

unde E este estimarea.

Soluția optimă pentru coeficienții filtrului transversal este:

$c_{op}[n] = R^{-1}(n) \xi(n)$	(2.6)
--------------------------------	-------

unde:

- R este matricea simetrică de corelație a intrării în egalizor;

$$R = E[r^T r] \tag{2.7}$$

- ξ este vectorul de corelație între intrare și semnalul întârziat:
- $r(n)=[r(n) \ r(n-1) \ \dots r(n-M)]$ și $x(n-d)=[x(n-d) \ x(n-d-1) \ \dots x(n-d-M)]$

$\xi = E[r(n)^T x(n-d)]$	(2.8)
--------------------------	-------

Dezvoltând relația (2.5) și ignorând termenii constanți se poate obține:

$J(n) = \frac{1}{2} c^T R(n) c - \xi^T(n) c$	(2.9)
--	-------

Orice metodă de tip gradient descendent este adecvată pentru găsirea minimumului acestei funcții.

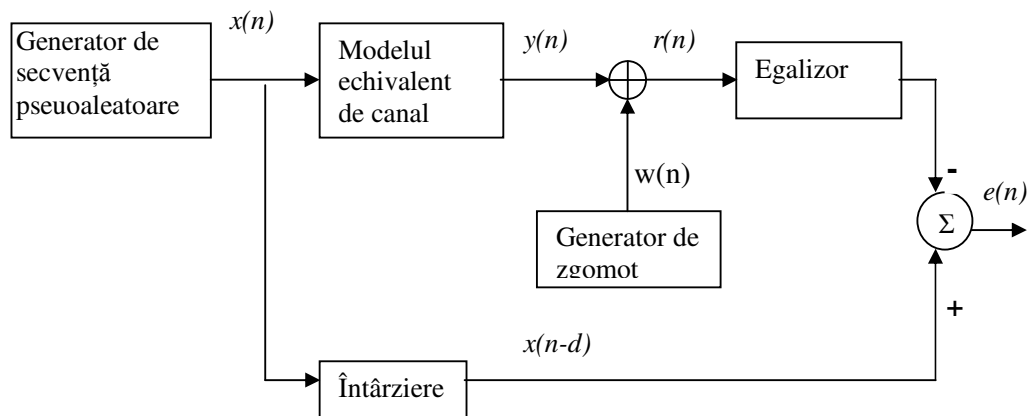


Fig.2.2 Modelarea problemei egalizării

2.5 Modelul canalului de comunicație

În paragraful anterior s-a introdus cel mai simplu model al unui canal de comunicație dispersiv liniar, și anume, filtrul liniar transversal. În situațiile reale canalul introduce însă și distorsiuni neliniare. Utilizarea unor modele adecvate ale canalului, respectiv ale dispozitivelor active de pe calea de transmisie, cât mai apropiate de cele reale este esențială în dezvoltarea sistemelor de comunicație ale viitorului.

Fig.2.3 reprezintă un model al canalului de comunicație care introduce atât distorsiuni liniare (L) cât și distorsiuni neliniare (NL). Literatura de specialitate menționează variate modele cu diferite liniarități, respectiv neliniarități.

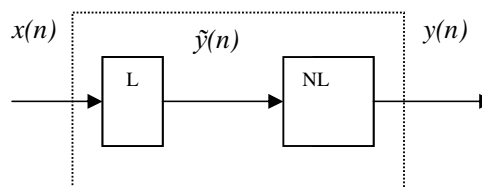


Fig.2. 3 Modelul neliniar al unui canal de comunicație

Uzual, partea liniară este modelată cu un filtru liniar transversal , cu valori reale și nenule ale coeficienților a_0, a_1, \dots, a_k . Semnalul aleator de intrare generează ieșirea conform relației (2.4):

$\tilde{y}(n) = \sum_{i=0}^{k-1} a_i x(n-i)$	(2.10)
--	--------

Pentru semnale complexe se utilizează modele de canal complexe . Modelul sugerat în [I.Cha, 95] generează ieșirea conform relației:

$\tilde{y} = (0.34 - 0.27j)x(n) + (0.87 + 0.43j)x(n-1) + (0.34 - 0.21j)x(n-2)$	(2.11)
--	--------

Partea neliniară este de ordinul al treilea și furnizează ieșirea:

$y(n) = \tilde{y}(n) + 0.1[\tilde{y}(n)]^2 + 0.05[\tilde{y}(n)]^3$	(2.12)
--	--------

Un alt model [S.Chen, 94] se bazează pe relațiile următoare:

$\tilde{y} = (0.7409 - 0.7406j)x(n) - (0.8890 - 0.2961j)x(n-1) + (0.1556 - 0.0223j)x(n-2)$	(2.13)
--	--------

$y(n) = \tilde{y}(n) - 0.055[\tilde{y}(n)]^2 + 0.14[\tilde{y}(n)]^3$	(2.14)
--	--------

Un dispozitiv activ neliniar, tubul cu undă progresivă poate fi modelat printr-o amplificare complexă :

$G(r) = A(r)e^{j\phi(r)}$	(2.15)
---------------------------	--------

unde r^2 este puterea momentană a semnalului și pentru amplitudine, respectiv fază se folosește modelul lui Saleh [M.Ibnkahla, 04]:

$A(r) = \frac{2}{1+r^2}$	(2.16)
--------------------------	--------

$\phi(r) = \frac{4.0033r^2}{1+9.104r^2}$	(2.17)
--	--------

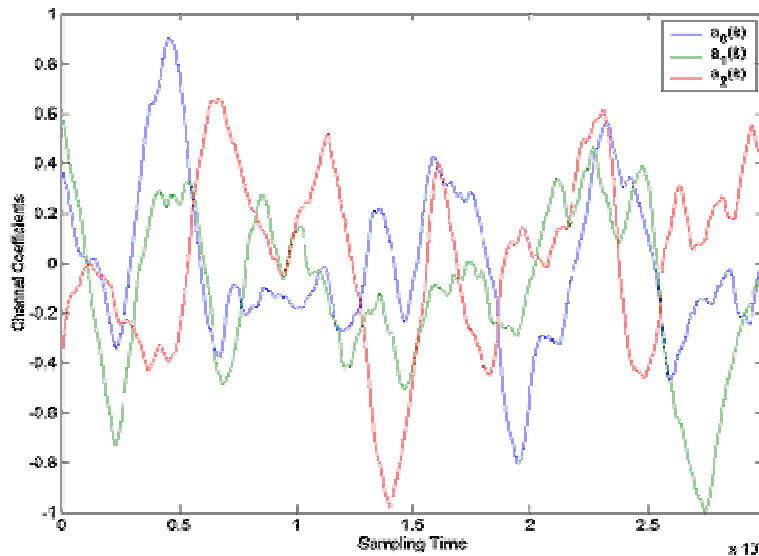


Fig.2.4 Coeficienții unui canal de comunicație variant în timp

Modelarea unui canal neliniar variant în timp, ca de exemplu un canal ai cărui coeficienți variază în timp (fading channel) se poate face cu următoarea funcție de transfer:

$H(z) = a_0(k) + a_1(k).z^{-1} + a_2(k).z^{-2} + \dots + a_n(k).z^{-n}$	(2.18)
---	--------

unde coeficienții $a_i(k)$ variază cu timpul k și poți fi generați prin trecerea unui zgomot gaussian printr-un filtru Butterworth . Banda filtrului determină banda relativă (fading rate) a canalului. Presupunând că avem un canal cu o frecvență de 2 kHz, cu o viteză de transmisie de 2400 simboluri pe secundă și un filtru Butterworth de ordinul 2 cu o bandă de 0.5 Hz la 3 dB, curbele răspunsului la impuls sunt prezentate în Fig. 2.4.

2. 6 Rețele neuronale multistrat

Incapacitatea rețelelor cu un strat adaptiv de a transforma modele arbitrare de intrare în modele arbitrare de ieșire, a fost depășită utilizând un strat intermediar, numit ascuns între intrare și ieșire.

În forma lor generală, rețelele neuronale multistrat RNM, (în engleză multilayer perceptrons) Fig.2.5 au un strat de intrare, un număr de straturi ascunse și un strat de ieșire. Când informația se propagă prin rețea de la intrare spre ieșire aceste rețele se numesc de tip "spre înainte" (feed-forward).

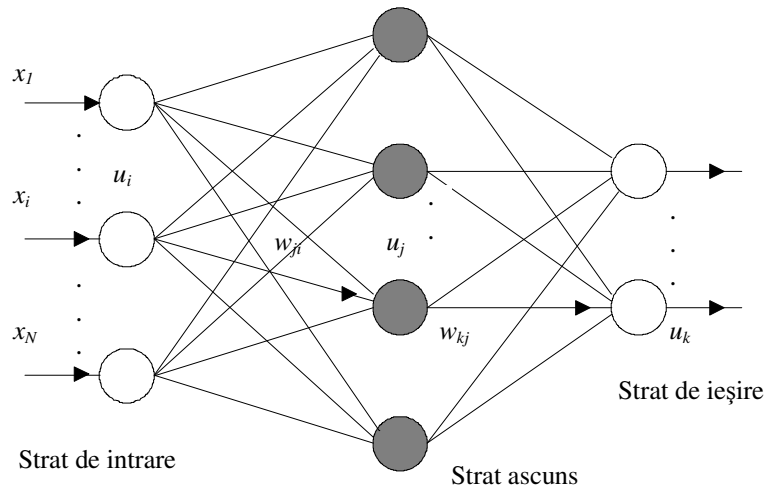


Fig.2. 5 Arhitectura unei rețele neuronale multistrat

Pentru majoritatea aplicațiilor este suficientă o funcție de activare liniară , cel mult semiliniară, pentru neuronii stratului de ieșire, ca de exemplu:

$a_i(t+1) = f(net) = \begin{cases} 0, & \text{pentru } net(t) < -\theta \\ \frac{net(t) + \theta}{2\theta}, & \text{pentru } -\theta < net < \theta \\ 1, & \text{pentru } net(t) > \theta \end{cases}$	(2.19)
---	--------

Pentru a se depăși performanțele rețelelor cu un strat, este însă necesar ca funcția de activare a unităților ascunse să fie cel puțin semiliniară. Uzual ea este o funcție așa zisă sigmoidă, dată de relația:

$O_{pj} = \frac{1}{1 + e^{-\beta(\sum_i w_{ji} \cdot x_{pi}(t) + \theta_j)}}$	(2.20)
---	--------

unde β este un factor de proporționalitate.

Avantajul unei astfel de funcții este calculul simplu al derivatei:

$f'(x) = f(x)[1 - f(x)]$	(2.21)
--------------------------	--------

Adesea se utilizează funcția tangentă hiperbolică, deoarece are valori în intervalul [-1,+1]:

$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	(2.22)
--	--------

Derivata ei este:

$\frac{d \tanh(x)}{dx} = [\sec h(x)]^2 = \frac{4}{(e^x + e^{-x})^2}$	(2.23)
--	--------

Pentru antrenarea rețelei MLP se utilizează uzual algoritmul cu "propagare inversă a erorii".

2.6.1 Algoritmul propagării inverse a erorii

Algoritmul propagării inverse a erorii sau retropropagării ereorii RPE (back-propagation) a fost fundamentat independent de mai mulți cercetători din domeniul analizei numerice (Bryson & Ho, 1969) și al statisticii (Werbos în 1974) până în cel al RN (Parker 1982 , Le Cun 1986, Rumelhart, Hinton & Wiliam 1986).. Algoritmul este o învățare cu control în două faze și este cunoscut și sub denumirea de "regula delta generalizată", denumire introdusă în "Parallel Distributed Processing" de grupul creat de Rumelhart și Mc Clelland .

Prima etapă

În prima faza, modelul p de intrare se propagă prin rețea din strat in strat, până la ieșire.

Fie notațiile:

N_{input} : numărul intrărilor în RN (care dă dimensiunea vectorilor de intrare);

N_h : numărul unităților stratului ascuns;

N_{output} : numărul unităților stratului de ieșire;

Intrarea netă în fiecare neuron ascuns este pentru fiecare model de intrare p:

$net_{pj} = \sum_{i=1}^{N_{input}} w_{ji} \cdot X_i + \theta_j$	(2.24)
---	--------

Ieșirea fiecărui neuron ascuns este o funcție de suma ponderată a intrărilor și pragul θ :

$o_j = f\left(\sum_{i=1}^{N_{input}} w_{ji} X_i + \theta_j\right) \quad j=1, \dots, N_h$	(2.25)
--	--------

În cazul unei rețele neuronale cu un singur strat ascuns, ieșirea o_k a neuronului k de ieșire se exprimă în funcție de informația primită din stratul ascuns cu relația:

$o_k = f\left(\sum_{j=1}^{N_h} w_{kj} o_j + \theta_k\right) \quad \text{ou} \quad net_k = \sum_{j=1}^{N_h} w_{kj} o_j + \theta_k \quad k=1, \dots, N_{output}$	(2.26)
--	--------

unde vectorul $w_k = (w_{k1} \ w_{k2} \dots w_{kj})$ este vectorul ponderilor conexiunilor neuronului k de ieșire. Funcția globală realizată de RN la ieșirea neuronului k este deci:

$o_k = f\left(\sum_{k=1}^{N_h} w_{kj} f\left(\sum_{i=1}^{N_{input}} w_{ji} x_i + \theta_j\right) + \theta_k\right) \quad k=1, \dots, N_{output}$	(2.27)
--	--------

Pentru neuronii unui strat funcția de transfer este în general aceeași.

Ieșirea curentă o_k se compară cu intrarea de control t_{pk} , generând, în unitățile de ieșire, o eroare δ_{pk} :

$\delta_{pk} = (t_{pk} - o_{pk}) \cdot f'(net_{pk})$	(2.28)
--	--------

unde f' este derivata funcției de activare a neuronului.

Etapa a doua

În faza a doua, erorile se propagă de la ieșire spre intrare, din strat în strat determinând schimbarea ponderilor conexiunilor în sensul minimizării erorii la nivelul fiecărui neuron în parte. Regula de învățare pentru conexiunile dintre neuronii de ieșire și cei ascunși este:

$\Delta_p w_{kj} = \eta \cdot \delta_{pk} \cdot o_{pj}$	(2.29)
---	--------

unde η este constanta de învățare care poate lua valori în intervalul (0,1).

Pentru unitățile ascunse, indexate după j , erorile δ_{pj} se calculează cu ajutorul erorilor neuronilor de ieșire δ_{pk} cu relația:

$\delta_{pj} = \left(\sum_k \delta_{pk} \cdot w_{kj}\right) \cdot f'(net_{pj})$	(2.30)
---	--------

Apoi se modifică ponderile conexiunilor dintre neuronii ascunși și intrări cu relația:

$\Delta_p w_{ji} = \eta \cdot \delta_{pj} \cdot o_{pi}$	(2.31)
---	--------

Dacă în RN există mai multe straturi ascunse, erorile se evaluează pentru fiecare strat cu relația (2.28) și apoi se determină noile ponderi dintre stratul anterior și cel succesiv. În rețea pot exista și ponderi fixe. Dacă există neuroni de ieșire și în straturile ascunse, acestea însumează două tipuri de erori: erori rezultate din compararea ieșirii cu răspunsul dorit și erori obținute prin propagare de la neuronii de ieșire spre neuronii cu care sunt cuplați.

Observație

Această procedură minimizează eroarea pătratică medie în fiecare iterație. Studiile arată că RN converge în general spre un minim local, care reprezintă în unele cazuri o soluție acceptabilă. În literatura de specialitate sunt prezentate mai multe metode pentru evitarea minimelor locale, aspect care va fi abordat ulterior.

Se poate demonstra că algoritmul implementează un gradient descendent al erorii totale în spațiul ponderilor, adică:

$w(t+1) = w(t) - \eta \cdot \frac{dE}{dw}$	(2.32)
--	--------

unde dE/dw este gradientul aleatoriu necunoscut al erorii totale dintre modelele de intrare și modelele de ieșire dorite. White a demonstrat că relația (14) este o aproximare stohastică.

Observații

Algoritmul presupune o etapă de antrenament, una de testare și apoi cea de utilizare. Crearea bazei de date și selectarea datelor de antrenament, respectiv de testare, au o influență majoră în succesul sau eșecul funcționării unei RN. În timpul antrenamentului, fiecare pereche p de model de intrare-model de ieșire dorit, este prezentată repetat. Necesitatea unei prezentări repetate a modelelor în timpul antrenamentului reprezintă una dintre carențele algoritmului cu propagarea inversă a erorii, deoarece conduce la un timp îndelungat afectat antrenamentului RN. În timpul testării se aplică doar modelele de intrare, verificându-se statistic corectitudinea funcționării. În cazul în care performanța obținută este acceptabilă, RN poate fi utilizată.

2.7 Aspecte importante ale algoritmului cu retropropagare a erorii

Vom prezenta în continuare câteva dintre elementele care influențează semnificativ performanțele algoritmului retropropagării erorii, ilustrând cauzele unor posibile rezultate nesatisfăcătoare și oferind câteva indicații utile în aplicațiile practice.

2.7.1 Crearea bazei de date

Crearea bazei de date solicită cel mai îndelungat efort în dezvoltarea unei aplicații și are o importanță majoră în succesul sau eșecul acesteia. Această etapă presupune:

- adunarea datelor;
- analiza datelor;
- alegerea variabilelor;
- preprocesarea variabilelor de intrare, astfel încât RN să poată învăța eficient;

Din practica curentă specialiștii afirmă că 9/10 din dezvoltarea unei aplicații este soluționată prin colectarea datelor adecvate. Selectarea datelor de intrare implică adesea alegerea dintre mai multe variabile, astfel încât familiaritatea cu domeniul aplicației este de neprețuit. Specialiștii în RN colaborează cu experții în domeniul aplicației pentru dezvoltarea RN.

Analiza datelor de intrare se face din perspectiva tehnicilor statistice. Corelația dintre o anumită intrare și o anumită ieșire poate sugera includerea sau excluderea variabilei. O corelație puternică dintre două variabile poate duce la eliminarea uneia dintre ele. După alegerea celor mai semnificative intrări se pune problema examinării distribuției lor și în general se elimină valorile în afara distribuției tipice. Este de dorit ca perechile de date intrare-ieșire folosite în procesul de antrenare să fie cât mai diferite, pentru ca rețeaua să aibă la dispoziție un număr de "scenarii" cât mai mare pentru problema concretă considerată. În setul de antrenare se inserează câteodată (în special atunci când baza de date este redusă sau puternic redundantă) și date provenind din suprapunerea unor nivele de zgomot peste valorile originale, ceea ce are ca efect îmbunătățirea capacității de generalizare a rețelei.

Analiza datelor poate duce la identificarea de tendințe, cicluri sau alte relații care pot fi extrase prin preprocesare.

Preprocesarea transformă datele astfel încât ele să poată fi mai ușor învățate de RN. Preprocesarea poate implica calculul de sume, inverse, diferențe, derivate, puteri, radicali, medii,

transformate Fourier sau extragere de caracteristici. O RN poate pregăti datele de intrare pentru o altă RN, realizând de exemplu o grupare, înainte de o clasificare mai rafinată.

Cantitatea datelor suficiente este o problemă complexă, uneori afectată de considerații practice, ca de exemplu costul colectării datelor. Datele de antrenament trebuie să asigure un eșantion reprezentativ și suficient pentru un antrenament corect. Seturi mari de date reduc riscul sub eșantionării funcției de învățat, dar în acest caz RN necesită un timp îndelungat de antrenare. Seturi mici de date implică o antrenare rapidă, dar rețeaua poate eșua în faza de utilizare.

Suficiența datelor depinde mai mulți factori, ca de exemplu: mărimea RN, distribuția intrărilor și ieșirilor, cerințele testării. Din practica curentă s-a ajuns la concluzia că pentru antrenament eficient sunt suficiente aproximativ 10 modele pentru fiecare interconexiune a rețelei. O altă regulă empirică apreciază că trebuie să existe următoarea relație între dimensiunea bazei de date considerate D , folosite în procesul de învățare, numărul ponderilor N_w și valoarea finală a erorii pătratice E :

$D = \frac{N_w}{E(\infty)}$	(2.31)
-----------------------------	--------

Împărțirea judicioasă a bazei de date, într-un set de antrenare și altul de validare, este hotărâtoare în obținerea unor performanțe de generalizare satisfăcătoare. Există rezultate teoretice care justifică alegerea dimensiunii setului de date de antrenament la aproximativ 90% și a celui de validare la aproximativ 10% din baza de date originală.

2.7.2 Modalități de antrenare

Strategia uzuală presupune antrenarea RN multistrat "off-line", cu date disponibile de la început. crearea bazei de date

Există două metode de antrenare "off-line", pentru o RN multistrat cu algoritmul retropropagării erorii:

1. Modul "model cu model" (în engleză "pattern by pattern mode") constă în modificarea setului de ponderi după câte o prezentare a unei singure perechi de date model de intrare- model de ieșire. Se recomandă o prezentare aleatoare a datelor de antrenament, pentru a evita ca rețeaua să considere eronat că acestea prezintă un caracter periodic intrinsec. Există indicații empirice că viteza de convergență a algoritmului este mai mare în acest caz.
2. Modul "tampon" (în engleză "batch mode") în care ponderile sunt ajustate după prezentarea întregului set de date de antrenament. Aplicarea datelor de antrenament o singură dată se numește epocă de antrenare. Varianta optimă de antrenare depinde de aplicația concretă.

Antrenarea în timp real, "on-line" este mai apropiată de condițiile reale de utilizare a RN. Datele de antrenare sunt obținute în timp real din măsurători curente ale procesului de modelat și determină modificarea imediată a parametrilor RN. Dezavantajul constă în faptul că în urma unei modificări radicale a parametrilor, la prezentarea unei perechi model de intrare- model de ieșire dorit, RN își poate pierde capacitatea de interpolare a unor date ulterioare. Dacă dimpotrivă, noua asociere este prea aproape de cea anterioară, modificarea parametrilor poate fi nesemnificativă.

2.7.3 Generalizarea

Odată antrenată RN, se pune problema cât de bine generalizează. A generaliza, conform dicționarului explicativ, înseamnă a formula principii sau concluzii din detalii obținute prin experiență. Abilitatea unei RN de a generaliza se referă la identificarea de reguli ce-i permit să facă predicții corecte asupra unor date necunoscute. Capacitatea de generalizare a unei RN se verifică pe setul de date de testare. Factorii care o influențează sunt numărul eșantioanelor de date, complexitatea problemei, mărimea rețelei. Există studii care stabilesc limita inferioară și superioară a capacității de generalizare a unei RN (măsură cunoscută sub denumirea de "dimensiunea Vapnik-Chervonenkis"), în funcție de dimensiunea modelelor de intrare, numărul neuronilor ascunși, numărul numărul total al neuronilor RN, numărul ponderilor RN

2.7.4 Criterii de oprire a învățării

Există mai multe criterii de oprire a antrenării unei RN multistrat cu algoritmul RPE:

- După un număr de iterații predefinit, dar există riscul ca eroarea să nu scadă suficient;
- La o valoare impusă a erorii, dar numărul de iterații ar putea fi prea mare;
- La o valoare dată a variației erorii pentru două iterații consecutive. Metoda este inadecvată pentru situațiile în care eroarea are paliere de-a lungul cărora se modifică foarte puțin, deși în valoare absolută este considerabilă.
- Validarea prin intersecția datelor ("cross-validation"). Se antrenează RN cu datele de antrenament, timp în care eroarea descrește continuu. Se verifică funcționarea RN cu datele de testare, timp în care eroarea descrește, apoi crește din nou, ca în Fig.2.6. În acest interval, numit de suprapotrivire ("overfitting"), RN nu mai modelează procesul care a generat asocierile model de intrare- model de ieșire dorit, ci învață chiar datele. Ponderile RN se îngheață la un număr de iterații optim, considerat în momentul în care eroarea din faza de testare începe să crească.

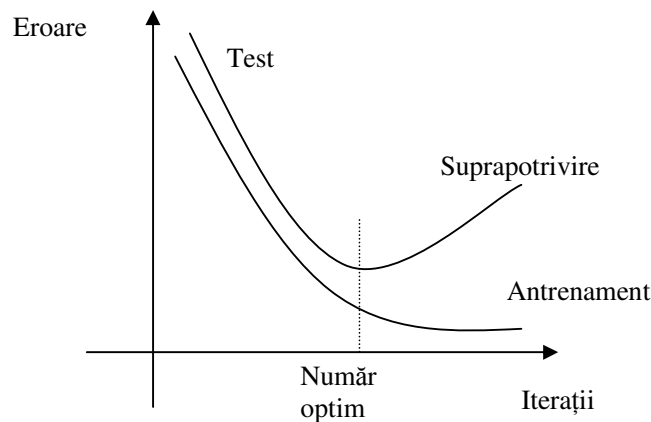


Fig.2.6 Evoluția erorii în funcție de numărul de iterații pentru faza de antrenament, respectiv testare

2.7.5 Funcția de activare

Utilizarea unei funcții de activare neliniare conferă unei RN avantajul unui potențial de procesare superior comparativ cu cazul utilizării unei funcții liniare. Astfel o RN cu funcție de activare neliniară va necesita un număr mai mic de neuroni ascunși, chiar de straturi decât o RN cu o funcție de activare liniară pentru a rezolva o aceeași problemă. Ușual, funcția de activare a unui neuron este o funcție neliniară bipolară (2.32) sau unipolară (2.33), funcții de inspirație biologică:

$$f(x) = \tanh(ax) = \frac{1 - e^{-ax}}{1 + e^{-ax}} \quad (2.34)$$

$$f(x) = \frac{1}{1 + e^{-ax}} \quad (2.35)$$

S-au observat creșteri semnificative ale vitezei de convergență în cazul utilizării unei funcții de activare bipolare. Ajustarea parametrului a , respectiv scăderea sa în timp, are efecte benefice, în special în faza inițială a procesului de învățare.

Un avantaj major al acestor funcții îl constituie simplitatea obținerii derivatelor de ordinul I cu relația ($\frac{df(x)}{dx} = af(1-f)$), care permite evaluarea rapidă a erorilor δ în algoritmul RPE. Găsirea tipului

adecvat de neliniaritate pentru o aplicație dată conferă forță de calcul superioară unei rețele neuronale.

În aplicațiile de clasificare se utilizează adeseori funcția denumită softmax, care permite interpretarea ieșirilor rețelei neuronale ca probabilități condiționate.

$f(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$	(2.36)
---	--------

Derivata funcției de activare are un efect semnificativ asupra vitezei de convergență. Observația conform căreia variația ponderilor, direct proporțională cu derivata funcției de activare, este neglijabilă pentru semnale mari (pentru care derivata se anulează) a impus creșterea artificială a valorii derivatei. Prin adăugarea unei constante de 0.1 s-a redus în unele situații timpul de procesare la jumătate.

2.7.6 Funcția de eroare

Funcția de eroare în cazul unei rețele neuronale multistrat este o funcție multidimensională neliniară ai cărei parametri sunt valorile interconexiunilor și pragurile neuronilor. Un algoritm de antrenare oarecare urmărește modificarea acestor parametri în sensul minimizării erorii.

Într-o formă generală, pentru o RN cu N neuroni, funcția de eroare poate fi dată de relația:

$E = \sum_p \sum_{j=1}^N o_j(X_p; W) - d_{jp} ^R$	(2.37)
--	--------

- d_{jp} este răspunsul dorit pentru modelul p de intrare;
- o_j este ieșirea;
- W reprezintă totalitatea parametrilor rețelei (ponderi și praguri);
- X_p este modelul de intrare p;

Pentru R=2 se obține eroarea pătratică medie.

Observații:

- Eroarea pătratică medie este extrem de sensibilă la prezența unor erori individuale mari (modele numite în engleză outliers). Performanțe superioare se obțin pentru alte tipuri de distanțe metrice, ca de exemplu, pentru distanța Manhattan, care are R=1.
- Funcția de eroare definită anterior este specifică modului de antrenare de tip tampon (batch) (ajustarea parametrilor rețelei se face după fiecare prezentare integrală a bazei de date disponibile). Reprezentarea geometrică a funcției de eroare pune în evidență existența unui minim global și a mai multor minime locale, ca în Fig.2.7. Este important de observat că aspectul funcției de eroare este independent de algoritmul de învățare. În cazul unei funcții de activare neliniare suprafața funcției de eroare nu este concavă, astfel încât există posibilitatea ca algoritmul de antrenament să se oprească într-unul dintre minimele locale. În acest minim local eroarea poate să fie mare. Pentru unele aplicații oprirea într-unul dintre minimele locale reprezintă o soluție a problemei, din moment ce eroarea obținută este acceptabilă. Un exemplu este implementarea funcției SAU EXCLUSIV

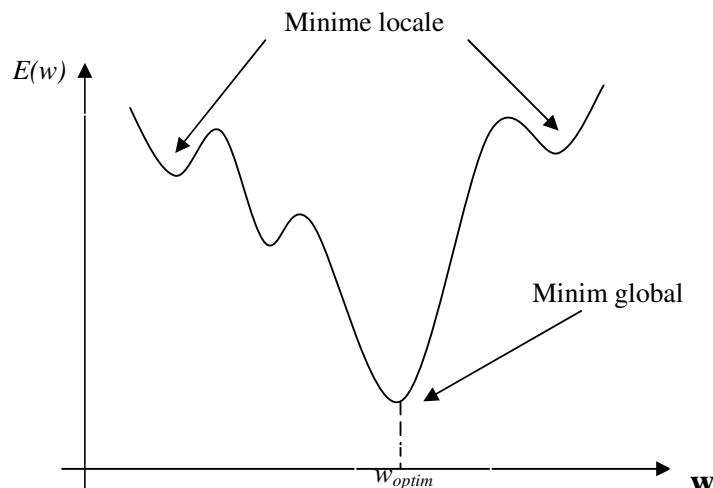


Fig.2.7 Aspect tipic al funcției de eroare

- Observând natura statistică a algoritmului retropropagării erorii White a propus tehnici de optimizare ale statisticii robuste pentru a crește insensibilitatea RN la perturbații. Eroarea se poate înlocui cu o funcție de eroare, ca de exemplu: $\tan(e_i/2)$, $e_i/(1+e_i)^2$, $\max[-e_i, \min(e_i, x_i)]$. În alegerea funcției de eroare se apeleză adesea la metoda încercării.

Tehnicile de optimizare utilizate în prezent se încadrează într-una dintre următoarele două categorii:

a) Metode în care funcția de eroare descrește sau rămâne constantă de la o iterație la alta, fără posibilitatea de a crește temporar. Dezavantajul acestora este imposibilitatea de a "evada" din minimele locale. Exemple din această categorie sunt algoritmul RPE de tip gradient conjugat, algoritmul RPE de tip quasi-Newton, precum și varianta tampon a algoritmului RPE cu scădere după gradient.

b) Metode în care eroarea evoluează în medie către valoarea minimă, permițând creșteri temporare ale valorii acesteia. Din această categorie sunt varianta model cu model a algoritmului RPE cu scădere după gradient și varianta RPE cu impuls.

Unele dintre metodele performante de optimizare neliniară, ca de exemplu algoritmul RPE de tip gradient conjugat și algoritmul Levenberg-Marquardt, fac apel la aproximări pătratice locale ale funcției de eroare și utilizează matrici Jacobienne ale derivatelor parțiale de ordinul întâi sau matrici Hessienne ale derivatelor parțiale de ordinul doi

2.7.8 Inițializarea ponderilor și pragurilor

Valorile inițiale ale ponderilor și pragurilor se aleg cu atenție pentru a preîntâmpina saturarea prematură a neuronilor și deci scăderea vitezei de convergență a algoritmului. De obicei acestea se aleg aleatoriu, eventual uniform distribuit într-un interval de valori:

$$[-2,4/\sqrt{F_i} ; 2,4/\sqrt{F_i}]$$

unde F_i este numărul total de intrări în neuron.

2.7.9 Viteza de învățare

Alegerea vitezei de învățare η , în intervalul de valori (0,1) are o importanță deosebită în evoluția procesării.

- O viteză de învățare mare, asigură o convergență rapidă, dar poate determina oscilații ale rețelei;
- O viteză de învățare mică, în intervalul (0.05,0.25) are ca efect mărirea timpului de procesare și poate duce la împotmolirea în minime locale cu o probabilitate mai mare;

Deci, pentru a mări convergența algoritmului de antrenare este necesară creșterea constantei de învățare. Există mai multe metode folosite pentru satisfacerea acestei condiții:

Metoda impulsului

Metoda impulsului a fost introdusă de Rumelhart, Hinton și Williams în 1986 într-o carte de referință a RN, "Parallel Distributed Processing". Prin introducerea unei relații între schimbarea curentă a ponderii și modificarea ei anterioară se pot asigura pași mai mari în procesare (convergență mai rapidă) și se evită oscilațiile rețelei la o viteză mare de învățare:

$$\Delta_p w_{ij}(t+1) = \eta \cdot \delta_{pj} \cdot o_{pi} + \alpha \cdot \Delta_p w_{ij}(t) \quad (2.38)$$

unde α este o constantă, numită impuls (în engleză momentum), ce determină efectul ponderilor anterioare asupra ponderii curente. Dacă α este zero se poate regăsi în relația (2.36) variația utilizată în algoritmul clasic al retropropagării erorii. Dacă variația ponderii are de la o iterație la alta același semn, atunci, în pasul următor, ea va crește cu $\alpha \Delta w_{ij}(t)$. Astfel este favorizată parcurgerea dealurilor în funcția de eroare. Dacă $\Delta w_{ij}(t)$ își schimbă semnul de la o iterație la alta atunci $\Delta w_{ij}(t)$ scade în pasul următor cu $\alpha \Delta w_{ij}(t)$. Astfel metoda are un efect stabilizator, realizând practic o filtrare a variațiilor mari ale ponderilor.

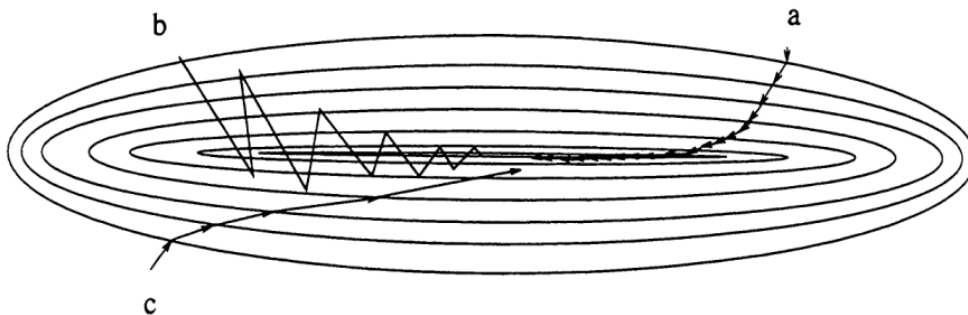


Fig.2.8 Convergența algoritmului de antrenare

- a) pentru o viteză mică de învățare; b) pentru o viteză mare de învățare; c) pentru o viteză mare de învățare și un termen de impuls;

Fig.2.8 prezintă convergența rețelei în spațiul ponderilor în trei cazuri posibile

- a) pentru o viteză mică de învățare, fără impuls, când este necesar un timp îndelungat pentru atingerea minimului;
- b) pentru o viteză mare de învățare, fără impuls, când minimul nu poate fi atins datorită oscilațiilor rețelei;
- c) pentru o viteză mare de învățare și un termen de impuls când minimul este atins rapid;

Metoda netezirii

Este o generalizare a metodei impulsului, introdusă de Sejnowski și Rosenberg în 1987, care utilizează relația:

$\Delta_p w_{ij}(t+1) = \eta.[b.\Delta_p w_{ij}(t) + (1-b).\delta_{pj}.o_{pj}]$	(2.39)
---	--------

Dacă:

- constanta $b=0$, atunci relația (2.37) este forma standard a algoritmului RPE ;
- $b=1$, atunci ponderile rămân neschimbate;
- $b \in (0,1)$ atunci variația ponderii este "netezită" cu constanta b ;

Metoda vitezei de învățare variabile

Pentru a evita oscilațiile RN, care pot apare în jurul soluției optime, se poate alege o viteză de învățare descrescătoare în timp. Descrșterea rapidă poate fi asigurată de condiția :

$\lim_{k \rightarrow \infty} \sum_{k=1} (\eta[k])^2 < \infty$	(2.40)
---	--------

unde $\eta[k]$ este viteza de învățare la momentul k

Dacă descresțerea vitezei de învățare este prea rapidă, există riscul ca ponderile să uite modelele deja învățate. Astfel încât viteza de învățare trebuie să descrescă suficient de lent, satisfăcând relația :

$\lim_{k \rightarrow \infty} \sum_{k=1} (\eta[k])^2 = \infty$	(2.41)
---	--------

Dacă $\eta[k]=1/k$ condițiile (2.38) și (2.39) sunt simultan satisfăcute. Cele două condiții reprezintă așa numita dilemă plasticitate-stabilitate, descoperită de Grossberg, care poate fi formulată astfel :

O RN trebuie să satisfacă două condiții contradictorii :

- 1. Trebuie să fie suficient de stabilă pentru a-și aminti modelele anterior învățate ;**
- 2. Trebuie să fie suficient de plastică pentru a putea învăța modele noi ;**

Satisfacerea celor două condiții menționate anterior nu garantează convergența algoritmului RPE, ci asigură doar o comportare "rezonabilă".

2.7.10 Problema minimelor locale

Există mai multe metode pentru evitarea minimelor locale:

- schimbarea constantei de învățare;
- schimbarea ponderilor inițiale;
- modificarea numărului de neuroni ascunși;

Adăugarea unor valori aleatoare mici ponderilor, procedură cunoscută sub numele de "scuturarea rețelei", permite RN ieșirea dintr-un minim local. Dacă noua stare este suficient de îndepărtată de minimul local, procesarea se poate desfășura într-o nouă direcție, fără a exista riscul revenirii în minim.

Dacă performanța obținută într-un minim local este acceptabilă, acesta poate fi considerat o soluție a problemei.

2.7.11 Arhitectura

Prin alegerea adecvată a arhitecturii rețelei se poate diminua probabilitatea apariției unor minime locale ale funcției de eroare. Pentru alegerea arhitecturii optime a unei RN pentru o aplicație dată, respectiv pentru determinarea numărului de straturi ascunse necesare și a numărului neuronilor dintr-un strat, nu există criterii cantitative sau reguli, ci doar reguli empirice. Probabil că acest aspect nu se va rezolva în cazul general, din moment ce fiecare aplicație are alte cerințe pentru rețea. Numărul neuronilor trebuie să fie suficient de mare pentru a genera o configurație a regiunilor de decizie suficient de complexă, pentru o problemă dată.

Dimensiunea RN trebuie judicios aleasă astfel încât să nu fie subdimensionată sau supradimensionată. O rețea neuronală prea mică nu va fi capabilă să generalizeze în mod adecvat. O rețea neuronală prea mare ar putea fi prea capabilă, prezentând dezavantajul manipulării greoaie. Numărul conexiunilor ar fi în acest caz foarte mare și pentru determinarea ponderilor ar fi necesar un număr foarte mare de date, respectiv un timp îndelungat de antrenament. Pentru determinarea mării optime se poate recurge la metoda empirică a încercărilor.

Literatura de specialitate consemnează mai multe modalități de determinare a mării optime a unei rețele:

- Se construiește cea mai mică rețea posibilă pentru aplicația în cauză și apoi i se adaugă treptat neuroni, până când performanța realizată este satisfăcătoare;
- Se construiește o rețea mare și apoi se elimină treptat neuronii inutili, verificându-se de fiecare dată performanța rețelei ;

Se apreciază că numărul de neuroni din primul strat ascuns, trebuie de trei ori mai mare decât numărul neuronilor din al doilea strat ascuns pentru ca acest strat să furnizeze trei sau patru laturi pentru fiecare regiune identificată de al doilea strat ascuns. În general structura unei RN multistrat trebuie să fie una de tip compresie, de la un număr mai mare de neuroni spre un număr redus de neuroni. În cazul în care problema de instruire presupune divizarea spațiului vectorilor de intrare în m clase, RN trebuie să aibă m ieșiri, fiecare ieșire corespunzând unei clase.

În unele cazuri este necesară o creștere exponențială a numărului de straturi pentru a obține o creștere liniară a vitezei de învățare. Există însă și probleme pentru care viteza de instruire descrește cu mărirea numărului de straturi.

2.8 Egalizoare implementate cu rețele neuronale multistrat

Structura unui egalizor implementat cu o rețea neuronală multistrat , cu un strat ascuns, este dată în Fig.2.9.

Structura unui egalizor neliniar pentru semnale complexe, implementat cu o rețea neuronală multistrat este prezentată în Fig.2.10. Se observă că partea reală este abordată separat de cea imaginară a semnalului de intrare. Variabilele de intrare în rețeaua neuronală sunt eșantioanele întârziate ale semnalului de intrare. Rețeaua neuronală va avea cel mult două straturi ascunse, întrucât se știe că o rețea neuronală cu două straturi ascunse poate implementa orice transformare neliniară cu precizia dorită dacă numărul neuronilor ascunși este adecvat ales. Stratul de ieșire va avea un neuron pentru partea reală, respectiv unul pentru partea imaginară. Ieșirea acestuia se compară cu semnalul dorit, adică semnalul de intrare întârziat. Funcția de activare este uzual o funcție neliniară de tip sigmoidă . Pentru determinarea coeficienților filtrului liniar transversal și a rețelei neuronale se utilizează în mod uzual algoritmul retropropagării erorii. Algoritmul este iterativ și minimizează orice funcție de cost diferențibilă, ca de exemplu eroarea medie pătratică.

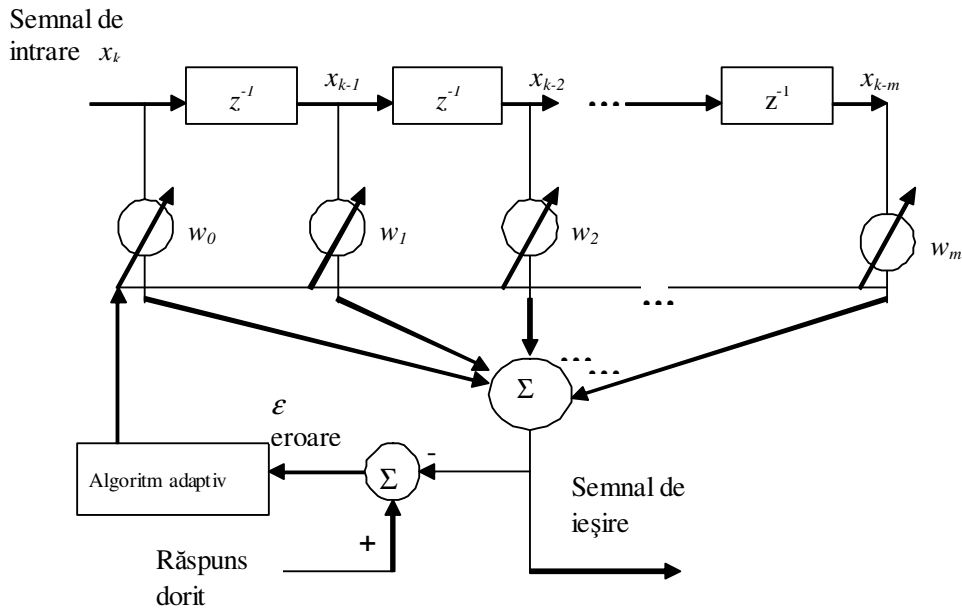


Fig.2. 9 Egalizor implementat cu o rețea neuronală multistrat , cu un strat ascuns

În cazul unor semnale complexe este necesară pentru antrenare o variantă complexă a acestui algoritm. (CBKP complex backpropagation), care are și el aceleași curențe ca și varianta clasică, reală. Algoritmii de tip gradient, prezintă dezavantajele timpului îndelungat de antrenare și a atragerii soluției în minimele locale ale funcției eroare, adesea nesatisfăcătoare din punctul de vedere al preciziei realizate. O altă problema care poate apare în CBKP este selecția funcției de activare. În cazul semnalelor reale, funcțiile de activare sunt reale și limitate. În cazul semnalelor complexe, cu excepția constantelor, toate funcțiile de activare uzuale ar putea fi nelimitate. Pentru a satisface situația conflictuală între limitarea și diferențiabilitatea unei funcții complexe au fost dezvoltate două variante ale algoritmului CBKP, unul care utilizează o funcție de activare complet complexă [T.Kim, 02](fully complex activation function BKP FCBKP) și alta cu o funcție de activare separată în parte reală și respectiv imaginară [A.Kantszla, 04] (split complex activation function (SCBKP)

Pentru a rezolva problema convergenței lente a fost introdus un alt algoritm de antrenament resilient propagation (CRPROP). Studiile efectuate arată că din punctul de vedere al erorii de bit algoritmul CRPROP are performanțe comparabile cu CBKP, dar în condițiile unei convergențe mult mai rapide și cu un cost computațional semnificativ redus [].

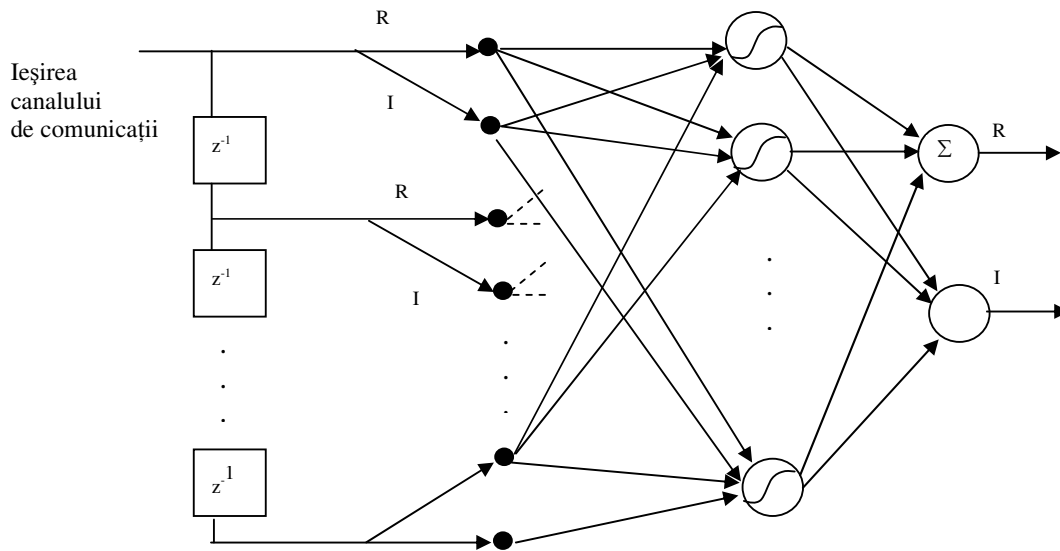


Fig.2.10 Structura unui egalizor complex implementat cu o rețea neuronală multistrat

2.9 Rezultate experimentale

Au fost testate mai multe arhitecturi de egalizoare cu RN multistrat, pentru mai multe modele de canale de comunicație, pentru diferite valori ale dispersiei zgomotului alb aplicat semnalului de la ieșirea canalului de comunicație și diferite variante ale algoritmului retropropagării erorii. Semnalele de intrare au fost semnale 4-QAM complexe.

Câteva dintre rezultatele obținute sunt prezentate în Tab.2.1, în care canalul 1 se referă la modelul lui Cha și Kassam, dat de relația (2.10) și canalul 2 se referă la modelul lui Chen și Mulgrew, dat de relația (2.12)

Fig.2.11 prezintă evoluția erorii pentru 350 de epoci, pentru o rețea multistrat cu o intrare, 15 neuroni ascunși cu o funcție de activare tangentă hiperbolică ('tansig'), funcția de activare ieșire: liniară ('purelin'), metoda de antrenament: backpropagation gradient descent cu moment ('traingdm'). S-au folosit 100 semnale de intrare, canalul de comunicație dat de relația (2.12) și o valoare a dispersiei zgomotului alb aplicat semnalului la ieșirea canalului de comunicație $\sigma^2=0.01$.

Tabelul 2.1

Nr. secv. intrare	Tip canal	Dispersie zgomot alb σ^2	Nr. neuroni strat ascuns	Funcție activare strat ascuns	Funcție activare ieșire	Metoda de antrenament	Eroarea medie pătratică
100	Canal 1	0.01	15	'tansig'	'purelin'	'traingdm'	0.079917
100	Canal 1	0.01	15	'tansig'	'purelin'	'traingda'	0.0933353

100	Canal 1	0.01	15	'tansig'	'tansig'	'traingdm'	0.00998917
100	Canal 2	0.01	15	'tansig'	'purelin'	'traingdm'	0.0151786
500	Canal 2	0.01	15	'tansig'	'purelin'	'traingdm'	0.013542
100	Canal 1	0.1	15	'tansig'	'purelin'	'traingdm'	0.106008
100	Canal 1	0.1	15	'tansig'	'purelin'	'traingda'	0.0144468
100	Canal 1	0.1	15	'tansig'	'purelin'	'traingdx'	0.00999958
100	Canal 1	0.1	15	'tansig'	'purelin'	'traingdx'	0.0810842
500	Canal 2	0.1	15	'tansig'	'tansig'	'traingdm'	0.242658
200	Canal 1	0.1	4	'tansig'	'purelin'	'traingdm'	0.110038
200	Canal 1	0.1	10	'tansig'	'purelin'	'traingdm'	0.0932373
200	Canal 1	0.1	10	'tansig'	'tansig'	'traingdm'	0.520107
200	Canal 2	0.1	4	'tansig'	'purelin'	'traingdm'	0.111026
200	Canal 2	0.1	10	'tansig'	'purelin'	'traingdm'	0.099854
200	Canal 1	0.1	10	'tansig'	'purelin'	'traingd'	0.0942374

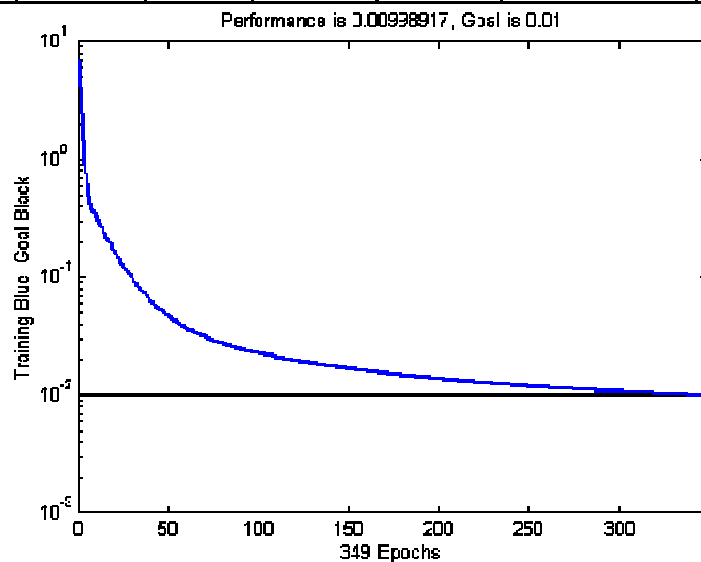


Fig.2.12

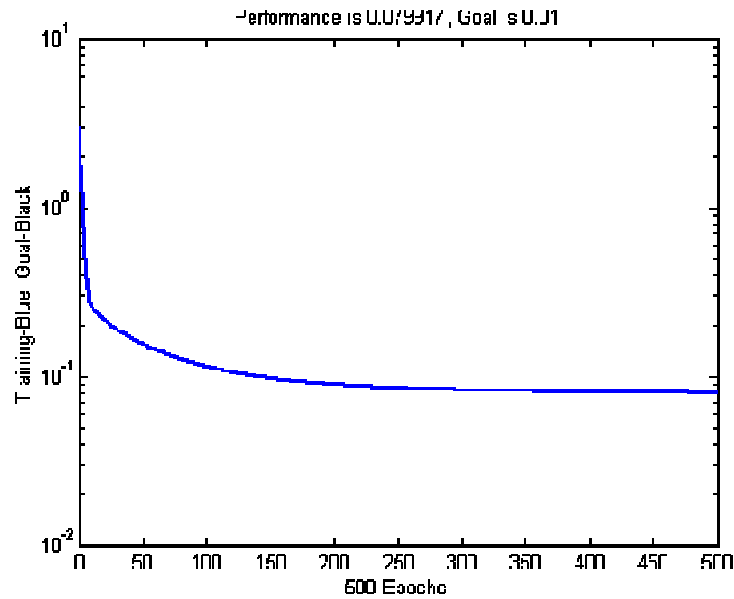


Fig.2.13

Fig.2.13 prezintă evoluția erorii unui egalizor cu aceeași structură și în aceleași condiții ca în cazul anterior, dar o dispersie a zgomotului $\sigma^2=0.1$, într-un interval de timp de 500 de epoci.

Fig.2.14 și Fig. 2.15 prezintă evoluția erorii, pentru o rețea multistrat cu o intrare, 15 neuroni ascunși cu o funcție de activare tangentă hiperbolică ('tansig'), cu o funcție de activare ieșire: liniară ('purelin'), antrenată cu metoda retropropagării erorii de tip gradient descendent ('traingd'), pentru o dispersie a zgomotului $\sigma^2=0.01$, respectiv $\sigma^2=0.1$. S-au folosit 100 semnale de intrare și canalul de comunicație dat de relația (2.12).

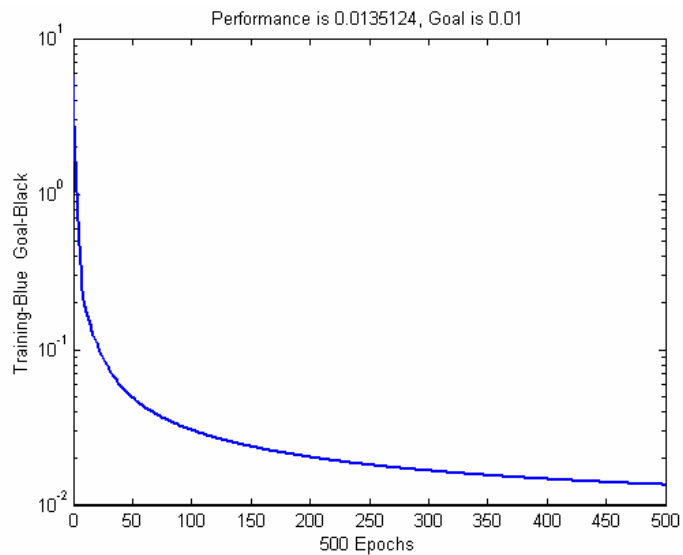


Fig.2.14

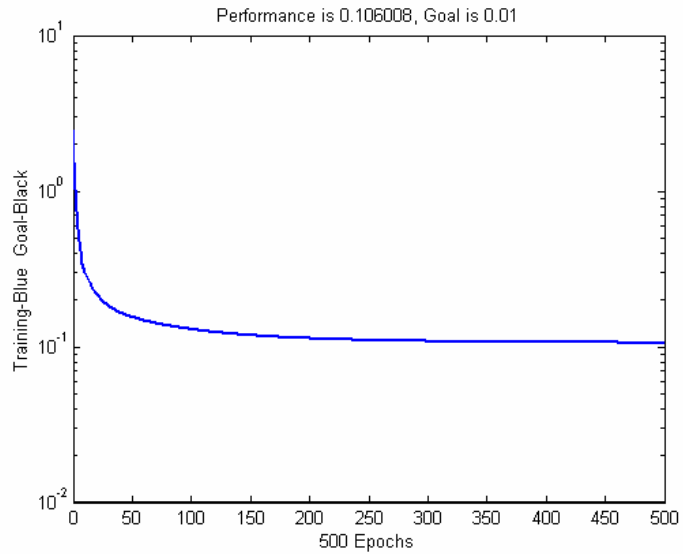


Fig.2.15

Fig.2.16 și Fig. 2.17 prezintă evoluția erorii, pentru o rețea multistrat cu o intrare, 15 neuroni ascunși cu o funcție de activare tangentă hiperbolică ('tansig'), cu o funcție de activare ieșire: liniară ('purelin'), antrenată cu metoda retropropagării erorii de tip gradient descendent, cu impuls și rată de învățare adaptivă ('traingdx'), pentru o dispersie a zgomotului $\sigma^2=0.01$, respectiv $\sigma^2=0.1$. S-au folosit 100 semnale de intrare și canalul de comunicație dat de relația (2.12).

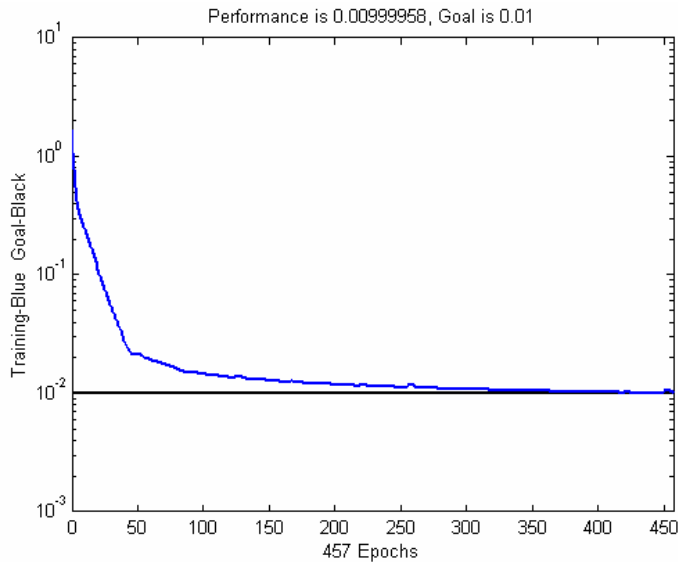


Fig.2.16

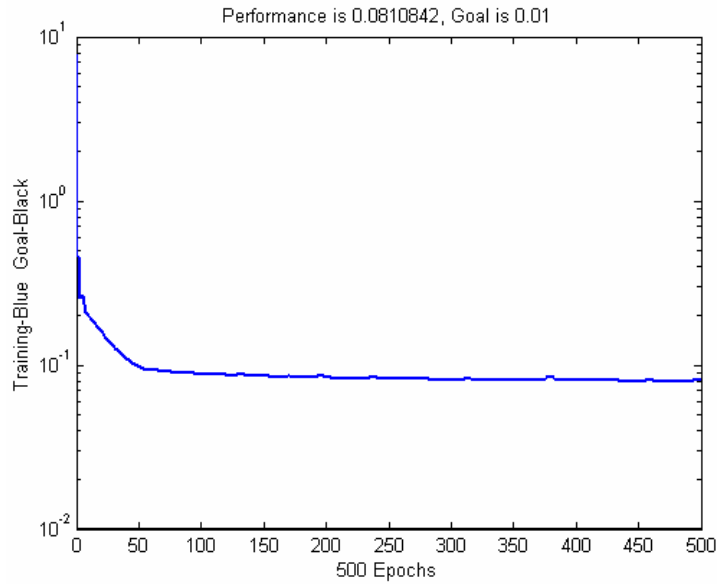


Fig.2.17

Fig.2.18 și Fig. 2.19 prezintă evoluția erorii, pentru o rețea multistrat cu o intrare, 15 neuroni ascunși cu o funcție de activare tangentă hiperbolică ('tansig'), un neuron de ieșire cu o funcție de activare liniară ('purelin'), antrenată cu metoda retropropagării erorii de tip gradient descendent și rată de învățare adaptivă ('traingda'), pentru o dispersie a zgomotului $\sigma^2=0.01$, respectiv $\sigma^2=0.1$. S-au folosit 100 semnale de intrare și canalul de comunicație dat de relația (2.12).

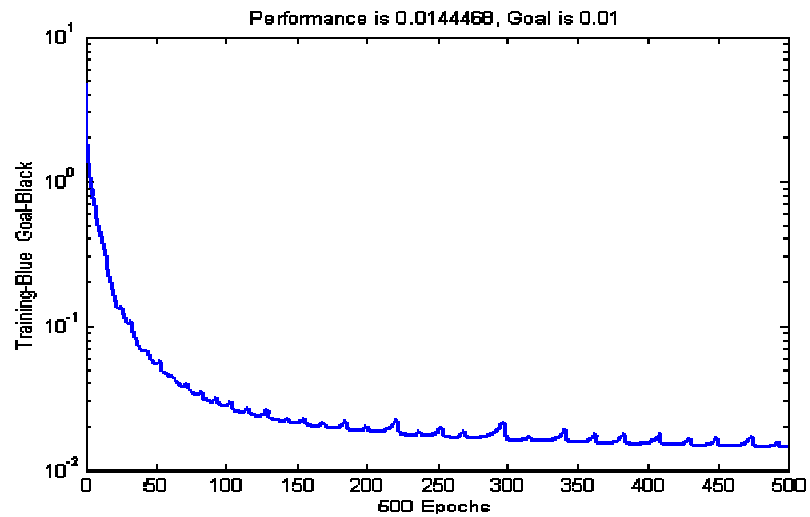


Fig.2.18

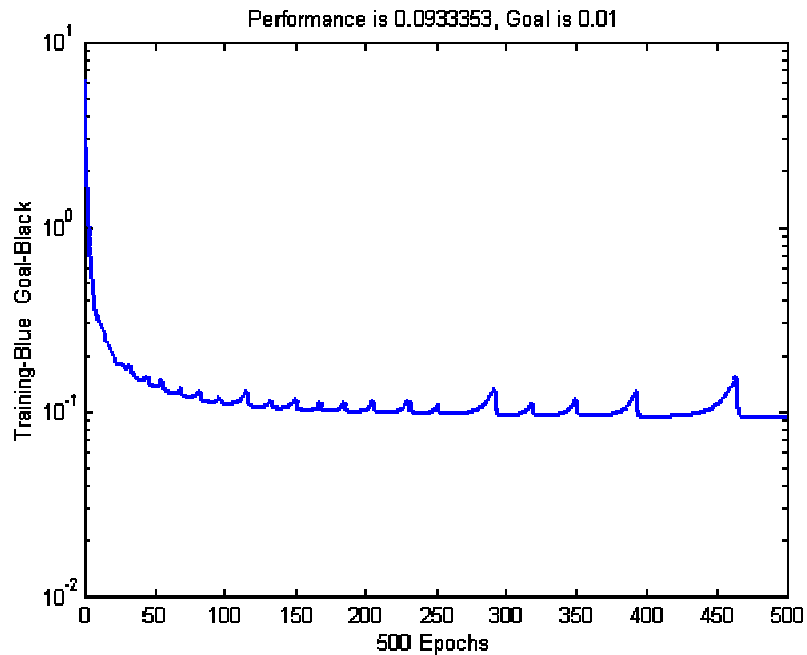


Fig.2.19

Fig.2.20 și Fig. 2.21 prezintă evoluția erorii, pentru o rețea multistrat cu o intrare, 15 neuroni ascunși cu o funcție de activare tangentă hiperbolică ('tansig'), un neuron de ieșire cu o funcție de activare liniară ('purelin'), antrenată cu metoda retropropagării erorii de tip gradient descendent cu impuls ('traingdm') pentru o dispersie a zgomotului $\sigma^2=0.01$, pentru 100 modele de intrare, respectiv 500 modele de intrare și canalul de comunicație dat de relația (2.10) .

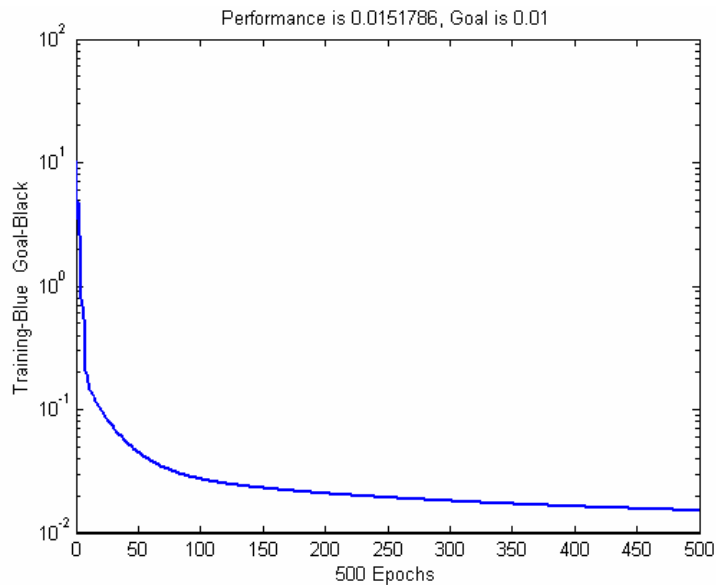


Fig.2.20

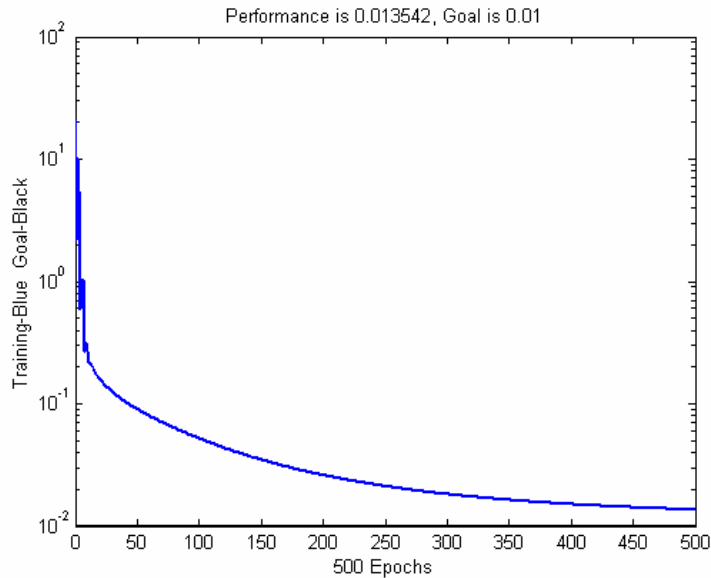


Fig.2.21

Observații

Canalele implicate în testele efectuate sunt puternic neliniare, implicând neliniarități de ordinul doi și trei. Limitele de decizie în cazul semnalelor 4-QAM testate sunt și ele neliniare, astfel încât posibilitățile egalizoarelor liniare tradiționale sunt evident depășite. Comparativ cu alte egalizoare neuronale raportate în literatură, în urma simulărilor efectuate, am constatat că performanța egalizorului multistrat implementat este similară, la o aceeași complexitate. Performanța crește cu numărul neuronilor din stratul ascuns al RNM, precum și cu numărul datelor de antrenament, ceea ce era de așteptat, conform teoriei. Performanțele obținute au fost superioare în cazul utilizării funcției de activare tangentă hiperbolică ('tansig') atât pentru stratul ascuns cât și pentru cel de ieșire, comparativ cu funcția de activare liniară ('purelin'). Pentru același canal, o aceeași dispersie a zgomotului, o aceeași structură a RN și aceleași date de intrare, performanța egalizorului este mai bună în cazul utilizării metodei de antrenament backpropagation gradient descent cu impuls ('traingdm') și în cazul metodei de antrenament backpropagation gradient descent cu impuls și rată de învățare adaptivă ('traingdx'). În cazul unui zgomot cu dispersie ($\sigma^2 = 0.01$) performanța dorită a fost atinsă pentru aceste două cazuri în 349 de epoci respectiv 457 de epoci. Pentru un număr mai mare de epoci de antrenament performanța egalizorului crește, dar de la un număr de aproximativ 600 de epoci nu se mai observă o creștere semnificativă. În cazul unui zgomot mai mare, cu dispersie ($\sigma^2 = 0.1$) performanța dorită nu a fost atinsă, posibilitățile sistemului de calcul, disponibil la momentul respectiv, fiind depășite.

Am observat o îmbunătățire a performanței, în cazul în care am folosit canalul 1 date de relația (2.10), deoarece acesta distorsionează mai puțin semnalul transmis.

Se poate realiza relativ simplu o extensie a egalizorului pentru semnale m-QAM, crescând numărul neuronilor ascunși ai rețelei și numărul neuronilor de ieșire la $2m$, m neuroni pentru partea reală, respectiv m neuroni pentru partea imaginară a semnalelor complexe. În acest caz limitele de decizie ce trebuie implementate sunt mai complexe, deci și capacitatea de calcul implicată trebuie să fie superioară.

2.10 Concluzii și perspective

Câteva dintre carențele rețelelor neuronale multistrat sunt timpul îndelungat de antrenare, datorat necesității de prezentare repetată a modelelor de antrenament, posibilitatea blocării în minimele locale ale funcției de eroare și structurile relativ mari rezultate în aplicațiile practice

[S.Haykin, 94.2]. Astfel încât în ultimul deceniu interesul cercetării internaționale în domeniu s-a focalizat asupra altor RN care evită deficiențele RNM, respectiv asupra rețelelor pe bază de funcții radiale (RBFR) și rețelelor neuronale recurente.

Ca și RNM, RBFR sunt aproximatoare universale, adică sunt capabile să aproximeze orice funcție neliniară într-un spațiu multidimensional complex cu o complexitate de procesare semnificativ redusă comparativ cu a RNM.

Vom prezenta în cele ce urmează alte câteva dintre caracteristicile RBFR comparativ cu ale RNM:

- Ca și RNM, rețeaua RBFR este o rețea cu procesarea informației "spre înainte". Spre deosebire de RNM care are poate avea mai multe straturi ascunse, RBFR are un singur strat ascuns.
- Neuronii stratului ascuns sunt neliniari și ai celui de ieșire sunt liniari, spre deosebire de cei ai RNM care în majoritatea aplicațiilor sunt exclusiv neliniari. Atunci când RNM sunt utilizate ca și clasificatoare ele utilizează neuroni neliniari atât în stratul ascuns cât și în cel de ieșire. Doar în cazul în care RNM rezolvă probleme de regresie neliniară, în stratul de ieșire, se preferă uzual neuroni liniari.
- În stratul ascuns RBFR utilizează funcții de activare de tip canal, al căror răspuns neliniar (uzual exponențial) este localizat într-o regiune restrânsă a spațiului modelelor de intrare. Ca și RNM, RBFR sunt aproximatoare universale. RNM realizează o aproximare globală a transformării neliniare intrare-ieșire, în timp ce RBFR, datorită utilizării funcțiilor radiale, implementează aproximări locale. Acesta este motivul pentru care RBFR învață mai rapid și sunt mai robuste, respectiv mai puțin sensibile la ordinea prezentării datelor de intrare, decât RNM.
- În plus, funcția de activare a neuronilor RBFR este o mărime reală și în cazul abordării unor semnale complexe, și nu o funcție complexă, ca în cazul RNM. Atunci când semnalul de intrare este complex, partea reală și cea imaginară sunt prelucrate separat, în aceeași manieră

Pentru antrenarea RBFR s-au propus mai mulți algoritmi de adaptare a ponderilor, constând de obicei dintr-un algoritm fără control pentru determinarea vectorilor centru ai stratului ascuns al rețelei și unul cu control pentru determinarea vectorilor pondere ale interconexiunilor stratului de ieșire. Proiectarea și antrenarea rețelelor RBFR depind esențial de modul de alegere a centrilor. Principalele strategii care s-au impus în practică sunt următoarele: alegerea aleatoare a unor centrii ficși din baza de date [S.Haykin, 94.2], alegerea unui subset din datele de intrare, folosind metoda valorilor pătrate ortogonale cele mai mici (orthogonal least squares) [S.Chen, 91], algoritmul standard competitiv [Hecht-Nielsen, 90], algoritmul sensibil la frecvența de câștigare a competiției [S.C.Ahalt, 90], algoritmul competitiv cu rivalului [L.Xu, 93].

Rețelele neuronale recurente (RNR) sunt cazul cel mai general de RN, având fiecare neuron conectat cu toți ceilalți neuroni ai rețelei. În acest caz rețeaua neuronală se numește complet conectată. Neuronii sunt neliniari și au o comportare dinamică complexă, astfel încât RNR sunt în mod special recomandate pentru aplicațiile în timp real. Ieșirea unui neuron la un moment dat depinde nu numai de intrările externe și de ieșirile altor neuroni, ca în cazul RNM și RBF, dar și de ieșirile sale anterioare. RNR cu o aceeași structură au comportări dinamice diferite în funcție de algoritmul de antrenament. În consecință o RNR este complet definită prin specificarea atât a arhitecturii cât și a algoritmului de antrenament. Algoritmul uzual pentru antrenarea unei RNR capabile să proceseze semnale complexe este învățarea recurentă în timp real (Real Time Recurrent Learning (CRTRL)) [G.Kechriotis, 99]. Acest algoritm se bazează pe o procedură de minimizare a erorii pătratice medii de tip gradient descendent, astfel încât se caracterizează printr-o viteză redusă de convergență și pot fi afectate de condiționare defectuoasă numerică. Complexitatea numerică a calculelor este de ordinul n^4 unde n este numărul neuronilor rețelei. Dimensiunea redusă a RNR în comparație cu a RNM și RBF pentru o aceeași performanță sunt principalul argument pentru utilizarea în aplicațiile de reconstrucție a semnalelor afectate de neliniarități rapid variabile în timp.

Există structuri de RN recurente obținute prin combinații ale unor filtre liniare transversale cu egalizoare cu decizie pe baza reacției inverse, ce încorporează RN [A.Guntsch, 98]. În [S.Bouchired, 99.1] se utilizează o RBFR sau o rețea competitivă, cu un filtru liniar transversal (FLT) pentru a îmbunătăți decizia unui egalizor cu decizie pe baza reacției inverse, pentru semnale 16-MAQ, aplicate unor dispozitive neliniare. Aceste arhitecturi sunt prezentate în Fig.2.22.a și b. Centrii rețelei RBF au fost calculați cu algoritmul K-mean. Pentru etichetarea neuronilor stratului de ieșire s-a utilizat algoritmul celei mai mici medii pătratice (least mean square LMS). În rețeaua competitivă pentru determinarea neuronului câștigător a fost utilizată regula lui Kohonen. Fiecare neuron al rețelei Kohonen a fost apoi asociat cu unul dintre simbolurile transmise printr-un tabel de referință. Mai multe

arhitecturi de RN recurente au fost implementate pentru egalizarea semnalelor transmise prin satelit în cadrul unui proiect european, NEWTEST ACTS European Project.

Cea mai importantă concluzie a acestui proiect a fost că structurile cu RN au depășit performanțele egalizatoarelor clasice în special în cazul unor neliniarități mari și atunci când au fost utilizate scheme de modulație complicate [S.Bouchired, 99.2]

În Fig.23 sunt reprezentate curbele de separare a semnalelor reconstruite, 16-MAQ modulate în amplitudine în cuadratură, în spațiul modelelor de ieșire a trei tipuri de rețele recurente [S.Bouchired, 99.1], obținute prin combinația dintr-un filtru liniar transversal și o rețea neuronală de tip multistrat cu două straturi ascunse (LF-NLN), prin combinația dintr-un filtru liniar transversal și o rețea competitivă (LTE-SOM) respectiv prin combinația dintr-un filtru liniar transversal și o rețea pe bază de funcții radiale LTE-RBF.

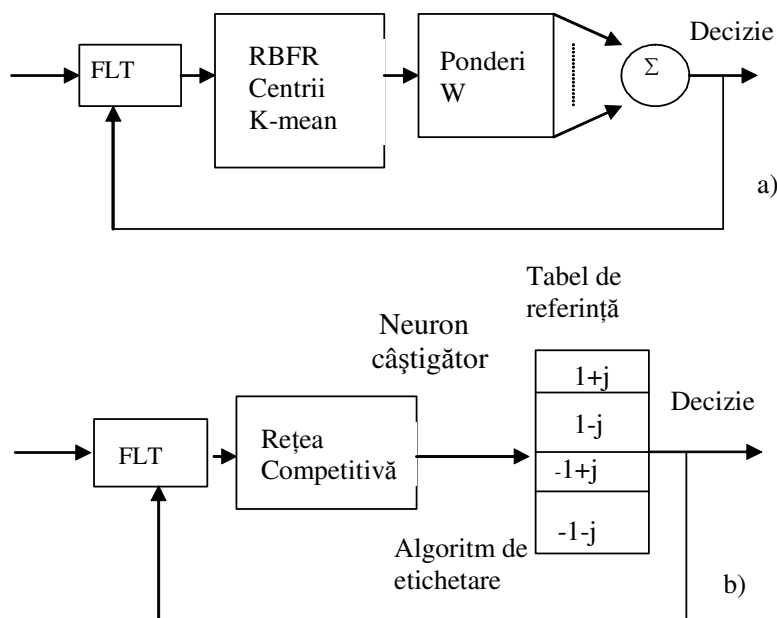


Fig.2.22. Structura unei rețele neuronale recurente obținute prin combinația unui filtru liniar transversal (FLT) cu o RBFR a) respectiv o rețea competitivă b)

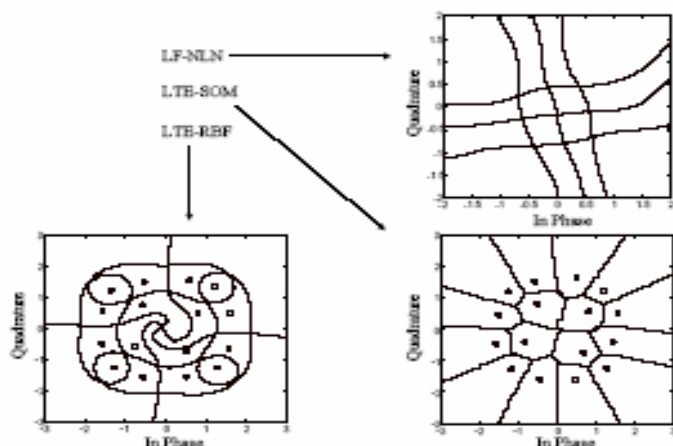


Fig 23 Curbele de separare obținute de trei rețele recurente obținute prin combinația dintr-un filtru liniar transversal și o rețea neuronală de tip multistrat cu două straturi ascunde(LF-NLN), o rețea competitivă (LTE-SOM) respectiv o rețea pe bază de funcții radiale LTE-RBF.

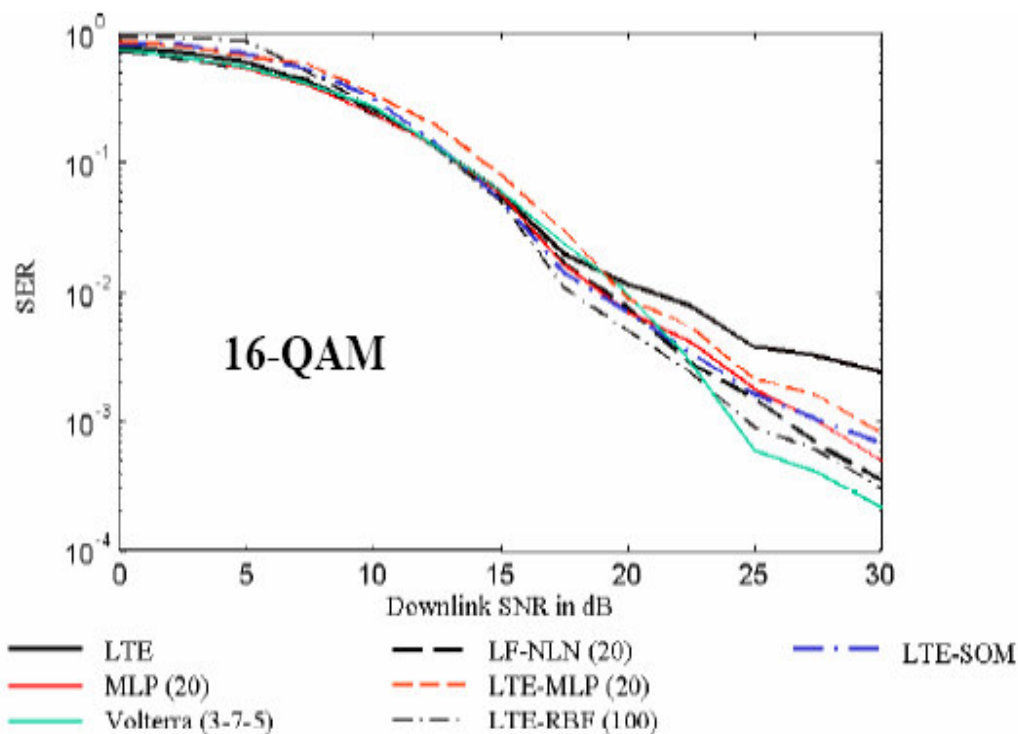


Fig. 24 Eroarea de bit în funcție de raportul semnal pe zgomot exprimat în dB pentru diferite egalizoare obținute cu: un filtru liniar transversal (LTE); rețea neuronală de tip multistrat (MLP) ; o rețea neuronală recurentă, combinație dintre un filtru liniar transversal o rețea neuronală de tip multistrat cu două straturi ascunde, cu o structură 1-20-1 (LF-NLN); o rețea neuronală recurentă combinație dintr-un filtru liniar transversal și o rețea neuronală de tip multistrat (LTE-MLP);o rețea neuronală recurentă, combinație dintre un filtru liniar traversa și o rețea competitivă (LTE-SOM) și o rețea neuronală recurentă, combinație dintre un filtru liniar transversal și o rețea pe bază de funcții radiale (LTE-RBF).

Fig. 24 reprezintă eroarea de bit în funcție de raportul semnal pe zgomot pentru egalizarea unui canal de transmisie prin satelit, conexiunea în jos, pentru semnale 16-QAM, pentru o viteză de deplasare de 150 km/h. Egalizorul recurent de tip LTE- RBF are cea mai bună performanță, respectiv cu un ordin de mărime mai bună decât egalizorul transversal.

Având în vedere avantajele oferite de rețelele neuronale pe bază de funcții de bază radiale (RBF), comparativ cu egalizoarele neuronale multistrat, respectiv simplitatea structurii, utilizarea unei funcții de activare reale, rapiditatea convergenței și robustețea, cercetarea noastră se va orienta în special asupra acestora. În plus arhitectura RBF similară cu structura optimă de tip Bayes garantează obținerea performanțelor optime din punctul de vedere al erorii de bit.

Urmează să testăm în Matlab mai mulți algoritmi de determinarea a centrilor rețelei RBF, respectiv algoritmul standard competitiv, algoritmul sensibil la frecvența de câștigare a competiției, algoritmul competitiv cu conștiință. Se vor face observații asupra avantajelor și dezavantajelor oferite de fiecare algoritm de învățare. Vom dezvolta și implementa noi algoritmi competitivi și vom dezvolta produse soft de aplicare a acestor algoritmi practic în clasificarea semnalelor complexe, în eliminarea zgomotului canalelor de comunicație.

Deoarece RNR depășesc performanțele egalizoarelor tradiționale și ale celorlalte egalizoare neuronale atunci când canalul de telecomunicații este variabil în timp și are distorsiuni neliniare, cercetarea noastră se va orienta asupra RNR. În mod special ne vom ocupa de rețelele neuronale recurente cu o structură de tip filtru liniar transversal- rețea neuronală pe bază de funcții radiale și vom extinde aplicarea algoritmilor competitivi proprii într-o structură recurentă. Avantajele ar fi o convergență rapidă și precizie superioară algoritmilor similari raportați în literatura de specialitate.

CAPITOLUL III

COMPENSAREA ECOULUI FOLOSIND TEHNICI DE FILTRARE NELINIARĂ

3.1 Problema compensării ecoului pe liniile telefonice

3.1.1 Generalități

Transmisia full-duplex în cadrul unei bucle de abonat ISDN este implementată pe o linie telefonică de două fire prin intermediul unui convertor care face trecerea de la două fire la patru fire, numit hibrid. Funcția hibridului într-o legătură este aceea de a izola semnalul transmis local de cel recepționat provenit de la transmițătorul îndepărtat. Izolarea totală se face printr-o impedanță de izolare plasată în interiorul hibridului care ar trebui să realizeze o adaptare perfectă între cele două fire. Ținând însă cont de faptul că impedanța liniei este o funcție de frecvență, timp și temperatură, acest lucru este dificil de realizat [K.Yamazaki, 91] [K. Feher, 93].

În practică, un hibrid poate realiza o atenuare de 6-10 dB între transmițătorul local și calea de recepție. Din păcate, atenuarea semnalului recepționat de la transmițătorul îndepărtat poate atinge 40-50 dB în cadrul buclelor lungi. Acest lucru determină un raport nedorit semnal/ecou de 30-40 dB. Valoarea acestui raport trebuie să fie de cel puțin 20 de decibeli pentru a asigura o detecție a semnalului de date din semnalul recepționat. În consecință, atenuarea impusă ecoului trebuie să fie cu peste 50-60 dB mai mare decât cea realizată de hibrid.

Rolul compensatorului de ecou este de a realiza o replică a semnalului de ecou bazată pe secvența de date transmisă. Compensatorul de ecou își adaptează astfel coeficienții încât să realizeze în mod eficient atenuarea de 50-60 dB a ecoului.

Semnalul cu ecou constă dintr-o componentă liniară și o componentă neliniară. Componenta liniară reprezintă componenta importantă ca amplitudine și durată. Componenta neliniară este de durată mult mai scurtă. Neliniaritățile sunt introduse, în principal, de către componentele circuitului din calea legăturii.

Se pot identifica 3 surse majore care determină apariția neliniarităților [K. Feher, 93]:

- asimetria impulsurilor transmise;
- saturarea transformatoarelor;
- convertoarele A/D și D/A ce operează în cadrul sistemului.

Nivelul neliniarităților poate fi redus printr-o proiectare atentă a circuitelor generatoare de impulsuri prin utilizarea unor transformatoare adecvate și impunând o toleranță restrânsă componentelor din circuit.

În practică, de cele mai multe ori, se realizează un compromis între reducerea neliniarităților, creșterea complexității calculelor și prețul circuitului. Acest lucru trebuie să se facă ținând cont de faptul că un nivel al neliniarității de -40 dB raportat la componenta liniară este considerat acceptabil.

3.1.2 Suprimarea ecoului în rețelele de transmisii vocale

Sursa ecoului poate fi înțeleasă considerând o conexiune simplă între doi abonați, Ab1 și Ab2, ca în figura 3.1. În trunchiul de patru fire, cele două direcții de transmisie sunt separate pe echipamente diferite. Ecoul este introdus în bucla de reacție din jurul porțiunii de patru fire a conexiunii. Un circuit telefonic tipic este ilustrat în figura 3.1 [Texas Instruments, 86] [K.Yamazaki, 91] [K. Feher, 93]:

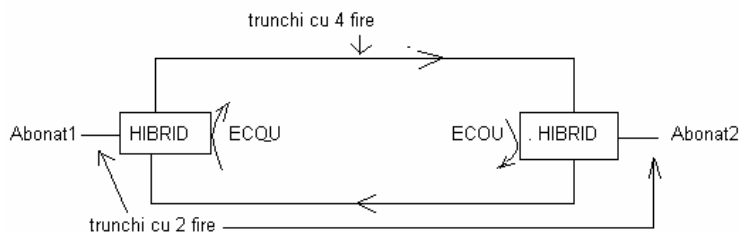


Figura 3.1 Conexiune simplă între 2 abonați

Această conexiune conține un segment de 2 fire spre capete, 4 fire la centru și un hibrid pentru fiecare capăt ce face conversia de la 2 fire la 4 fire și invers. Fiecare segment constă dintr-o buclă de abonat și porțiuni din rețeaua locală. Pe segmentul de 2 fire, ambele direcții de transmisie sunt purtate de aceeași pereche de fire astfel încât semnalele de la ambii vorbitori sunt suprapuse pe același segment. În secțiunea de 4 fire, cele două sensuri de transmisie sunt separate, convorbirea de la abonatul 1 urmărind partea superioară, iar convorbirea de la abonatul 2 urmărind partea inferioară. Separarea celor 2 semnale este de dorit acolo unde este necesară introducerea de terminale, amplificatoare sau alte echipamente de telecomunicații.

Rolul hibridului, în partea dreaptă, este de a direcționa energia semnalului care sosește de la abonatul 1 spre segmentul de 2 fire al abonatului 2 fără a-i permite să se întoarcă la abonatul 1 prin calea inferioară în segmentul de 4 fire. Din cauză că impedanța nu poate fi adaptată perfect o parte din energia semnalului de la abonatul 1 se întoarce la el, unde se aude o versiune întârziată a vorbirii sale, acest lucru constituind sursa ecoului vorbit. Efectul subiectiv al ecoului vorbitorului depinde de întârzierea cu care acesta ajunge la vorbitor. Pentru întârzieri scurte, acesta reprezintă o sursă nesemnificativă de perturbație dacă atenuarea este rezonabilă. Dacă avem conexiuni prin satelit, întârzierea pentru fiecare cale este de 270 ms, ceea ce ne dă o întârziere cu care este perceput ecoul de 540 ms, ceea ce este foarte deranjant pentru vorbitor și face dificilă conversația [Texas Instruments, 86].

Compensatorul de ecou are schema de principiu din figura 3.2 și prezintă următoarele părți importante: o parte din conexiunea pe patru fire, un hibrid și segmentul de abonat pe două fire. Se pot remarca cele două direcții de transmisie: prima între porturile A și C, iar cealaltă direcție de transmisie între porturile D și B.

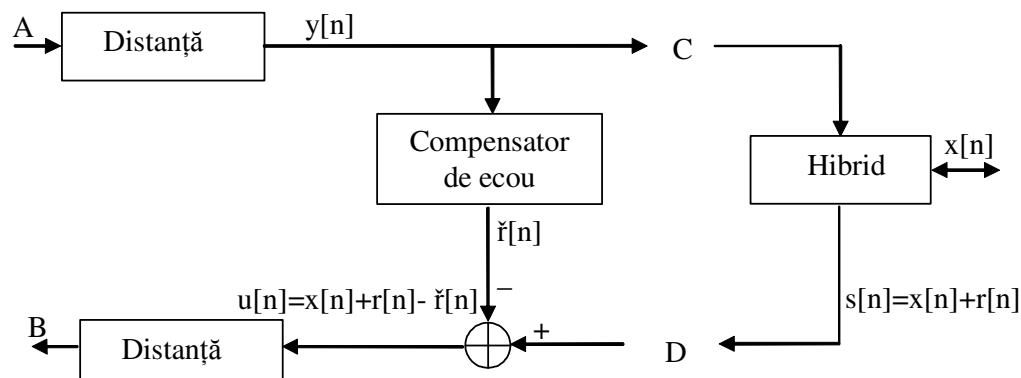


Figura 3.2 Configurația compensatorului de ecou

Toate semnalele din schemă sunt semnale numerice care apar în mod natural în transmisiile digitale.

Semnalul provenit de la vorbitorul îndepărtat este notat cu $y[n]$, ecoul nedorit cu $r[n]$, iar semnalul venit de la vorbitorul apropiat cu $x[n]$. Semnalul de la vorbitorul apropiat este suprapus cu ecoul în hibrid și este scos la portul D. Semnalul vorbitorului îndepărtat este luat ca și semnal de referință în suprimarea ecoului și este folosit pentru a genera replica ecoului notată cu $\tilde{r}[n]$. Această replică a ecoului este adunată cu semnalul de la portul D. Semnalul rezultat se notează cu $u[n]$ și are expresia de mai jos, unde $r[n] - \tilde{r}[n]$ constituie semnalul de eroare [Texas Instruments, 86]:

$$u[n] = x[n] + r[n] - \tilde{r}[n] \quad (3.1)$$

Eroarea reziduală a ecoului, $e[n] = r[n] - \tilde{r}[n]$, este foarte mică după compensare. Schema prezentată în figura 3.2 a fost realizată pentru un singur sens de transmisie.

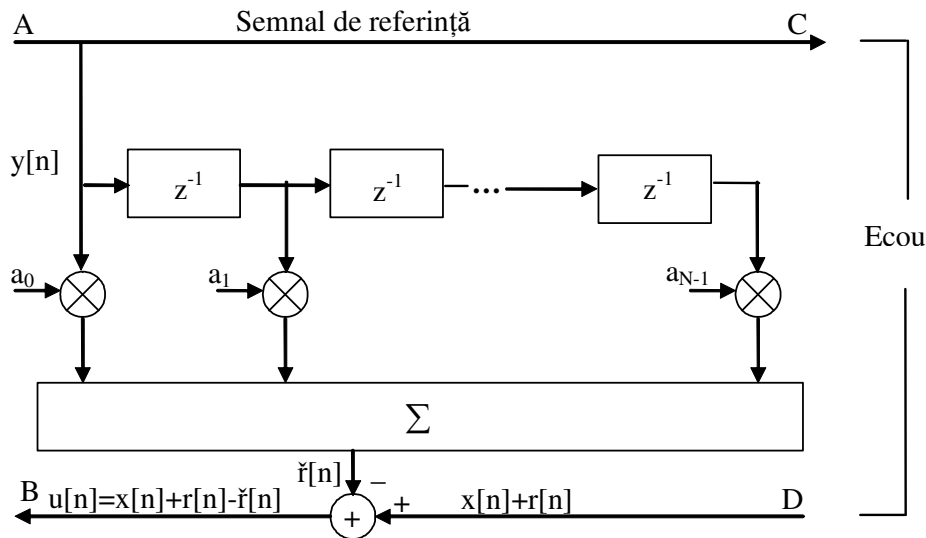


Figura 3.3. Estimarea ecoului utilizând un filtru transversal

Supresorul de ecou generează replica ecoului luând semnalul de referință de la un filtru transversal, așa după cum se arată în figura 3.3. Dacă funcția de transfer a filtrului transversal este identică cu cea a căii ecoului, replica ecoului va fi identică cu ecoul, obținând astfel compensarea totală. Întrucât funcția de transfer a căii de unde apare ecoul nu este cunoscută dinainte, modelarea ei se face prin metode adaptive, adică coeficienții filtrului transversal se vor modifica urmărind obținerea la ieșirea acestuia a unei replici a ecoului.

Numărul de pași de întârziere ai filtrului transversal din figura 3.3. este determinat de durata răspunsului la impuls a căii ecoului de la portul C la portul D. Intervalul de timp, pentru care acesta este diferit de zero, este tipic cuprins între 2 și 4 ms. Această durată implică, în cazul unei frecvențe de eșantionare de 8 kHz, alegerea unui filtru transversal cu 16- 32 coeficienți.

Oricum, pentru că legătura de 4 fire este între supresorul de ecou și hibrid, acest răspuns sigur nu începe de la zero, fiind întârziat. În consecință, ordinul filtrului N trebuie să țină cont și de această întârziere. Alegând $N=128$, întârzierile de peste 16ms pot fi adaptate.

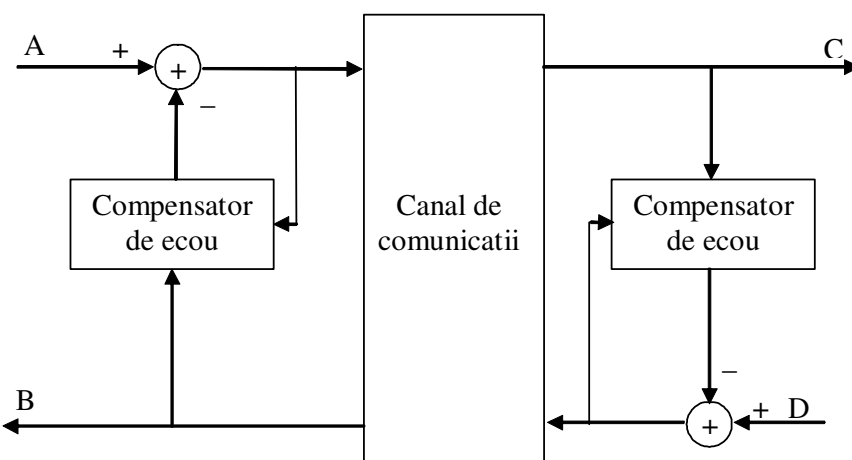


Figura 3.4 Configurația compensatorului de ecou pentru ambele sensuri de transmisie

În practică, este necesară compensarea ecourilor în ambele direcții ale convorbirii. În acest scop, sunt folosite două supresoare adaptive, dintre care unul compensează ecoul de la fiecare terminal al conexiunii. Vorbitorul apropiat pentru unul dintre supresoare, este vorbitorul îndepărtat pentru celălalt. În orice caz, vorbitorul apropiat este „cel mai apropiat vorbitor” și vorbitorul îndepărtat este acela ce generează ecoul care trebuie compensat. Este de dorit a poziționa aceste două „jumătăți” ale supresorului arătată în fig 3.4. unde întârzierea în legătura de patru fire a conexiunii este în mijloc. Motivul este că numărul coeficienților necesar în filtrul compensator de eco este direct legat de întârzierea porțiunii de circuit între compensatorul de eco și hibridul care generează ecoul.

3.2 Algoritm de suprimare a ecoului

3.2.1 Implementarea

Întrucât întârzierea cu care este perceput ecoul, energia acestuia, precum și comportarea hibridului în generarea ecoului constituie necunoscute, pentru suprimarea ecoului se folosesc metode de filtrare adaptive.

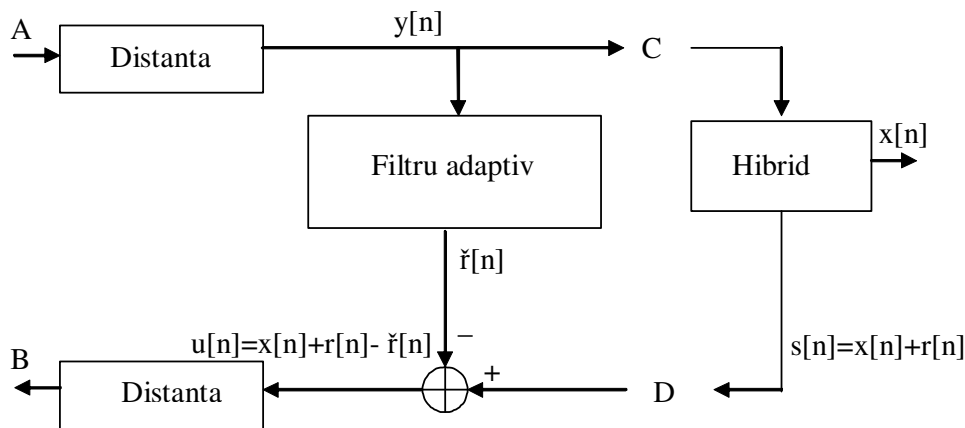


Figura 3.5 Supresor de eco implementat cu filtru adaptiv

Schema unui astfel de supresor este prezentată în figura 3.5 unde pentru generarea replicii ecoului s-a folosit un filtru Volterra adaptiv de ordinul al doilea. Semnalul $\check{r}[n]$ de la ieșirea acestui filtru este dat de relația:

$$\check{r}[n] = \sum_{n_1=0}^{N-1} h_1[n_1]y[n-n_1] + \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{N-1} h_2[n_1, n_2]y[n-n_1]y[n-n_2] \quad (3.2)$$

Deoarece, în general, răspunsul la impuls al ecoului este necunoscut și poate varia lent în timp, este necesară o buclă de program închisă a algoritmului pentru a minimiza eroarea medie pătratică dintre eco și replica sa. Se poate observa că semnalul $u[n]$ este format din semnalul de eroare, $e[n] = r[n] - \check{r}[n]$, și semnalul de la vorbitorul apropiat, $x[n]$, care este necorelat cu semnalul de la vorbitorul îndepărtat, $y[n]$. Acest lucru ne duce la ecuația:

$$E(u^2[n]) = E(x^2[n]) + E(e^2[n]) \quad (3.3)$$

După cum se observă, ecoul este minim când termenul din partea stângă din relația (3.3) este minim. Dacă semnalul de la vorbitorul apropiat lipsește ($x[n] = 0$) minimul se poate obține ajustând coeficienții de-a lungul direcției gradientului negativ al lui $E(u(n))$, după ecuația:

$h_k(n+1) = h_k[n] - \beta \frac{d(E(e^2[n]))}{dh_k(n)}$	(3.4)
--	-------

unde am notat cu β mărimea pasului de adaptare. Substituind termenii în relația de mai sus avem:

$h_k(n+1) = h_k(n) + 2\beta E(e[n]y[n-k])$	(3.5)
--	-------

În practică, operatorul E nu poate fi calculat fără cunoștințe apriori ale distribuției de probabilitate a semnalului de referință, de aceea, folosim un estimat al gradientului ce e bazat pe media temporală a corelației erorii. Înlocuind operatorul E cu o medie de timp, avem :

$h_k(n+1) = h_k[n] + \frac{2\beta}{M} \sum_{m=0}^{M-1} e[n-m]y[n-m-k]$	(3.6)
--	-------

Dacă avem $M = 1$, obținem algoritmul LMS.

Proprietățile de convergență pe scară largă a algoritmului sunt determinate de pasul de adaptare, β , și de puterea semnalului îndepărtat, $y[n]$. În general, făcând pasul de adaptare mare crește viteza de convergență, și făcând pasul de adaptare mic crește precizia reducându-se asimptotic eroarea, dar cu timp de convergență mare. A fost arătat că, convergența în timp constant este invers proporțională cu puterea lui $y[n]$ și, de aceea, algoritmul converge foarte lent la semnalele de putere mică. Pentru a remedia această situație câștigul în buclă este, uzual, normalizat cu un estimat al puterii.

$2\beta \rightarrow 2\beta[n] = \frac{\beta_1}{P_y[n]}$	(3.7)
---	-------

unde β_1 este o valoare de compromis a pasului de adaptare constant β și $P_y[n]$ este un estimat al puterii medii a lui $y[n]$, la momentul n :

$P_y[n] = (L_y[n])^2$	(3.8)
-----------------------	-------

unde $L_y[n]$ este dat de:

$L_y[n+1] = (1-\rho)L_y[n] + \rho y[n] $	(3.9)
--	-------

Estimatul $\rho_y[n]$ este folosit întrucât calcularea puterii medii exacte necesită un volum mare de calcul.

3.2.2 Detecția convorbirii apropiate

Când ambele semnale, și cel apropiat și cel depărtat, sunt în linie, această situație este denumită "dublă convorbire". În acel moment, semnalul de eroare, $u[n]$, conține o componentă a vorbitorului apropiat, $x[n]$, adunată cu eroarea reziduală și în acest caz este necesar să se oprească compensarea ecoului pe durata dublei convorbiri [Texas Instruments, 86].

Unul dintre algoritmi de detectare a convorbirii apropiate constă în a declara convorbirea abonatului apropiat din momentul în care:

$ s[n] = x[n] + r[n] \geq \frac{1}{2} \max\{ y[n] , y[n-1] , \dots, y[n-N] \}$	(3.10)
---	--------

unde N este numărul eșantioanelor din memoria filtrului. Este necesar să se compare $s[n]$ cu eșantioanele recente ale semnalului îndepărtat, $y[n]$, din cauza necunoașterii întârzierii din calea ecoului. De fapt, algoritmul face o comparare a puterilor instantanee într-o fereastră de timp de durată egală cu întârzierea maximă a ecoului în hibrid.

3.2.3 Suprimarea ecoului rezidual

Neliniaritățile în calea cu ecoul a circuitului telefonic limitează pragul superior în suprimarea ecoului la 30 – 35 Db. Astfel, utilizarea algoritmului de anulare a ecoului rezidual reprezintă o soluție în acest caz. Pe durata dublei convorbiri, după cum s-a arătat și mai sus, suprimarea ecoului rezidual trebuie stopată. O variantă de control a algoritmului este de a detecta când puterea semnalului întors este sub un prag calculat în funcție de puterea semnalului de referință recepționat. Astfel, dacă semnalul returnat constă numai din ecoul rezidual și dacă anularea s-a făcut corect, atunci nivelul acestuia se va situa sub pragul impus și semnalul de întoarcere va fi pus pe zero.

Dacă semnalul $u[n]$ este format doar din ecoul rezidual și anularea a convers, nivelul ecoului rezidual va fi sub pragul admis și semnalul transmis reîntors va fi aproape 0.

Puterea estimată a semnalului întors se calculează cu relația (3.11), [Texas Instruments, 86]:

$L_u[n+1] = (1 - \rho)L_u[n] + \rho u[n] $	(3.11)
--	--------

Compensarea e activată pentru semnalul transmis $u[n]$ ($u[n] = 0$) ori de câte ori avem îndeplinită condiția:

$\frac{L_u[n]}{L_y[n]} < 2^{-4}$	(3.12)
----------------------------------	--------

Aceasta corespunde unui prag de compensare de +24 Db.

3.3 Algoritm LMS în filtrarea neliniară adaptivă

3.3.1 Considerații privind complexitatea filtrului Volterra

Numărul calculelor efectuate în cazul implementării unui sistem Volterra crește rapid cu creșterea ordinului neliniarității și lungimea memoriei. Această complexitate afectează timpul ocupat, necesar pentru estimarea nucleului, durata simulărilor în cazul utilizării modelului, precum și viteza și complexitatea hardware pentru implementarea și compensarea neliniarităților.

Pentru un model Volterra dat, numărul coeficienților și numărul total al înmulțirilor necesare dau o măsură a complexității acestuia. Figura 3.6 prezintă implementarea unui sistem Volterra tipic, indicând creșterea numărului de coeficienți și a operațiilor de înmulțire odată cu creșterea ordinului neliniarității și a memoriei sistemului.

Se remarcă faptul că în scopul calculării complexității, se presupune un sistem Volterra de ordinul N ce are toate neliniaritățile până la cea de ordinul N , precum și nuclee Volterra simetrice. De asemenea se presupune că lungimea memoriei este aceeași pentru toate ordinele. Deși aceasta simplifică măsurarea complexității în scopul ilustrării, lungimi egale ale memoriei pot să nu fie întotdeauna presupuneri justificate.

Se poate arăta că:

-numărul coeficienților nucleului de ordinul n este dat de relația:

$nr. \text{ coef.} = \frac{(M-1+n)!}{n!(M-1)!}$	(3.13)
---	--------

În cazul modelului Volterra de ordinul N , ținând cont de toate nucleele de ordin mai mic sau egal cu N , numărul coeficienților este dat de relația următoare:

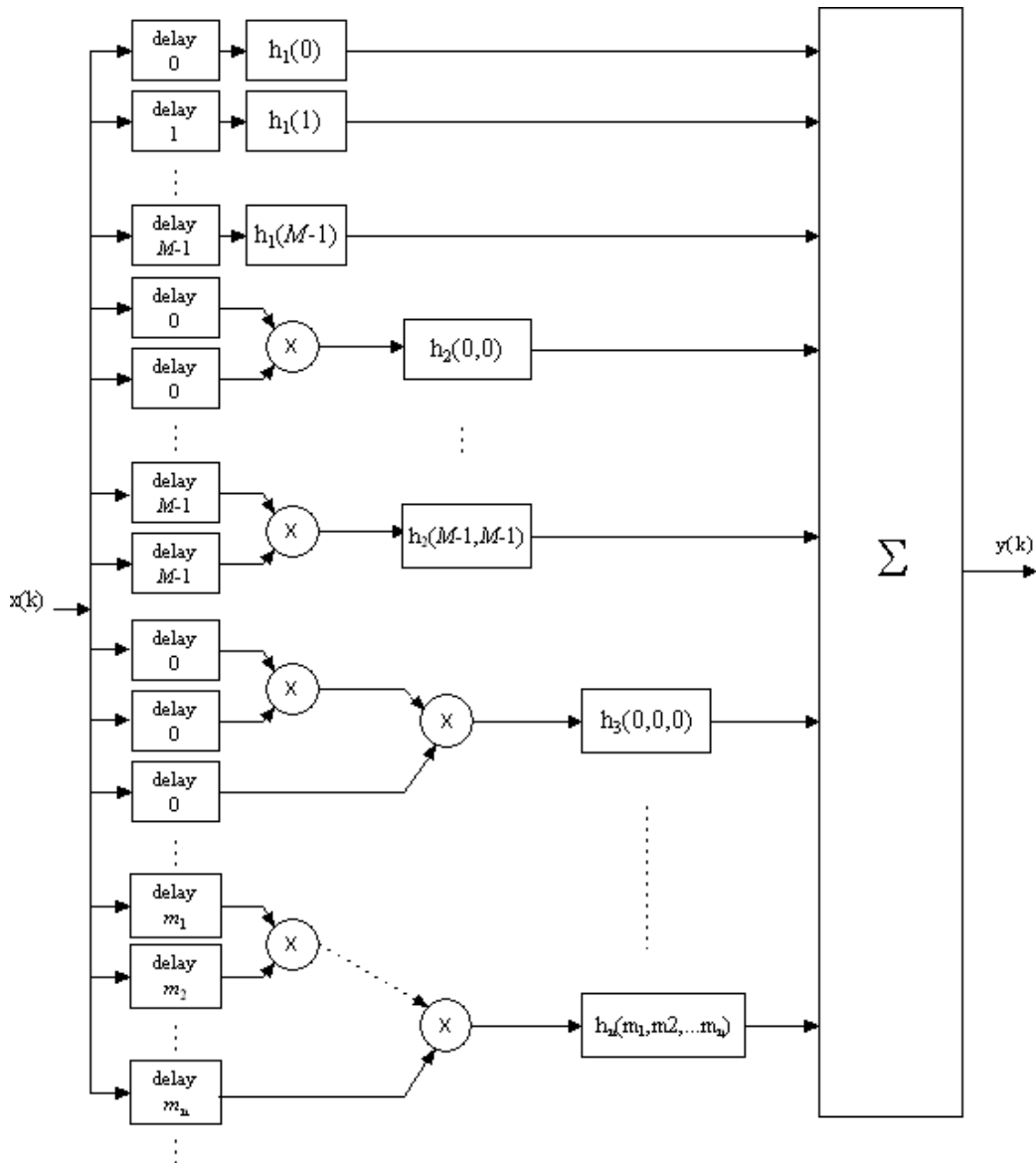


Figura 3.6 Implementarea modelului Volterra

$nr. \text{ coef.} = \sum_{n=1}^N \frac{(M-1+n)!}{n!(M-1)!}$	(3.14)
--	--------

Tabelul 3.1 și figura 3.7 arată cum variază numărul coeficienților Volterra cu ordinul neliniarității și cu memoria filtrului.

Ordinul neliniarității modelului, N	Fără memorie ($M=1$)	Memoria= 1 eșantion ($M=2$)	Memoria= 4 eșantioane ($M=5$)	Memoria= 9 eșantioane ($M=10$)
1	1	2	5	10
2	2	5	20	65
3	3	9	55	285
4	4	14	125	1000
5	5	20	251	3002

Tabelul 3.1 Numărul coeficienților Volterra ca funcție de modelul de ordin N și intervalul de memorie M

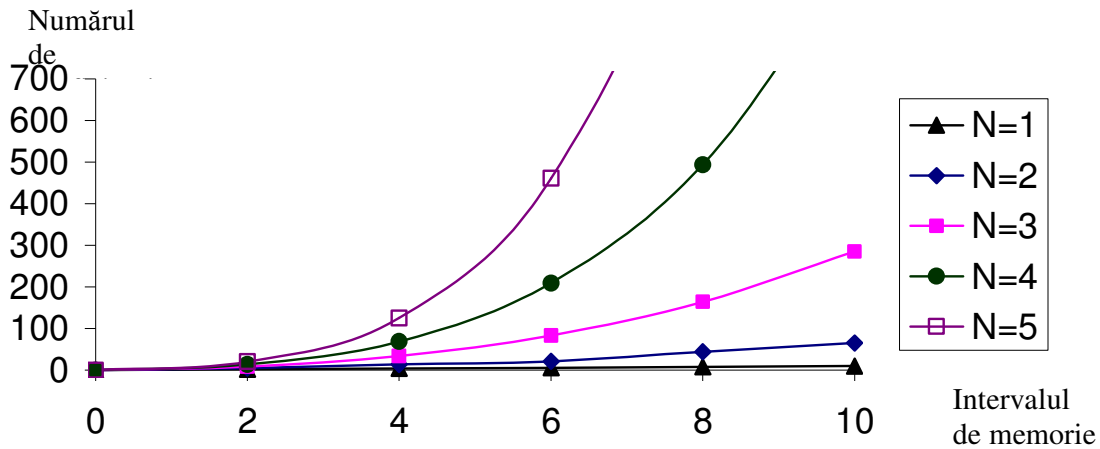


Figura 3.7 Numărul coeficienților Volterra ca funcție de ordinul N al modelului și intervalul de memorie M

Atunci când se evaluează numărul total de multiplicatoare se observă că, fiecare coeficient Volterra necesită un multiplicator, iar restul sunt necesare pentru a obține produsele corespunzătoare ale versiunilor întârziate ale intrării.

Numărul de înmulțiri pentru implementarea unui nucleu de ordin n este :

$$nr. \text{ inmultiri} = \frac{(M-1+n)!}{(n-1)!(M-1)!} \quad (3.15)$$

Numărul de înmulțiri pentru implementarea modelului Volterra de ordinul N este:

$$nr. \text{ inmultiri} = \sum_{n=1}^N \frac{(M-1+n)!}{(n-1)!(M-1)!} \quad (3.16)$$

Tabelul 3.2 și figura 3.8 arată cum variază numărul total de înmulțiri cu ordinul neliniarității modelului Volterra și cu memoria acestuia, în cazul nucleelor simetrice.

Ordinul neliniarității modelului, N	Fără memorie ($M=1$)	Memoria= 1 eșantion ($M=2$)	Memoria= 4 eșantioane ($M=5$)	Memoria= 9 eșantioane ($M=10$)
1	1	2	5	10
2	3	8	35	120
3	6	20	140	780
4	10	40	420	3640
5	15	70	1050	13650

Tabelul 3.2 Înmulțirile necesare ca funcție de memoria M

Scopul acestui subcapitol a fost de a da relații de bază între ordinul și intervalul de memorie al modelului Volterra și complexitatea calculului. O analiză detaliată a relațiilor între complexitatea de calcul a nucleului, ordinul și memoria nucleului se găsește în [J.Tsimbinos,95].

Complexitatea (Numărul de multiplicatoare)

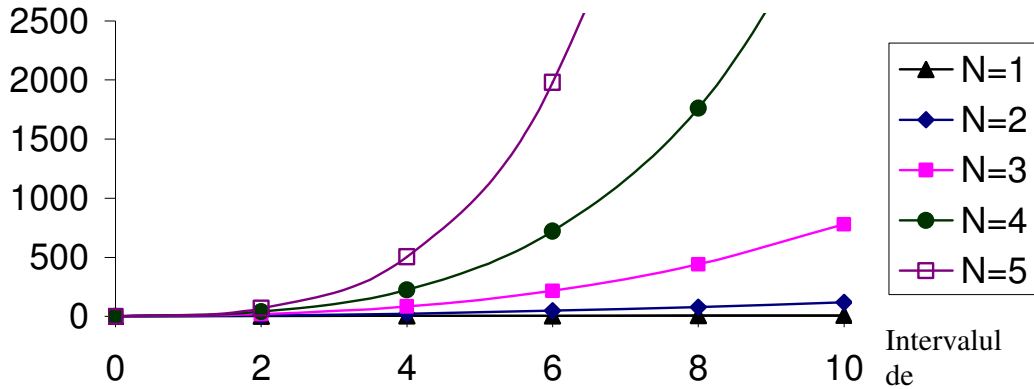


Figura 3.8 Complexitatea de calcul ca funcție de ordinul N al modelului Volterra și intervalul de memorie M

3.3.2 Implementarea algoritmului LMS în cazul filtrelor Volterra

În cele ce urmează se va considera un filtru Volterra ale cărei nuclee sunt simetrice. Sunt două părți ale acestei reprezentări: (1) estimările nucleelor Volterra și (2) produsele între valorile semnalului de intrare întârziat. Dacă exprimăm nucleele Volterra și produsele între valorile semnalului de intrare în formă vectorială, atunci putem scrie ieșirea filtrului adaptiv Volterra cu ajutorul notației vectoriale.

Fiecare nucleu Volterra (estimat la momentul k) se poate scrie în formă vectorială astfel:

- nucleu Volterra de ordin n și memorie M , la momentul k :

$$\underline{h}_k^{(n)T} = [h_k^{(n)}[0,0,\dots,0] \quad h_k^{(n)}[0,0,\dots,1] \quad \dots \quad h_k^{(n)}[M-1,M-1,\dots,M-1]] \quad (3.17)$$

- vectorul Volterra ce conține nucleele de diferite ordine:

$$\underline{h}_k^T = \left[h_k^{(0)} \mid \underline{h}_k^{(1)T} \mid \underline{h}_k^{(2)T} \mid \dots \mid \underline{h}_k^{(n)T} \mid \dots \mid \underline{h}_k^{(N)T} \right] \quad (3.18)$$

- vectorul de intrare corespunzător neliniarității de ordin n :

$$\underline{x}_k^{(n)T} = [x_k^n \quad x_k^{n-1}x_{k-1} \quad \dots \quad x_k^{n-1}x_{k-M+1} \quad \dots \quad x_{k-M+1}^n] \quad (3.19)$$

Un vector al produselor semnalului de intrare poate fi scris luând toți vectorii corespunzători diferitelor ordine și grupându-i ca în relația (3.20):

$$\underline{x}_k^T = \left[1 \mid \underline{x}_k^{(1)T} \mid \underline{x}_k^{(2)T} \mid \dots \mid \underline{x}_k^{(n)T} \mid \dots \mid \underline{x}_k^{(N)T} \right] \quad (3.20)$$

În consecință, ieșirea filtrului Volterra adaptiv la momentul k este:

$$\hat{y}_k = \underline{h}_k^T \cdot \underline{x}_k \quad (3.21)$$

La momentul k , ieșirea dorită este y_k , iar ieșirea filtrului Volterra este \hat{y}_k . Pentru algoritmul LMS, trebuie să minimizăm (3.22):

$$E\{e_k^2\} = E\{[y_k - \underline{h}_k^T \underline{x}_k]^2\} \quad (3.22)$$

Vectorul \underline{h}^* care minimizează relația (3.22) poate fi exprimat ca soluție a ecuațiilor normale date de (3.23):

$$\underline{h}^* = R_2^{-1} \cdot \underline{g} \quad (3.23)$$

unde: $R = E[\underline{x}_k \underline{x}_k^T]$ reprezintă matricea autocorelației semnalului de intrare și conține momentele semnalului de intrare până la ordinul $2N$, pentru un model de ordin N , iar $\underline{g} = E[\underline{x}_k y_k]$ este vectorul intercorelației dintre intrare și ieșirea dorită.

Ecuația de actualizare a coeficienților filtrului Volterra este din punct de vedere formal aceeași ca și în cazul filtrului liniar:

$$\underline{h}_{k+1} = \underline{h}_k + \mu e_k \underline{x}_k \quad (3.24)$$

unde: \underline{h}_k este vectorul coeficienților nucleelor Volterra la momentul k , $e_k = y_k - \hat{y}_k = y_k - \underline{h}_k^T \cdot \underline{x}_k$ iar μ este o constantă de valoare mică care, ca și în cazul filtrului liniar, determină viteza de convergență și de asemenea afectează eroarea finală a ieșirii filtrului. Trebuie remarcată asemănarea ca formă, în această notație, între cazul filtrului LMS liniar și cazul filtrului neliniar adaptiv Volterra.

Pentru a garanta convergența algoritmului LMS, este necesar să punem limite constantei μ . Dacă pasul de avans este mic, algoritmul LMS se adaptează încet, iar dacă algoritmul i se permite să ruleze un timp îndelungat, eroarea finală va fi redusă. Totuși, dacă se folosește un pas al avansului mai mare, algoritmul se adaptează mai rapid, dar eroarea finală va fi mai mare. Dacă pasul de avans este prea mare, atunci algoritmul poate deveni instabil și divergent. De aceea, fixarea limitelor superioare pentru convergență este de mare interes. Limita maximă a pasului de avans este legată de valorile proprii maxime ale matricei de autocorelație a vectorilor de intrare.

Se vor prezenta în cele ce urmează câteva din aspectele cele mai importante în problematica alegerii mărimii pasului μ . În acest scop vectorii \underline{h}_k și \underline{x}_k vor fi înlocuiți de matricele H respectiv X iar scalarul μ va fi înlocuit de către matricea M . Trebuie făcută observația că vectorii ce alcătuiesc matricile H și X au dimensiuni diferite, locurile rămase libere vor fi completate cu zerouri. Matricea H va conține ca linii vectorii nucleee $\underline{h}_k^{(n)}$:

$$H_k^T = [\underline{h}_k^{(0)} \quad \underline{h}_k^{(1)} \quad \underline{h}_k^{(2)} \quad \dots \quad \underline{h}_k^{(n)} \quad \dots \quad \underline{h}_k^{(N)}] \quad (3.25)$$

În mod similar matricea X are liniile formate din vectori ce conțin produse de același ordin ale semnalului de intrare.

$$X_k^T = [1 \quad \underline{x}_k^{(1)} \quad \underline{x}_k^{(2)} \quad \dots \quad \underline{x}_k^{(n)} \quad \dots \quad \underline{x}_k^{(N)}] \quad (3.26)$$

Matricea pasului de adaptare M este de forma:

$$M = \begin{bmatrix} \mu_0 & 0 & 0 & \dots & 0 \\ 0 & \mu_1 & 0 & \dots & 0 \\ 0 & 0 & \mu_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \mu_N \end{bmatrix} \quad (3.27)$$

În aceste condiții ecuația de actualizare a coeficienților filtrului devine de forma:

$$H_{k+1} = H_k + e_k M X_k \quad (3.28)$$

adică:

$$\begin{bmatrix} \underline{h}_{k+1}^{(0)} \\ \underline{h}_{k+1}^{(1)} \\ \underline{h}_{k+1}^{(2)} \\ \vdots \\ \underline{h}_{k+1}^{(N)} \end{bmatrix} = \begin{bmatrix} \underline{h}_k^{(0)} \\ \underline{h}_k^{(1)} \\ \underline{h}_k^{(2)} \\ \vdots \\ \underline{h}_k^{(N)} \end{bmatrix} + e_k \begin{bmatrix} \mu_0 & 0 & 0 & \cdots & 0 \\ 0 & \mu_1 & 0 & \cdots & 0 \\ 0 & 0 & \mu_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \mu_N \end{bmatrix} \begin{bmatrix} 1 \\ \underline{x}_k \\ \underline{x}_k^{(2)} \\ \vdots \\ \underline{x}_k^{(N)} \end{bmatrix} \quad (3.29)$$

Această reprezentare permite alegerea unor pași de adaptare diferiți pentru nucleele de diferite ordine.

3.4 Implementări eficiente ale filtrului Volterra adaptiv

3.4.1 Construirea modelului de ordinul al-III-lea

În cele ce urmează se va prezenta implementarea unui filtru Volterra adaptiv, bazat pe algoritmul LMS, de ordinul al III-lea. Performanțele filtrului adaptiv vor fi verificate în cadrul unei aplicații privind identificarea neliniarităților de ordinul întâi și trei. Filtrul astfel realizat poate fi utilizat într-o aplicație practică de compensare a ecoului [G. Budura, 06].

Relația intrare-ieșire a filtrului Volterra este dată de relația (3.30). Așa după cum se poate observa s-a ales aceeași fereastră de filtrare, M , pentru nucleele de diferite ordine.

$$y[n] = \sum_{k_1=0}^{M-1} h_1[k_1] x[n-k_1] + \sum_{k_1=0}^{M-1} \sum_{k_2=0}^{M-1} h_2[k_1, k_2] x[n-k_1] x[n-k_2] + \sum_{k_1=0}^{M-1} \sum_{k_2=0}^{M-1} \sum_{k_3=0}^{M-1} h_3[k_1, k_2, k_3] x[n-k_1] x[n-k_2] x[n-k_3] \quad (3.30)$$

Primul pas în implementarea eficientă îl constituie definirea vectorilor de intrare de diferite ordine. Astfel, vectorul de intrare de ordinul întâi corespunzător unei ferestre de filtrare $M = 3$ este definit ca în relația de mai jos:

$$X^{(1)T} = [x[n] \ x[n-1] \ x[n-2]] \quad (3.31)$$

Vectorul de intrare de ordinul al II-lea se poate calcula cu formula:

$$X^{(2)} = X^{(1)} * X^{(1)T} \quad (3.32)$$

Pentru nuclee Volterra simetrice în relația intrare-ieșire vor fi selectate numai elementele $x_i x_j$ având $i \geq j$. În consecință, forma finală a vectorului de intrare de ordinul al-II-lea este:

$$X^{(2)T} = [x^2[n] \ x[n]x[n-1] \ x[n]x[n-2] \ x^2[n-1] \ x[n-1]x[n-2] \ x^2[n-2]] \quad (3.33)$$

Pentru construirea vectorului de intrare de ordinul al-III-lea, vom exprima produsele multiple între elementele semnalului de intrare prin elemente ale unor matrice obținute după cum urmează:

$$X^{(2)} * x[n] = \begin{bmatrix} x^3[n] & x^2[n]x[n-1] & x^2[n]x[n-2] \\ \cdots & x[n]x^2[n-1] & x[n]x[n-1]x[n-2] \\ \cdots & \cdots & x[n]x^2[n-2] \end{bmatrix} \quad (3.34)$$

$$X^{(2)} * x[n-1] = \begin{bmatrix} \dots & \dots & \dots \\ \dots & x^3[n-1] & x^2[n-1]x[n-2] \\ \dots & \dots & x[n-1]x^2[n-2] \end{bmatrix} \quad (3.35)$$

$$X^{(2)} * x[n-2] = \begin{bmatrix} \dots & \dots & \dots \\ \dots & \dots & \dots \\ \dots & \dots & x^3[n-2] \end{bmatrix} \quad (3.36)$$

Prin puncte au fost marcate acele elemente ale matricelor care nu vor intra în forma finală a vectorului de intrare de ordinul al-III-lea corespunzător unui nucleu de ordinul al-III-lea simetric.

$$X^{(3)T} = [x^3[n] \ x^2[n]x[n-1] \ \dots \ x[n]x^2[n-2] \ x^3[n-1] \ \dots \ x[n-1]x^2[n-2] \ x^3[n-2]] \quad (3.37)$$

3.4.2 Implementarea algoritmului LMS

Performanțele filtrului Volterra astfel implementat au fost testate în cadrul unei aplicații privind identificarea unui sistem neliniar ilustrată în figura 3.9.

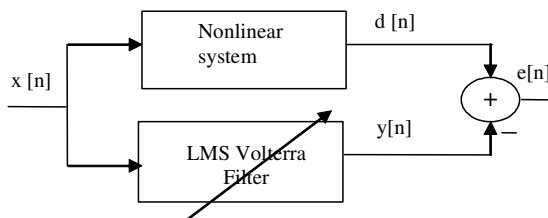


Fig.3.9 Utilizarea filtrului Volterra adaptive în identificarea neliniarităților

Pentru simplificarea calculului am implementat un filtru Volterra adaptiv ce conține doar nuclee de ordin impar. În acest caz matricea de intrare la momentul k este definită de relația:

$$X_k = \begin{bmatrix} X_k^{(1)T} \\ X_k^{(3)T} \end{bmatrix} \quad (3.38)$$

Dimensiunea matricii de intrare este determinată de dimensiunea vectorului de intrare de ordinul trei, $X_k^{(3)}$. Matricea coeficienților filtrului Volterra la momentul k este:

$$H_k = \begin{bmatrix} H_k^{(1)T} \\ H_k^{(3)T} \end{bmatrix} \quad (3.39)$$

Unde $H_k^{(1)T}$ și $H_k^{(3)T}$ se obțin pe baza relației 3.17 alegând $n=1$ respectiv $n=3$. Astfel pentru $H_k^{(3)T}$ se obține expresia:

$$H_k^{(3)T} = [h_k^{(3)}[0,0,0] \ h_k^{(3)}[0,0,1] \ \dots \ h_k^{(3)}[M-1, M-1, M-1]] \quad (3.40)$$

Ecuția de actualizare a coeficienților filtrului Volterra adaptive este în acest caz:

$\begin{bmatrix} H_{k+1}^{(1)T} \\ H_{k+1}^{(3)T} \end{bmatrix} = \begin{bmatrix} H_k^{(1)T} \\ H_k^{(3)T} \end{bmatrix} + e_k \begin{bmatrix} \mu_1 & 0 \\ 0 & \mu_3 \end{bmatrix} \begin{bmatrix} X_k^{(1)T} \\ X_k^{(3)T} \end{bmatrix}$	(3.41)
---	--------

Schema sistemului nelinier identificat este prezentată în figura 3.10. Prin LS s-a marcat un sistem liniar având răspunsul la impuls dat de relația 3.42.

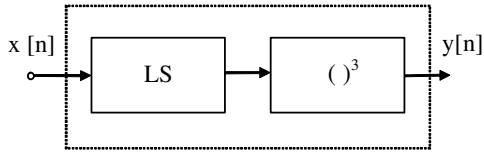


Fig.3.10 Sistem nelinier de ordinal al-III-lea cu memorie

$h_1(t) = \frac{1256}{0.98} \exp(-251.2t) \sin(1231t) \sigma(t)$	(3.42)
--	--------

Se calculează în continuare răspunsul la impuls al sistemului obținut prin discretizarea relației 3.42:

$h_1[n] = \frac{1}{T_e} h_1(nT_e)$	(3.43)
------------------------------------	--------

Unde T_e reprezintă perioada de eșantionare. Cunoscînd structura sistemului nelinier cu memorie, nuclee Volterra ale acestuia pot fi determinate pe cale analitică. Se obțin astfel:

$h_1[n] = 0$	(3.44)
$h_3[n_1, n_2, n_3] = h_1[n_1] h_1[n_2] h_1[n_3]$	

Expresii ce vor fi folosite pentru a compara nucleele determinate pe cale experimentală.

Pentru semnalul de intrare au fost alese secvențe ce conțin 2000 eșantioane de ygomot alb Gaussian de medie nulă. Întrucât nuclee ce s-a identificat au același tip de paritate s-a ales $\mu_1 = \mu_3$.

În figura 3.11 este reprezentată eroarea medie pătratică, $E\{(d[n] - y[n])^2\}$ după un număr de 140 pași de adaptare.

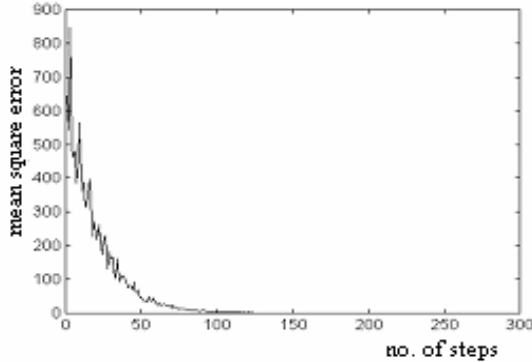


Fig.3.11 Evoluția erorii medii pătratice în cadrul algoritmului LMS

Acuratețea algoritmului implementat a fost apreciată pe baza erorii calculate conform relației 3.45 și reprezentată în figura 3.12.

$eh = 10 \lg \left(\sum_{k_1=0}^{M-1} \sum_{k_2=0}^{M-1} \sum_{k_3=0}^{M-1} (h_3[k_1, k_2, k_3] - \hat{h}_3[k_1, k_2, k_3])^2 \right)$	(3.45)
---	--------

Unde $h_3[k_1, k_2, k_3]$ este dat de relația 3.44 iar $\hat{h}_3[k_1, k_2, k_3]$ a fost determinat experimental pe baza algoritmului LMS.

Pentru a verifica robustețea algoritmului procedeul de adaptare a fost reluat în condițiile în care semnalului de ieșire $y[n]$ i s-a adăugat un zgomot inerent oricărei măsurări. S-au efectuat experimentele în două cazuri: zgomot aditiv având $\sigma^2 = 0.01$ și respectiv cu $\sigma^2 = 0.1$. Eroarea medie pătratică corespunzătoare celor două cazuri considerate este reprezentată în figura 3.13, respectiv 3.14.

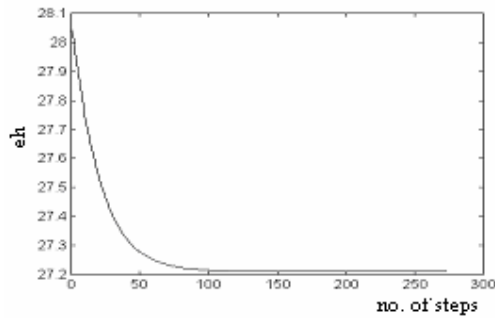


Fig.3.12 Curba de adaptare a coeficienților filtrului Volterra

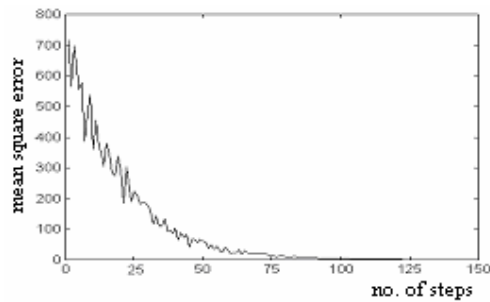


Fig.3.13 Eroarea medie pătratică ($\sigma^2 = 0.01$)

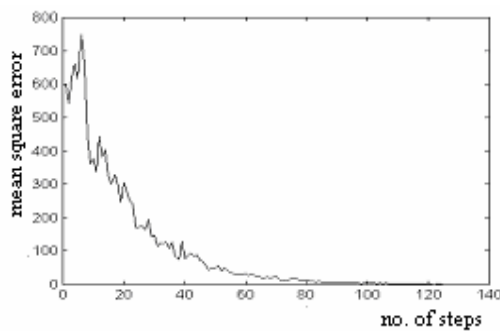


Fig 3.14 Eroarea medie pătratică ($\sigma^2 = 0.1$)

În figura 3.15 apar cele trei curbe de adaptare:

- cu linie punctată - $\sigma^2 = 0.1$
- cu linie continuă - $\sigma^2 = 0.01$
- cu linie și punct - fără zgomot

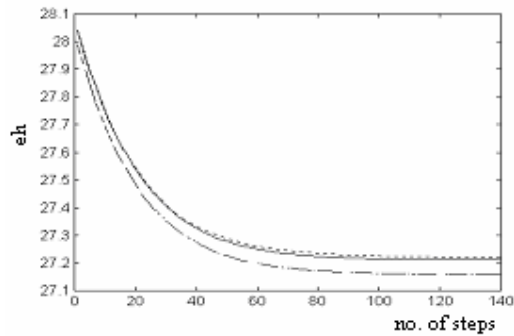


Fig.3.15 Curbele de adaptare a coeficienților în cele trei cazuri

Pentru ambele nuclee lungimea ferestrei a fost aleasă aceeași: $M=40$.

Performanțele algoritmului prezentat au fost apreciate pe baza răspunsului sistemului Volterra la semnale de intrare constând din sinusoidale de diferite frecvențe. De fiecare dată acest răspuns a fost comparat cu cel al sistemului real. Calculul răspunsului s-a făcut pe baza relației:

$$y[n] = H^{(3)T} \cdot X^{(3)} \quad (3.46)$$

Pentru a calcula răspunsul sistemului Volterra s-a folosit nucleul determinat experimental $H^{(3)T}$. Pentru sistemul real expresiile folosite sunt cele date de relația 3.44. Rezultatele experimentărilor sunt ilustrate în figurile 3.16- 3.18. În toate cazurile s-a reprezentat Transformata Fourier a semnalului de ieșire.

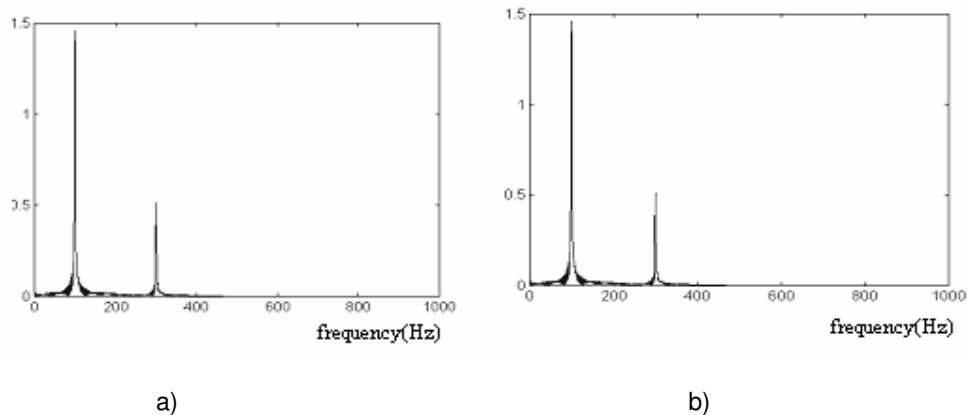


Fig. 3.16 Răspunsul sistemului Volterra a) și al celui real b) la un semnal sinusoidal de 100 Hz

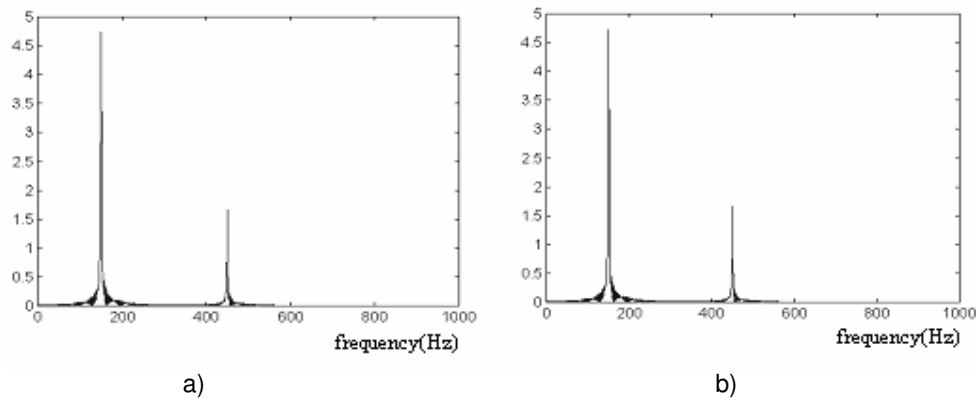


Fig. 3.17 Răspunsul sistemului Volterra a) și al celui real b) la un semnal sinusoidal de 150 Hz

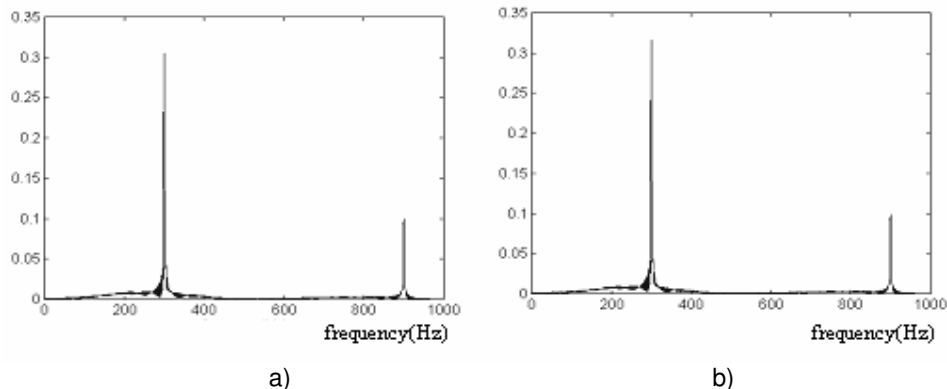


Fig. 3.18 Răspunsul sistemului Volterra a) și al celui real b) la un semnal sinusoidal de 300 Hz

6. Concluzii

În cadrul capitolului a fost prezentată o metodă adaptivă de determinare a nucleelor Volterra, ce se bazează pe algoritmul LMS. Implementarea metodei este precedată de un amplu studiu privind filtrarea neliniară adaptivă bazată pe algoritmul LMS. Sunt prezentate concluzii importante referitoare la alegerea pasului de adaptare. Este propus un algoritm de eficientizare a implementării filtrului Volterra. Sunt determinate cu ajutorul algoritmului LMS nucleele Volterra ale unui sistem de ordinul al-III-lea cu memorie. Calitatea aproximării a fost apreciată pe baza erorii calculate ca diferență între coeficienții reali și cei estimați cu ajutorul algoritmului adaptiv. S-a calculat și comparat răspunsul sistemului model cu cel al sistemului real. Deși metoda de identificare prezentată ridică probleme de convergență, precizia ei depinde în mai mică măsură de calitatea semnalului gaussian utilizat spre deosebire de metodele bazate pe intercorelație. Nucleele determinate prin tehnica adaptivă prezentată, sunt calitativ superioare celor determinate prin metoda Lee-Schetzen.

Bibliografie

- [S.C.Ahalt, 90] S.C.Ahalt, A.K. Krishnamurty, P.Chen,D.E.Melton, Competitive algorithms for vector quantization, Neural Networks, 3; pag. 277-291, 1990
- [S.B.Z. Azami, 00] S.B.Z. Azami, G.Feng " Robust Vector Quantizer Design Using Self-Organizing Neural networks" Signal Proc. Magazine, 80, www.elsevier.com/locate/sigpro/, 2000
- [B. Baykal, 97] B. Baykal, A. Constantinides, "A Neural Network Approach to the Underdetermined Order Recursive Least-squares Adaptive Filtering", Neural Networks vol.10, no.8, pag.1523-1531, 1997
- [C.Botoca, 00] Corina Botoca, Georgeta Budura, "Using Cellular Neural Network To Adaptive Equalization", Proceedings of the Symposium on Electronics and Telecommunications, pag.245-249, 2000
- [C.Botoca, 03] Corina Botoca Georgeta Budura, Neural Networks Intelligent Tools For Telecommuncations Poblemes, Buletinul stiintific al UPT tom 48 (62) Electronica si Telecomunicatii, fascicula 1, pag.51-58, 2003
- [C.Botoca, 04] Corina Botoca, N Miclau, , Georgeta Budura, " Nonlinear Complex Channel Equalization Using A Radial Basis Function Neural Network", NEUREL, Belgrade, Proceedings, 2004, pag. pag.73-78
- [S.Bouchired, 99.1] S.Bouchired, M. Ibnkahla, "Decision Neuronale Appliquee a L'egalisation De Canaux Satellitaires Mobiles" , Dix-septieme colloque , Grets, Vannes, Proceedings, pag.1125-1128, 1999
- [S. Bouchired, 99.2] S. Bouchired, Equalization of time-varying non-linear channels using neural networks: application to the satellite mobile channel, Ph.D. Thesis, INP, Toulouse, France,1999.

- [H.C.Card, 98] H.C.Card, S. Kamasu, Competitive learning and vector quantization in digital VLSI, *Neurocomputing*, 18; 195-227, 1998
- [C. Carson, 99]] C. Carson, M. Thomas, S. Belongie, J.M. Hellerstein and J. Malik, Blobworld: A system for region-based image indexing and retrieval, *Proceedings of Third Int. Conf. On Visual Information and Information Systems*, pp. 509-516, Amsterdam (Netherlands), 1999
- [I. Cha, 95] I. Cha , S.Kassam, "Channel Equalization Using Complex-Valued Radial Basis Function Networks", *IEEE. Journal Select. Areas Commun*, vol13, pag.122-131, Jan, 1995
- [C.H.Chang, 05] C.H.Chang, P. Xu, R.Xiao, T. Srikanthan, New adaptive quantization method based on self-organization, *IEEE Trans. on Neural Networks*, 16; 237-249, 2005
- [S. Chen, 90] S. Chen G. Gibson, C. Cowan, P. Grant, "Adaptive Equalization of Finite Non-linear Channels Using Multilayer Perceptron", *Signal Processing* 20, pag.107-119, 1990.
- [S.Chen , 91] S.Chen.C.F.Cowan, P.M.Grant Orthogonal least squares learning algorithm for radial basisi function networks, *IEEE.Transactions on Neural Networks*, 2 (2), 1001, pag.302-309, 1991
- [S. Chen, 94a] S. Chen, S. McLaughlin, B. Mulgrew, "Complex-valued radial basis function networks, Part I: Network Architecture and Learning Algorithms", *Signal Processing* 35, January, pag.19-31, 1994
- [S. Chen, 94b] S. Chen, S. McLaughlin, B. Mulgrew, "Complex-valued Radial Basis Function Networks, Part II: Application to Digital Communication Channel Equalization", *Signal Processing*, 36, March, pag. 175-188 , 1994
- [J.Choi, 93] J.Choi, S. Bang, B. Sheu, "A Programmable Analog VLSI Neural Network processor for Communication Receivers", *IEEE Trans. Neural Networks* ,May, pag. 484-495, 1993
- [J.M.Cioffi, 87] J.M.Cioffi, Limited-Precision Effects in Adaptive Filtering, *IEEE Transactions On Circuits And Systems*, vol. Cas-34, no. 7, July, pag.821-833, 1987
- [T.Clarkson, 01] L.Chua, T.Roska, " Cellular Neural Network and Visual Computing - Foundations and Applications", Cambridge University Press, 2001
- [T.Clarkson, 02] T.Clarkson, "Applications of Neural Networks in Telecommunications" , 2002
- [G.De Veciana, 92] G. De Veciana, A. Zakhor, "Neural Net-based Continuous Phase Modulation Receivers" , *IEEE Trans. Communications*, 40 August, 1992
- [C.Douligeris, 97] C.Douligeris, G.Develekos, "Neurofuzzy Control In ATM Networks", *IEEE Communi-cations Magazine*, May, pag.154-162, 1997
- [Dorffner , 99] Dorffner, Internal Report for NeuroNet, <http://klc.ac.uk/neuronet/> , 1999
- [K. Feher, 93] Feher, K., "Comunicații digitale avansate", Vol.1, Ed. Tehnică, București, 1993.
- [M.Fréchet,10] M Fréchet,., "Sur les Fonctionelles Continués", *Annals Scientifiques de L'Ecole Normale Superieure, Third Series, Vol.27*, pag. 193-216, 1910, citată în [J.Tsimbinos,95].
- [Q. Gan, 99] Q. Gan, P. Saratchandran, N. Sundararajan, K. R. Subramanian, "A Complex Valued Radial Basis Function Network for Equalization of Fast Time Varying Channels", *IEEE Trans. On Neural Networks*, vol. 10, issue 4, pag. 958-960, 1999
- [X.M. Gao, 97] X.M. Gao, X.Z. Gao, J. Tanskanen, S. Ovaska, "Power Prediction in Mobile Communication Systems Using an Optimal Neural Network Structure", *IEEE Trans. Neural Networks* 8,November,1997.
- [G.J. Gibson, 91] G.J. Gibson, S. Siu, C. F. N. Cowan, "The Application of Nonlinear Structures to The Reconstruction of Binary Signals" ,vol. 39, no. 8, August, 1991
- [R.Gray, 84] R.Gray , Vector quantization, *IEEE ASSP Magazine* , 4; 4-49, 1984
- [A. Guntsch, 98] A. Guntsch, M. Ibnkahla, G. Losquadro, M. Mazzella, D.Roviras, A. Timm, "EU's R&D Activities on Third Generation Mobile Satellite Systems ", *IEEE Communications Magazine*, February, 1998

- [D. Hamad, 98] D. Hamad, S. El Assad et J. Postaire, "Algorithmes d'apprentissage compétitif pour la classification automatique", Proceedings, TISVA'98, Oujda, Maroc, April, 1998
- [S. Haykin, 94.1] S. Haykin, L. Li, "Non Linear Adaptive Prediction of Nonstationary Signals", IEEE Trans. Signal Processing, 1995, pag.526-535
- [S. Haykin, 94.2] S. Haykin, "Neural Networks", Mcmillan Publishing Co., Englewood Cliffs, 1994
- [Hecht-Nielsen, 90] Hecht-Nielsen "Neurocomputing" Addison-Wesley Publishing Company, 1990
- [T. Hofmann, 98] T. Hofmann, J.M. Buhmann, "Competitive learning algorithms for robust vector quantization", *IEEE Transactions on Signal Processing*, 46;1665-1675, 1998
- [Y.H. Hu, 02] Y.H. Hu, Jenq-Neng Hwang, Handbook of Neural Networks Signal Processing, New York, CRC Press, 2002
- [A. Hussain, 97] A. Hussain, J. Soraghan, T. Durrani, "A New Adaptive Functional-link Neural Network-based DFE for Overcoming Co-channel Interference", IEEE Trans. Commun. 11, November, pag. 1358-1362, 1997
- [J. Hwang, 97] J. Hwang, S. Kung, M. Niranjani, J. Principe, "The Past, Present, and Future of Neural Networks for Signal Processing", IEEE Signal Process. Mag, November, pag.28-48, 1997
- [M. Ibnkahla, 97.1] M. Ibnkahla, J. Sombrin, F. CastanieH, N.J. Bershad, "Neural Networks for Modeling Non-linear Memoryless Communication Channels", IEEE Trans. Commun.7, July, pag.768-771, 1997
- [M. Ibnkahla, 97.2] M. Ibnkahla, "Applications of Neural Networks to Digital Communication a Survey" IEEE Signal Processing
- [M. Ibnkahla, 98] M. Ibnkahla, N.J. Bershad, J. Sombrin, F. CastanieH, "Neural Network Modeling and Identification of Non-linear Channels with Memory: Algorithms, Applications and Analytic models", IEEE Trans. Signal Process. , May, pag.1208-1220, 1998
- [M. Ibnkahla, 04] M. Ibnkahla, Q.M. Rahman, A.I. Sulyman, H.A. Al-Asady, J. Yuan, A. Safwat, "High-Speed Satellite Mobile Communications: Technologies and Challenges", Proceedings of the IEEE, Vol. 92, No. 2, February, pag. 312-339, 2004
- [S. Jagannathan, 01] S. Jagannathan, A. Tohmaz, J. Talluri, "Congestion Control of ATM Networks Using a Learning Methodology," Proceedings of 2001 IEEE International Conference on Control Applications, 2001
- [D. Jianping, 02] D. Jianping, Narasimhan Sundararajan, P. Saratchandran, "Communication Channel Equalization Using Complex-Valued Minimal Radial Basis Function Neural Network", IEEE Trans. On Neural Networks, Vol 13, No.3, May, pag.687-696, 2002
- [A. Kantsila, 04] A. Kantsila, M. Lehtokangas, J. Saarinen, "Complex RPROP-algorithm for Neural Network Equalization of GSM Data Bursts", accepted to Neurocomputing magazine, 2004
- [Kartaloupoulos, 99] Kartaloupoulos Internal Report for IEEE Communications Society, <http://www.comsoc.org/1999>
- [G. Kechriotis, 94] G. Kechriotis, E. Zervas, E. S. Manolakos, "Using Recurrent Neural Networks for Adaptive Communication Channel Equalization", IEEE Trans on Neural Networks, vol. 5, issue 2, pag. 267-278, 1994
- [G. Kechriotis, 96] G. Kechriotis, E. Manolakos, "Hopfield Neural Network Implementation of The Optimal CDMA Multiuser Detector", IEEE Trans. Neural Networks, 1, January, pag.131-141, 1996
- [T. Kim, 02] T. Kim, T. Adali, "Fully complex multi-layer perceptron network for nonlinear signal processing," Journal of VLSI Signal Processing, Vol. 32, Iss. 1, pag. 29-43, Aug./Sep. 2002.
- [T. Koh, 85] T. Koh, E.J. Powers, "Second Order Volterra Filtering and its Application to Nonlinear System Identification" IEEE Trans on ASSP, Vol. ASSP-33, No. 6, pag.1445-1455, decembrie 1985.
- [T. Kohonen, 96] T. Kohonen, E. Oja, O. Simula, A. Visa, J. Kangas, "Engineering applications of the self-organizing map", IEEE Proc., October, 1358-1384, 1996
- [B. Kosko, 92] B. Kosko "Neural Networks and Fuzzy Systems", Prentice Hall, Englewood Cliffs, NJ, 1992

- [A.K.Y.Lai, 02] A.K.Y.Lai, K.K.M.Cheng M.N.S.Swamy "Neural Methods For Antenna Array Signal Processing: A Review", Signal Processing Magazine, www.elsevier.com/locate/sigpro, 2002
- [L.T.Law, 03] L.T.Law, Y.M.Cheung, Color image segmentation using rival penalized controlled competitive learning, *Proceedings of International Joint Conference on Neural Networks (IJCNN'2003)*, July 20-23, Portland (Oregon, USA), 2003
- [J. Lee, 96] J. Lee, C. D. Beach, N. Tepedelenlioglu, "Channel Equalization Using Radial Basis Function Network", IEEE Int. Conf. On Neural Networks, vol. 4, pag. 1924-1928, 1996.
- [J.Lee, 99] J.Lee, Ch.Beach, N.Tepedelenlioglu, A Practical Radial Basis Function Equalizer, IEEE Transactions on Neural Networks, vol10,nr.2,pag.450-455, march, 1999
- [N Leitao , 00] N Leitao, L.Ahmeida, "Neural Networks In BIDN Flow Control:ATM Traffic Prediction Or Network Modeling", IEEE Communications Mag,October,1995, pag.50-56
- [Lester, 00] Lester, S.H.Ngi, "Efficient Training of Neural Nets for Nonlinear Adaptive Filtering Using a Recursive Levenberg–Marquardt Algorithm", IEEE Transactions On Signal Processing, vol. 48, no. 7, July, 2000
- [R.Logeswaran. 01] R.Logeswaran, "Transmission Issues of Artificial Neural Networks in a Prediction/ Based Lossless Data Compression Scheme", Proceedings IEEE ICT,, pag.578-583, 2001
- [N.Lo, 92] N.Lo, H.Hafez, "Neural Network Channel Equalization", Proceedings of International Joint Conference on Neural Networks , IJCNN'92, Vol. 2, pag. 981-986, 1992
- [V.J.Mattews, 91] V.J.Mattews, "Adaptive Polynomial Filters", IEEE Signal Processing Magazine, Vol. 8, No. 3, pag. 10-26, iulie 1991.
- [U. Mitra, 95] U.Mitra, V.Poor, "Adaptive Receiver Algorithms for Near-far Resistant CDMA", IEEE Transactions on Communications 43, February/March/April, pag.1713-1724, 1995
- [M. Moonen, 97] M. Moonen, Y.Dologlou, G.Rombouts, "Combination Of Multimicro phone Echo And Noise Suppression Techniques For Enhancement Of Audio Signals", www.esat.kulueven.ac.be/sista/yearreport97
- [R.E. Morthensen, 04] R.E.Morthensen, "Random Signals and Systems", John Wiley & Sons, 1987.
- [T. Nakamura, 98] T. Nakamura, K. Terada, A. Shibata, J. Morimoto, H. Adachi, H. Takeda, Development of a cheap on-board vision mobile robot for robotic soccer research, *Proceeding of International Conference on Intelligent Robots and System*, pp. 431-436, Oct., Victoria (BC, Canada) 1998
- [E.Nordstrom, 95] E.Nordstrom, J.Carlstrom, O.Gällmo, "Neural Networks For Adaptive Traffic Control In ATM Networks", IEEE Communications magazine, vol 33, no. 10, pag.43-49, 1995
- [J.Nie, 99] J.Nie, S.Haykin, A Dynamic Channel Assignment Policy through Q-Learning, IEEE Transactions on Neural Networks, vol 10, no.6, pag.1443-1455, 1999
- [J.C.Patra, 99] J.C.Patra, R.N.Pal, R.Baliarsingh, G.Panda, Nonlinear Channel Equalization For Qam Constellation Using Artificial Neural Networks, Trans. Syst. Man, Cyb, vol 29,april, pag.162-271, 1999
- [R.Parisi, 97] R.Parisi, E.Di Claudio, G. Orlandi, B. Rao, "Fast Adaptive Digital Equalization by Recurrent Neural Networks", IEEE Trans. Signal Process. 45, November, 1997, pag.2731-2739, 1997
- [R.Perfetti, 93] R.Perfetti, "Cellular Neural Networks for Fast Adaptive Equalization", International Journal of Theory and Applications, vol.21,pp 165-175, 1993
- [T.Pitas, 90] Pitas, T., Venetsanopoulos, A., N., " Nonlinear Digital Filters; Application", Kluwer Academic Publishers, 1990.
- [I. Popescu, 00] I.Popescu, I.Naforniță, P. Constantinou, A. Kanatas, N. Moraitis, "Prediction of Outdoor Propagation Path Loss with Neural Networks", Proceedings of the 8th telecommunications Forum Telfor'2000, 21-23 November, Belgrade, Serbia, pag. 140-143, 2000

- [I.Reljin, 97] I.Reljin "Neural Network Based Low Cell Scheduler", Proceedings of 4th Neural Network Applications in Electrical Engineering, NEUREL'97, Belgrade, pag. 168-172, 1997
- [B.Reljin, 00] B.Reljin, I.Reljin, "Neural Network In Telegraphic Control. Pro Et Contra ", IEEE Proceedings of 4th International Conference On Telecommunications In Modern Satellite Broadcasting, TELSIS'99, pp 518-527, 1999
- [W.J.Rugh, 81] W..J.Rugh,, "Nonlinear System Theory: The Volterra Wiener Approach", John Hopkins Univ. Press, Baltimore, MD, 1981.
- [L.M.San, 02]. .M.San, J.Revuelta "A neuro-evolutionary framework for Bayesian blind equalization in digital communications " Signal Processing Magazine, 2002, www.elsevier.com/locate/sigpro
- [W.I. Sandberg, 83] I.W.Sandberg, "Series Expansions for Nonlinear Systems", Circuits, Systems and Signal Processing, Vol. 2, No. 1, pag. 77-87, februarie 1983.
- [S.Sezer, 01] S.Sezer, Ph.Power, Neural Networks for High-Bandwidth QAM Channels, Proceedings IEEE ICT, pag.225-229, 2001
- [M.Schetzen, 89] M Schetzen, "The Volterra and Wiener Theories of Nonlinear Systems, Reprint Edition, Krieger, Malabar, Florida, 1989.
- [G.L.Sicuranza, 93] G. L.Sicuranza, "Theory and Approximation of Polynomial Filters, Proc. IEEE, Vol.3, pag. 931-937,1993.
- [B.E. Shi, 98] B.E.Shi, Gabor-Type Filtering in Space and Time with Cellular Neural Networks, IEEE Transactions On Circuits And Systems—I: Fundamental Theory And Applications, vol. 45, no. 2, February, pag.121-132, 1998
- [C.Srivaree-ratana, 99] C.Srivaree-Ratana, A.E.Smith, Estimation of All Terminal Networks Realilty using an Artificial Neural Network, Computer and Operation Research, vol 29, issue7, pag.849-868
- [A.Taraff, 95] A.Taraff, I.Habib, T.Saadowi, "Intelligent Traffic Control For ATM Broadband Networks" , IEEE Communications Magazine, October, pag. 76-82, 1995
- [Texas Instruments, 86] *** Texas Instruments, "Digital Signal Processing Application with TMS320 Family", 1986.
- [J.Tsimbinos,95] J.Tsimbinos, "Identification and Compensation of Nonlinear Distorsion", Thesis, University of South Australia, School of Electronic Engineering, feb. 1995.
- [M.Varanasi, 90] M. Varanasi, B. Aazhang, "Multistage Detection in Asynchronous Code-division Multiple Access Communications", IEEE Trans. Commun. 38, April, pag. 509-519, 1990
- [X.Wang, 02] X.Wang, H.Liu, J.Lu, T.Yohagy, Combining recurrent neural networks with self-organizing maps for channel equalization, *IEEE Transactions on Communications*, 85;2227-2235,2002
- [K.Weierstrass,03] Weierstrass, K., "Über die analytische Darshell-bankeit, sogenanter willkürlicher Funktionen reeler Argumente", Math. Werke, III Bd., 1903, citată în [J.Tsimbinos,95]
- [H.Yu, 97] H.Yu, S.Bang, "An Improved Time Series Prediction by Applying the Layer-by-layer Learning Method to FIR Neural Networks", Neural Networks 10, Nov,pag.1717-1729,1997
- [L.Xu, 93] L.Xu, A.Krzyzak, A.E.Oja, Rival penalized competitive learning for clustering analysis. RBF net and curve detection. IEEE Trans. on Neural Networks, 4; 636-648, 1993
- [K.Yamazaki, 91] K.Yamazaki, S.Aly, D.Falconer, "Convergence behavior of a jointly adaptive transversal and memory-based echo canceller", IEE Proceedings-F, Vol. 138, no.4, 1991, pp 361-370.
- [I.Zurada, 92] I.Zurada , Introduction to Artificial Neural Systems. Learning and Applications, West Publishing Co.,Wien,1992
- [*, 95] "Making Brain Waves", CIO Magazine, January 1995, [<http://www.cio.com>]
- [**, 02] "Time varying Channel Equalization", 2002, www.soma.mcmaster.ca/ASLWeb/Projects/ChannelEqualization/