

Capitolul 3

Filtrare adaptivă

- 3.1 Introducere
- 3.2 Algoritmi de filtrare adaptivă
- 3.3 Algoritmi adaptivi în domeniul frecvență
- 3.4 Discuție asupra algoritmilor adaptivi
- 3.5 Aplicații ale filtrelor adaptive

Cuvântul *adaptiv* este întâlnit deseori în vorbirea curentă, în contexte care nu au neapărat o conotație tehnică. Adaptarea vitezei de rulare a unei mașini condițiilor meteorologice, adaptarea unei plante la un mediu ostil sau adaptarea unei persoane la un nou loc de muncă sunt exemple care ilustrează un același principiu: modificarea comportării unui sistem fizic sub acțiunea unor factori externi în sensul optimizării performanțelor acestuia. Aceleași exemple practice pun în evidență și câteva caracteristici specifice oricărui proces adaptiv: a) modificările nu se petrec brusc, ci gradual, iar efectul acestora poate avea, temporar, un efect nedorit, deși pe ansamblu conduc la efectul scontat (controlul unui autoturism pe un drum acoperit de polei este un exemplu edificator în acest sens). În plus, modificările țin cont în mod nemijlocit de diferența existentă între comportarea considerată “dorită” și cea reală; b) pe măsură ce procesul de adaptare evoluează, iar comportarea sistemului analizat se apropie de cea dorită, amplitudinea modificărilor succesive se diminuează; c) de regulă, putem cuantifica “ținta” înspre care tinde sistemul. În multe situații practice, chiar dacă obiectivul nu este atins cu exactitate, plasarea performanțelor “suficient de aproape” de acesta este acceptabilă; d) este posibil ca uneori procesul de adaptare să scape de sub control, iar rezultatele să fie complet nedorite.

În cuprinsul acestui capitol vom trece în revistă principiile de operare, terminologia, parametrii caracteristici și ariile de aplicabilitate ale principalilor algoritmi adaptivi descriși în literatură. Fără a diminua importanța versiunilor neliniare ale tehnicilor adaptive, accentul va fi pus pe prezentarea metodelor cu caracter liniar care, pe de o parte, beneficiază de posibilitatea analizei riguroase a dinamicii sistemelor considerate, iar pe de altă parte domină cu autoritate “piața” aplicațiilor practice. Ne vom concentra cu precădere pe algoritmul de tip “scădere după gradient”, respectiv pe implementarea practică a acestuia sub forma familiei de algoritmi de tip LMS (*Least Mean-Squares*), a căror eficiență va fi ilustrată prin intermediul unor exemple sugestive. Nu în ultimul rând, vom evidenția și o serie de elemente specifice implementării algoritmilor adaptivi prin metode digitale.

Înainte de a intra efectiv în subiectul filtrării adaptive este important să recapitulăm sumar câteva dintre noțiunile fundamentale specifice teoriei *proceselor*

aleatoare, care oferă suportul matematic necesar formalizării riguroase a diverselor aspecte specifice acestui domeniu. În Dicționarul Explicativ al limbii române întâlnim următoarea definiție a cuvântului *aleator*:

Aleator: care depinde de un eveniment nesigur;
supus întâmplării

Sunetul emis de un receptor radio dezacordat, imaginea de pe ecranul televizorului la întreruperea accidentală a emisieii, distanța totală pe care o parcurgem pe jos în fiecare zi, ori rezultatul aruncării cu zarul reprezintă exemple care ilustrează în mod concret cele de mai sus. Caracteristica distinctivă a unor astfel de *semnale* (în sensul de manifestări fizice purtătoare de informație) constă în faptul că acestea nu pot fi descrise prin relații matematice explicite, care să permită precizarea cu exactitate a valorii lor la un moment dat. În fapt, evoluția acestora este ghidată de *legi probabilistice*, care presupun un anumit grad de șansă, de întâmplare pentru ca manifestarea respectivă să conducă la un rezultat predefinit. Din acest punct de vedere, semnalele aleatoare se deosebesc de cele *deterministe*, ale căror valori pot fi precizate cu acuratețe în orice moment.

Suportul teoretic care permite analiza semnalelor cu caracter aleator este oferit de *teoria probabilităților* care, în principiu, analizează **valorile medii** ale unor manifestări fizice care se produc pe scară largă (de exemplu succesiunea simbolurilor binare în transmisiunile de date, rezultatele jocurilor de noroc, ori evoluția seriilor financiare). Puntea de legătură dintre multitudinea de manifestări fizice concrete și formalismul matematic unificator îl joacă noțiunea de *variabilă aleatoare*, exprimată prin **funcția** ce asociază câte un număr fiecărui rezultat posibil al unui experiment dat.

Un ansamblu de variabile aleatoare definește un *proces aleator*. În Fig. 3.1 se prezintă câteva *realizări individuale* ale unui proces aleator fictiv. Reprezentarea grafică este caracteristică unui proces continuu în timp (analogic) cu valori necuantizate (de exemplu, un semnal radio), însă putem întâlni și procese aleatoare discrete în timp, cu valori cuantizate (rezultatul aruncării cu zarul) sau necuantizate (cantitatea totală de apă băută în decursul unei zile). În exemplele enumerate se

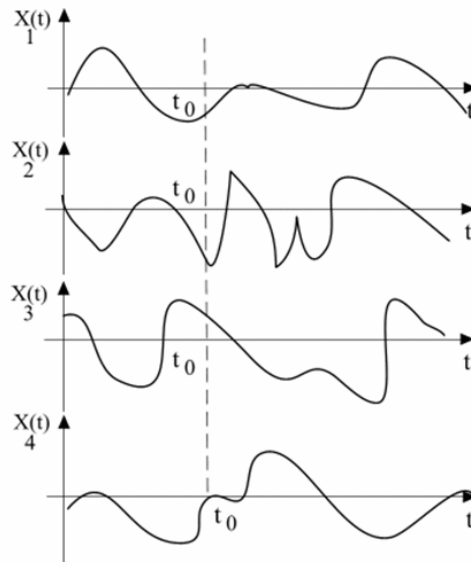


Fig. 3.1 Realizări individuale ale unui proces aleator

presupune că avem la îndemână mai mulți “subiecți” care permit înregistrarea succesivă a rezultatelor instantanee ale experimentelor propuse.

Un rol central în teoria probabilităților îl joacă operatorul denumit “**speranță matematică**” (*expectation*) care atașează unei variabile aleatoare valoarea rezultată prin medierea aritmetică a unui număr teoretic infinit de realizări individuale ale manifestării fizice considerate. În funcție de natura discretă sau analogică a experimentului, definițiile sunt [7]:

$$\begin{aligned}
 E\{\mathbf{X}\} &= \sum_{i=1}^n p_i x_i \\
 E\{\mathbf{X}\} &= \int_{-\infty}^{\infty} x f_X(x) dx
 \end{aligned}
 \tag{3.1}$$

în care \mathbf{X} desemnează variabila aleatoare presupusă a avea n realizări posibile cu probabilitățile p_i în cazul discret, respectiv *densitatea de probabilitate* $f_X(x)$ în cazul analogic.

În cele ce urmează ne vom referi exclusiv la procese aleatoare discrete în timp, ale căror realizări individuale sunt denumite *serii de timp*. Astfel de semnale se întâlnesc în mod natural în numeroase aplicații practice sau pot proveni în urma

eșantionării unor semnale analogice. După cum s-a menționat anterior valorile exacte ale unor variabile aleatoare (și, în consecință, ale unor procese aleatoare la un moment dat) nu pot fi precizate cu exactitate. În schimb, pot fi determinate mărimi statistice care oferă o caracterizare parțială a procesului considerat. Astfel, pentru un proces aleator oarecare notat $\mathbf{u}[n]$ se definesc următoarele mărimi [4]:

$$\text{Valoarea medie:} \quad \mu[n] = E\{\mathbf{u}[n]\} \quad (3.2)$$

$$\text{Funcția de autocorelație:} \quad r[n, n-k] = E\{\mathbf{u}[n]\mathbf{u}^*[n-k]\} \quad (3.3)$$

$$\text{Funcția de autocovarianță:} \quad c[n, n-k] = E\{(\mathbf{u}[n] - \mu[n])(\mathbf{u}[n-k] - \mu[n-k])^*\} \quad (3.4)$$

Această formă de caracterizare parțială oferă două avantaje majore: a) se pretează la măsurători practice; b) permite procesarea liniară a proceselor aleatoare.

Un proces aleator este denumit **staționar** dacă toate mărimile statistice care îl caracterizează nu depind în mod explicit de timp. În acest caz relațiile de mai sus capătă forme mai simple, după cum urmează:

$$\text{Valoarea medie:} \quad \mu[n] = \mu, \quad \forall n \quad (3.5)$$

$$\text{Funcția de autocorelație:} \quad r[n, n-k] = r[k], \quad \forall n \quad (3.6)$$

$$\text{Funcția de autocovarianță:} \quad c[n, n-k] = c[k], \quad \forall n \quad (3.7)$$

Mărimea $c[0] = \sigma_u^2$ desemnează *dispersia* procesului aleator. Un rol important îl joacă *matricea de autocorelație*, definită astfel (simbolul \mathbf{H} desemnează operația de *transpunere Hermitică*, obținută prin combinația dintre conjugarea complexă și transpunerea uzuală):

$$\mathbf{R} = E\{\mathbf{u}[n]\mathbf{u}^H[n]\} = \begin{bmatrix} r[0] & r[1] & \dots & r[M-1] \\ r[1] & r[0] & \dots & r[M-2] \\ \vdots & \vdots & \ddots & \vdots \\ r[-M+1] & r[-M+2] & \dots & r[0] \end{bmatrix} \quad (3.8)$$

3.1 Introducere

Metoda cea mai răspândită prin care se introduc, în literatura de specialitate, algoritmi adaptivi o reprezintă exemplul binecunoscutei *probleme de filtrare (liniară) optimă* din teoria transmisiunii informației [4], descrisă generic prin schema-bloc din Fig. 3.2. În esență, aplicația urmărește determinarea parametrilor care definesc un filtru (liniar) la intrarea căruia se aplică un semnal care include două componente, una considerată utilă, iar cealaltă nedorită, astfel încât răspunsul acestuia să fie cât mai “curat”, adică să conțină cu precădere semnal util. În figură se distinge prezența unui semnal de eroare rezultat din compararea răspunsului real al sistemului cu cel dorit, ce urmează a fi utilizat (tipic, împreună cu semnalul de intrare) pentru a determina valorile parametrilor filtrului. Acesta poate fi analogic sau discret, liniar sau neliniar, iar semnalele de intrare, respectiv de ieșire dorită, sunt considerate, în general, *realizări individuale ale unor procese aleatoare*¹. Deși prezente în literatură, filtrele adaptive analogice sunt foarte rar utilizate, astfel încât în cele ce urmează ne vom ocupa numai de varianta discretă a acestora. De asemenea, vom aborda doar tangențial cazul filtrelor adaptive neliniare, teoria acestora fiind pe larg descrisă în lucrările dedicate așa-numitelor *rețele neurale artificiale* [2].

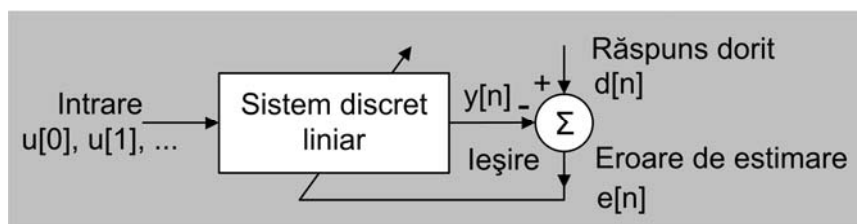


Fig. 3.2 Schema-bloc a unui filtru adaptiv

¹ Cuvântul care ilustrează intuitiv noțiunea de semnal aleator este *zgomotul*. De regulă, acest cuvânt este asociat cu senzația de disconfort, care trebuie înlăturată sau măcar atenuată. Deseori, zgomotul se suprapune peste ceea ce denumim de regulă “informație utilă”, iar efortul de a le separa poate fi uneori extrem de anevoios.

În legătură cu schema-bloc din Fig. 3.2 se pot face câteva observații:

- deși caracterul liniar al filtrului utilizat nu este implicit, o astfel de condiție simplificatoare a permis obținerea unor rezultate teoretice importante, referitoare în particular la determinarea, într-o formă compactă și elegantă, a expresiei setului optim de coeficienți ai filtrului sau a valorii erorii de estimare.
- filtrul este cu funcționare discretă în timp, cu avantajul particular că algoritmi de procesare pot fi implementați folosind circuite digitale specializate. Mai mult, în cele ce urmează vom utiliza în exclusivitate filtre discrete cu răspuns finit la impuls (*Finite Impulse Response* - FIR) datorită stabilității intrinseci a acestora.
- ieșirea filtrului, notată $y[n]$, furnizează o *valoare estimată* a unui semnal dorit $d[n]$. Diferența dintre aceste semnale constituie *eroarea de estimare* $e[n]$. În condițiile în care atât semnalul de intrare cât și cel dorit reprezintă realizări individuale ale unor procese aleatoare, eroarea devine ea însăși un proces aleator cu caracteristici statistice proprii. Scopul urmărit este cât se poate de evident: minimizarea erorii de estimare conform unui criteriu statistic precizat. Acesta este ales de regulă dintre următoarele variante: a) valoarea pătratică medie a procesului $e[n]$; b) media aritmetică a valorilor absolute ale erorii; c) media aritmetică a unor puteri de ordin superior ale valorilor absolute ale erorii. Teoria filtrării optime utilizează prima dintre cele 3 variante dintr-un motiv justificat: doar în această situație *funcția de cost* (indexul de performanță), a cărei minimizare va furniza ca rezultat valorile optime ale coeficienților filtrului, este o **funcție convexă cu o valoare minimă unică**.

Fără a prezenta o demonstrație matematică care să conducă la soluția problemei de filtrare liniară optimală expusă anterior, menționăm direct rezultatul de interes, exprimat sub forma celebrelor **ecuații Wiener–Hopf** [4]:

$$\mathbf{R} \mathbf{w}_{opt} = \mathbf{p} \Rightarrow \mathbf{w}_{opt} = \mathbf{R}^{-1} \mathbf{p} \quad (3.9)$$

în care \mathbf{w}_{opt} desemnează valorile optime ale coeficienților filtrului (denumit în acest context *filtru Wiener*), \mathbf{R} este matricea de autocorelație a procesului aleator aplicat la intrare, iar \mathbf{p} este vectorul de intercorelație dintre intrare și ieșirea dorită:

$$\mathbf{p} = E\{\mathbf{u}[n]d^*[n]\} \quad (3.10)$$

Este ușor de observat că determinarea valorilor setului de coeficienți ai unui filtru optimal Wiener presupune cunoașterea exactă a proprietăților statistice ale datelor prelucrate. Când aceste informații nu sunt disponibile (iar aceasta este situația tipică întâlnită în practică!) vectorul \mathbf{w}_{opt} nu se poate calcula, iar dacă se încearcă rezolvarea ecuațiilor de mai sus folosind valori estimate (și deci imprecise) ale matricii \mathbf{R} și vectorului \mathbf{p} valorile coeficienților filtrului nu vor mai fi optime.

Ca urmare, în aceste condiții se dovedește utilă folosirea unui **filtru adaptiv**, adică a unui filtru care se “autoproiectează” astfel încât răspunsul acestuia să se apropie cât mai mult (în sens statistic) de cel dorit. Din nou sunt necesare câteva precizări:

- funcționarea unui filtru adaptiv se bazează pe utilizarea unui algoritm (a unei “rețete”) care permite modificarea într-o manieră *recursivă* a valorilor setului de coeficienți - aleși inițial în mod arbitrar! - astfel încât să fie asigurată convergența **în sens statistic** către valorile optime corespunzătoare soluției ecuațiilor Wiener-Hopf.

- după cum vom vedea, în decursul procesului de adaptare valorile setului de coeficienți depind în mod explicit de valorile semnalelor de intrare, astfel încât un filtru adaptiv este în realitate un sistem *neliniar*, în sensul că nu respectă principiul superpoziției (deși răspunsul filtrului este totuși obținut sub forma unei combinații liniare a semnalelor de intrare).

În decursul timpului au fost elaborate numeroase variante de algoritmi adaptivi, fiecare cu avantaje și dezavantaje specifice. Se pot distinge următoarele 3 abordări de principiu [4]:

a) **Filtrul Wiener:** ecuațiile Wiener-Hopf prezentate anterior sunt rescrise sub o formă convenabilă folosind o tehnică binecunoscută de optimizare denumită “scădere după gradient” (*gradient descent*). Deoarece în continuare ecuațiile includ valorile matricii \mathbf{R} și vectorului \mathbf{p} (presupuse necunoscute), se înlocuiesc valorile *exacte* ale acestora cu valori *estimate* (în fapt, cu valorile instantanee). Algoritmul obținut este denumit LMS (*Least Mean-Squares*) și reprezintă în multe situații referința în raport cu care se compară performanțele altor algoritmi.

b) **Filtrul Kalman:** teoria filtrului optimal Wiener a fost elaborată pentru procese aleatoare staționare. În cazul în care proprietățile statistice ale proceselor aleatoare

implicate se modifică în timp abordarea anterioară devine mult mai dificilă deoarece suprafața de eroare al cărei minim este căutat se modifică în permanență, astfel încât algoritmul adaptiv trebuie să asigure nu numai convergența către soluția optimă, dar și urmărirea modificării neîncetate a acestei valori optime. Soluția este oferită de teoria filtrului Kalman, care admite drept punct de pornire formularea unui model al aplicației considerate sub forma ecuațiilor de stare. Algoritmul recursiv rezultat este mult mai rapid decât algoritmul LMS și mai puțin dependent de caracteristicile statistice ale datelor de intrare, însă presupune un volum de calcul considerabil sporit.

c) **Metoda *Least-Squares***: cele 2 tehnici prezentate anterior utilizează în mod explicit o abordare statistică, bazată pe considerarea acțiunii operatorului $E\{\cdot\}$ asupra unui ansamblu de realizări individuale ale unor procese aleatoare. Un punct de vedere complementar este oferit de luarea în considerație a mediilor aritmetice calculate în domeniul timp, pe câte o singură realizare individuală a procesului aleator de intrare, respectiv de ieșire. Problema de filtrare liniară optimală se formulează în acest caz într-o manieră principal asemănătoare cu ecuațiile Wiener-Hopf (relațiile poartă denumirea de *ecuații normale*), iar rezolvarea lor iterativă este oferită de algoritmul *Recurrent Least-Squares*.

Funcția de cost

După cum s-a arătat anterior, coeficienții unui filtru adaptiv suferă un proces repetitiv de modificare sub acțiunea unui algoritm menit să conducă la obținerea unui răspuns cât mai “apropiat” de cel dorit, asemănarea fiind apreciată conform unui criteriu statistic precizat. Unul dintre cele mai des utilizate puncte de vedere în definirea cadrului matematic al unei astfel de aplicații îl reprezintă *teoria optimizării*. Conform acestei abordări fiecărui set posibil de valori ale vectorului de coeficienți ai filtrului adaptiv i se asociază o valoare scalară, corespondența fiind asigurată de o așa-numită *funcție de cost*. Scopul urmărit este de a identifica valori particulare ale setului de parametri pentru care acest index de performanță atinge valori extreme.

În cea mai mare măsură, teoria filtrării adaptive se bazează pe utilizarea criteriului denumit *eroare pătratică medie*, definit pe baza erorii de estimare în felul următor:

$$J(\mathbf{w}) = E\{|e[n]^2|\} = E\{|d[n] - y[n]|^2\} \quad (3.11)$$

unde $y[n] = \mathbf{w}^H \mathbf{u}[n]$ desemnează ieșirea filtrului adaptiv. Definiția anterioară include acțiunea operatorului $E\{\cdot\}$ asupra unui ansamblu de realizări individuale ale proceselor aleatoare implicate și este specifică abordării de tip filtru Wiener. În mod alternativ, se poate utiliza și operația de mediere în domeniul timp, specifică abordării de tip *Least-Squares* și care conduce la următoarea definiție [4]:

$$J(\mathbf{w}, n) = \sum_{i=1}^n |e[i]|^2 = \sum_{i=1}^n |d[i] - \mathbf{w}^H \mathbf{u}[i]|^2 \quad (3.12)$$

Este important de observat că în acest caz expresia funcției de cost depinde în mod explicit și de intervalul de timp considerat, astfel încât valorile optime ale vectorului de coeficienți se vor modifica ele însele în timp.

Avantajul major al unei astfel de funcții de cost rezidă în caracterul convex al acesteia, manifestat prin prezența unei **valori minime unice** care va conduce la determinarea unui set de asemenea unic de valori optime ale coeficienților filtrului adaptiv considerat. Merită menționat în acest context faptul că această caracteristică nu este prezentă în cazul unor filtre adaptive discrete cu răspuns infinit la impuls (*Infinite Impulse Response* - IIR) și nici al celor neliniare (implementate sub forma rețelelor neurale), situații în care sunt prezente numeroase valori minime locale, corespunzătoare unor soluții suboptimale. În aceste condiții, alegerea judicioasă a valorilor inițiale ale setului de coeficienți ai filtrului este determinantă pentru obținerea unor performanțe acceptabile¹.

În Fig. 3.3 sunt ilustrate modalitățile tipice de reprezentare ale funcției de cost sub forma unei suprafețe de eroare, respectiv prin curbe de nivel constant (de forma unor elipse). Valoarea minimă a funcției de eroare (atinsă pentru $\mathbf{w} = \mathbf{w}_{opt}$) este:

$$J_{min} = \sigma_d^2 - \mathbf{p}^H \mathbf{W}_{opt} \quad (3.13)$$

¹ Acesta este motivul pentru care, în practică, în astfel de situații se vor efectua experimente repetate, pornind din condiții inițiale diferite.

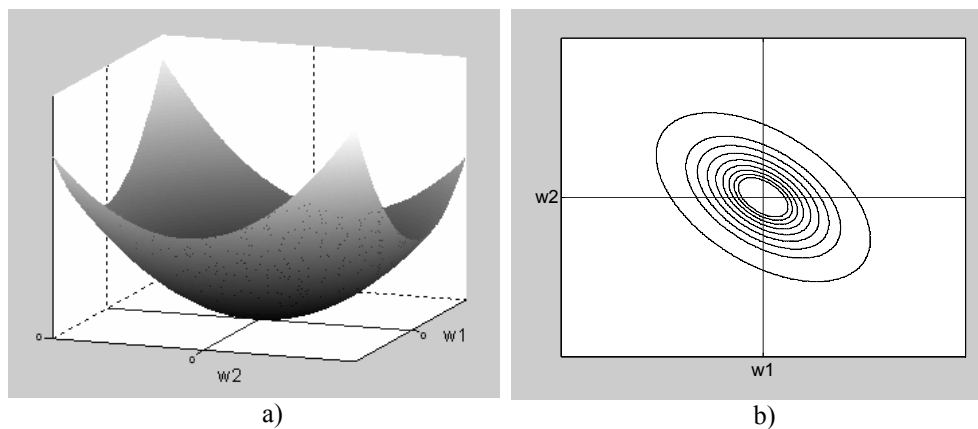


Fig. 3.3 Eroarea pătratică medie: a) hipersuprafață parabolică; b) curbe de nivel constant

Parametri specifici

Există o serie de parametri care caracterizează modul de operare al unui filtru adaptiv, utili în a aprecia comparativ performanțele numeroaselor soluții propuse în literatură [4]:

- viteza de convergență: reprezintă numărul de iterații necesar unui algoritm adaptiv care operează cu semnale provenind din procese aleatoare *staționare* pentru a ajunge “suficient de aproape” de (ideal, să coincidă cu) soluția corespunzătoare ecuațiilor Wiener-Hopf. Deși este de dorit ca viteza de convergență să fie cât mai mare, în realitate optimizarea acestui parametru se poate face doar în detrimentul altor caracteristici.

- *misadjustment*: vom vedea pe parcursul următorului paragraf că nu este întotdeauna posibil ca valorile finale ale setului de coeficienți ai filtrului să coincidă cu cele furnizate de soluția ecuațiilor Wiener-Hopf, notate \mathbf{w}_{opt} . Ca urmare, nici valoarea finală a funcției de eroare nu va fi optimă, ci va fi mai mare decât cea corespunzătoare utilizării setului \mathbf{w}_{opt} (acesta este unul dintre “prețurile” plătite pentru imposibilitatea practică de a preciza cu exactitate mărimile statistice de interes, adică matricea de autocorelație \mathbf{R} și vectorul \mathbf{p}). Parametrul denumit *misadjustment* măsoară (printr-o mărime relativă, exprimată în procente) tocmai această “îndepărtare” a erorii finale față de cea ideală.

- proprietăți numerice: în implementarea oricărui algoritm de prelucrare numerică de semnal sunt fundamentale aspectele legate de efectul utilizării unei rezoluții

finite în reprezentarea numerelor în formă binară. De interes în acest context sunt erorile de cuantizare, respectiv cele de trunchiere. Două sunt aspectele care trebuie avute în vedere în cazul particular al algoritmilor adaptivi: *stabilitatea numerică*, care se referă la faptul că setul de coeficienți ai filtrului converge către valori finite chiar în condițiile efectelor datorate procesului de cuantizare, respectiv *acuratețea numerică*, care indică rezoluția necesară (numărul de biți) în reprezentarea datelor de lucru, respectiv a setului de coeficienți ai filtrului.

- arhitectura sistemului digital: se referă de asemenea la aspecte vizând implementarea *hardware* a diversilor algoritmi de procesare numerică de semnal, fiind atractive soluțiile care presupun un grad înalt de paralelism, de modularitate și un flux sporit de date prelucrate.

- complexitatea algoritmului: operațiile tipice implicate în definirea unor tehnici de prelucrare numerică de semnal sunt adunarea și înmulțirea. Timpul necesar efectuării unei singure iterații a unui algoritm adaptiv exprimă complexitatea sa și va fi influențat direct de numărul acestor operații aritmetice, de frecvența apelurilor de memorie și nu în ultimul rând de efortul de programare solicitat.

Aplicații specifice

Filtrele adaptive au fost aplicate cu succes în numeroase domenii, printre care transmisiuni de date, prelucrarea semnalelor radar, seismologie, inginerie biomedicală. Se disting patru clase principale de aplicații, ale căror scheme-bloc se prezintă în Fig. 3.4, abordate cu succes folosind atât filtre adaptive liniare cât și versiuni neliniare (în toate aceste categorii de aplicații rezultatele obținute cu metodele neliniare sunt de regulă mai performante, însă “prețul” plătit se referă la dificultățile sporite de a asigura stabilitatea sistemelor, volumul mai mare de calcul și posibilitatea de cantonare în soluții suboptimale). Diferențele dintre acestea se referă în special la natura semnalului cu rol de răspuns dorit la ieșirea filtrului. De asemenea, menționăm din nou că abordarea folosită predominant este de natură discretă, însă au fost raportate și aplicații bazate pe folosirea unor filtre adaptive analogice.

• **identificare de sistem:** rolul filtrului adaptiv este de a furniza *un model* al unui sistem necunoscut. În acest gen de aplicații atât filtrul adaptiv cât și sistemul necunoscut primesc la intrare același semnal, iar diferența dintre ieșirile acestora

definește semnalul de eroare care este folosit pentru modificarea recursivă a coeficienților filtrului. Există mai multe elemente care trebuie avute în vedere în acest gen de aplicații, indiferent dacă metoda utilizată are sau nu caracter adaptiv, dintre care menționăm: achiziționarea datelor de intrare-ieșire într-o manieră care să limiteze (sau măcar să permită estimarea cât mai precisă pentru) nivelul de zgomot suprapus peste semnalele utile, alegerea unui model adecvat și a algoritmului de estimare a valorilor parametrilor acestuia, utilizarea drept intrare a unui semnal capabil să excite toate frecvențele naturale ale sistemului necunoscut (de regulă, de tip zgomot alb), folosirea unei metode de validare a calității modelului obținut. Această direcție de cercetare este foarte bine acoperită în literatură și beneficiază de o paletă extrem de largă de tehnici, nu neapărat adaptive. Testarea validității modelului obținut este fundamentală, fiind disponibile criteriile statistice riguroase în special în cazul modelelor liniare.

- **modelare inversă (deconvoluție):** în acest caz rolul filtrului adaptiv este de furniza un model invers pentru un sistem necunoscut, în general însoțit de zgomot. În cazul unor sisteme liniare, modelul căutat are o funcție de transfer egală cu inversa funcției de transfer a sistemului necunoscut. Semnalul dorit este dat de versiunea, în general întârziată, a semnalului de intrare în sistem. Un aspect fundamental este legat de asigurarea stabilității modelului invers obținut, un exemplu în acest sens fiind oferit de aplicațiile în care sistemul original este un sistem discret liniar și invariant în timp, care nu este *de fază minimă* (zerourile funcției de transfer sunt plasate în afara doemniului de stabilitate al filtrului). Au fost elaborate tehnici speciale care permit asigurarea stabilității unor astfel de filtre adaptive, de exemplu prin readucerea “forțată” a singularităților sistemului în domeniul de stabilitate ca etapă premergătoare actualizării propriu-zise a coeficienților filtrului considerat. Un exemplu practic de aplicație inclusă în această categorie este egalizarea adaptivă a canalelor de transmisiuni de date.

- **predicție:** rolul filtrului adaptiv este de a aproxima cât mai bine valoarea unui semnal la un moment dat pe baza unui număr finit de valori *anterioare* ale acestuia. Ideea fundamentală care justifică atingerea unui asemenea obiectiv constă în supoziția că valorile succesive ale semnalului analizat respectă în mod obiectiv o dependență

funcțională (în cazul cel mai simplu, liniară) bazată pe un număr limitat de parametri, ale căror valori pot fi estimate folosind un algoritm adaptiv adecvat. În cazul liniar, modelele considerate se aleg de regulă dintre următoarele 3 variante: autoregresiv (AR), cu medie alunecătoare (MA), respectiv combinația acestora (ARMA) [7]. Numărul de parametri care descriu modelul (și care definesc *ordinul* acestuia) se estimează folosind criterii statistice consacrate. În unele situații informația de ieșire este dependentă nu numai de valorile anterioare ale semnalului analizat, ci și ale altor semnale. În plus, natura acestei dependențe poate varia în timp, sistemul adaptiv fiind forțat să asigure pe de o parte convergența rapidă a valorilor parametrilor și pe de altă parte urmărirea modificărilor apărute în procesul fizic analizat. Exemple practice de aplicații sunt tehnica LPC (*Linear Predictive Coding*) utilizată în prelucrarea semnalelor vocale, metoda ADPCM (*Adaptive Differential Pulse Code Modulation*) folosită în transmisiuni de date, predicția seriilor financiare.

• **filtrare de zgomot:** spre deosebire de cazurile anterioare, în acest gen de aplicații apar 2 intrări. Intrarea primară este constituită dintr-un semnal util peste care este suprapus zgomot. La cea de a doua intrare se aplică numai un semnal de tip zgomot prelevat dintr-un punct foarte apropiat sursei de semnal primar, astfel încât acesta să fie puternic corelat cu cel prezent în semnalul primar. Rolul filtrului adaptiv este de a furniza la ieșire un semnal cât mai apropiat de componenta de zgomot prezentă în semnalul primar, astfel încât prin scădere să obținem un semnal mai “curat”. Exemple concrete sunt oferite de aplicațiile de tip ANC (*Active Noise Control*) folosite pentru diminuarea nivelelor de zgomot în instalații industriale (spre exemplu, în sistemele de ventilație sau țesătorii), dar și în spații închise de mici dimensiuni (habitaclul autoturismului, căști audio), eliminarea ecourilor pe liniile de comunicații (*echo cancelling*), îmbunătățirea calității recepției în medii afectate de nivele mari de zgomot (cabine de tancuri sau elicoptere). În unele situații componenta utilă din semnalul primar poate fi de amplitudine mult mai mică decât cea datorată zgomotului.

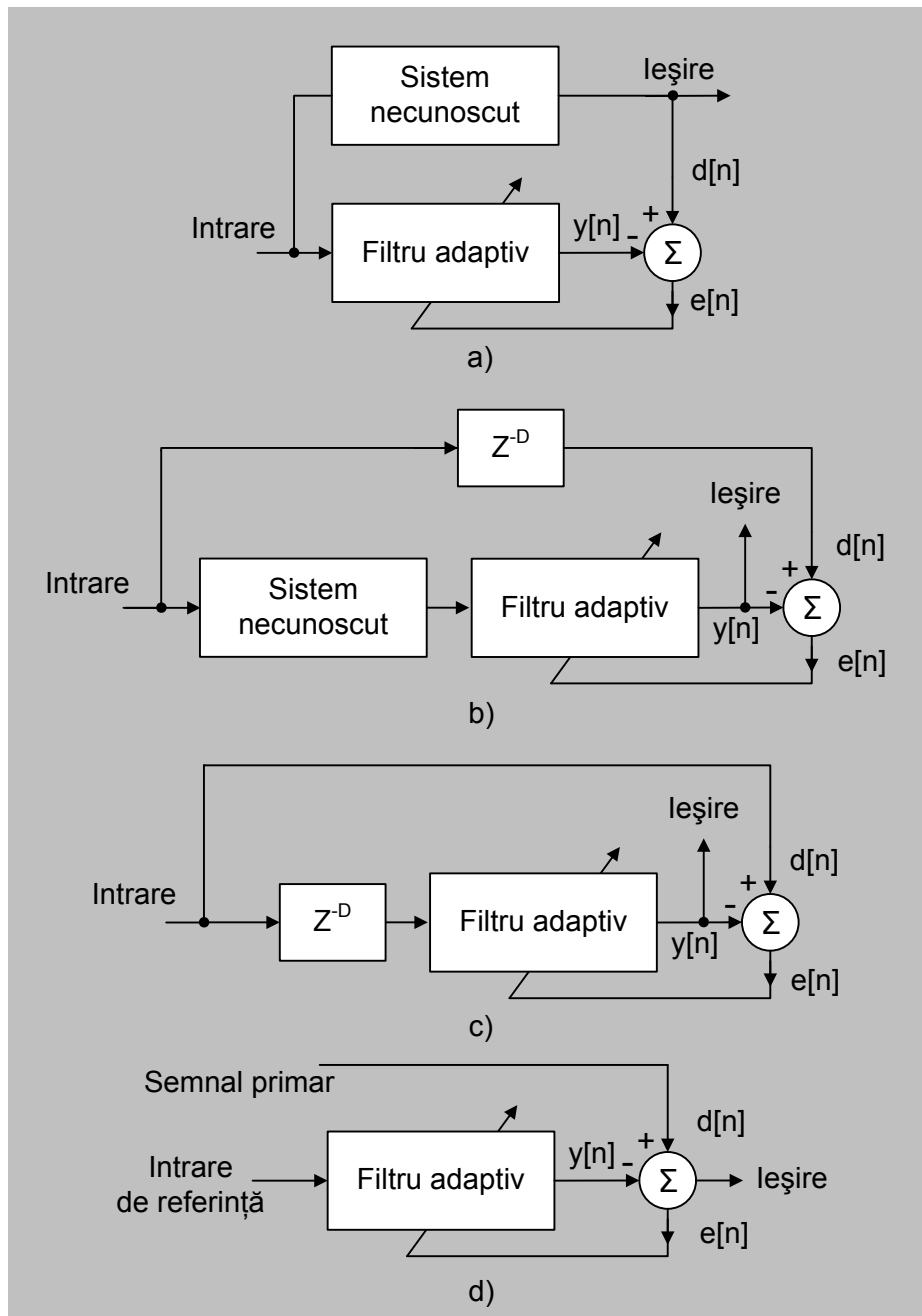


Fig. 3.4 Aplicații ale filtrelor adaptive:

a) identificare de sistem; b) filtrare inversă; c) predicție; d) filtrarea zgomotului

3.2 Algoritmi de filtrare adaptivă

Este important de subliniat încă de la început că o aplicație de filtrare adaptivă nu admite o soluție unică. În fapt, avem la dispoziție un întreg arsenal de tehnici diferite, fiecare având avantaje și dezavantaje specifice, iar alegerea uneia sau alteia dintre variantele posibile trebuie să ia în considerare criterii precum viteza de convergență, volumul de calcul și de memorie necesar, ori efectul apariției erorilor specifice implementării algoritmului folosind circuite care oferă precizie limitată. În cele ce urmează vom trece în revistă câteva exemple de algoritmi adaptivi reprezentativi, anume algoritmul LMS (*Least Mean-Squares*), algoritmul RLS (*Recurrent Least-Squares*) și filtrul Kalman. Acești algoritmi au fost elaborați în contextul utilizării unor filtre liniare, însă cu modificări specifice se regăsesc și în cazul filtrelor neliniare, în particular al rețelelor neurale. Aspectele teoretice fundamentale sunt prezentate în binecunoscute lucrări de referință [4, 9], astfel încât ne vom rezuma la a introduce terminologia corespunzătoare, a defini expresiile matematice proprii fiecărui algoritm și de a sublinia o serie de elemente utile în aplicațiile practice.

3.2.1 Algoritmul LMS

După cum s-a prezentat în paragraful anterior, una dintre abordările cele mai des utilizate în teoria filtrării adaptive este cea bazată pe formularea unei astfel de aplicații sub forma unei *probleme de optimizare*. În mod concret, se definește un criteriu de performanță sub forma unei *funcționale* care asociază fiecărui vector de coeficienți ai filtrului o valoare scalară care depinde și de proprietățile statistice ale semnalelor de la intrarea și ieșirea dorită a acestuia. Aspectul geometric al **suprafeței multidimensionale** corespunzătoare *funcției de eroare* rezultate poate fi foarte complicat, însă în cazul particular al unui filtru discret liniar de tip FIR și al erorii pătratice medii aceasta se prezintă ca o suprafață parabolică de formă convexă, având o valoare minimă unică. Ținând cont de relațiile (3.9) și (3.11) expresia funcției de eroare se poate scrie sub forma:

$$J(\mathbf{w}) = \mathbf{w}^H \mathbf{R} \mathbf{w} - \mathbf{p}^H \mathbf{w} - \mathbf{w}^H \mathbf{p} + \sigma_d^2 = J_{\min} + (\mathbf{w} - \mathbf{w}_{opt})^H \mathbf{R} (\mathbf{w} - \mathbf{w}_{opt}) \quad (3.14)$$

unde J_{\min} este valoarea minimă a erorii descrisă prin relația (3.13).

Se observă imediat că vectorul de coeficienți corespunzător acestei valori minime este chiar setul optim definit de ecuațiile Wiener-Hopf! În principiu, rezolvarea acestui sistem de ecuații s-ar putea efectua “dintr-un singur foc” printr-un calcul pur algebric indicat în ecuația (3.9), însă în multe situații practice apar dificultăți datorate dimensiunilor mari ale matricilor implicate sau probleme de stabilitate ale metodelor numerice folosite în inversarea matricii de autocorelație \mathbf{R} .

Determinarea valorilor extreme ale unei funcții de mai multe variabile poate fi asigurată printr-o paletă largă de metode, expuse cu acuratețe în textele referitoare la tehnicile de optimizare. Una dintre cele mai des folosite soluții o reprezintă **scăderea după gradient** (*gradient descent*), care în esență se bazează pe modificarea succesivă a variabilelor pe direcția și în sens invers gradientului funcției supuse procesului de optimizare. În cazul particular al erorii pătratice medii expresia gradientului se poate scrie compact sub forma [4]:

$$\nabla_w J = \left(\frac{\partial J}{\partial w_i} \right) = -2\mathbf{p} + 2\mathbf{R}\mathbf{w} \quad (3.15)$$

Modul de operare al algoritmului este descris prin relația următoare și prezentat detaliat în Tabelul 3.1 (constanta pozitivă η este denumită *constantă de adaptare*):

$$\mathbf{w}[n+1] = \mathbf{w}[n] - \frac{1}{2}\eta \nabla_w J = \mathbf{w}[n] + \eta(\mathbf{p} - \mathbf{R}\mathbf{w}[n]) \quad (3.16)$$

Analiza teoretică a algoritmului de tip scădere după gradient permite evidențierea următoarelor aspecte [4, 9]:

- stabilitatea algoritmului (convergența către valorile optime) este asigurată pentru o gamă de valori ale constantei de adaptare cupinsă în intervalul $0 < \eta < 2/\lambda_{\max}$, unde λ_{\max} desemnează valoarea proprie maximă a matricii de autocorelație \mathbf{R} .
- viteza de convergență a componentelor vectorului \mathbf{w} către valorile optime \mathbf{w}_{opt} poate fi apreciată cu ajutorul unor *constante de timp* cuprinse în intervalul:

$$\frac{-1}{\ln(1-\eta\lambda_{\max})} \leq \tau \leq \frac{-1}{\ln(1-\eta\lambda_{\min})} \quad (3.17)$$

- evoluția în decursul procesului de convergență a erorii pătratice medii este descrisă de relația:

$$J[n] = J_{\min} + \sum_{k=0}^{M-1} \lambda_k (1 - \mu \lambda_k)^{2n} |v_k[0]|^2 \quad (3.18)$$

în care vectorul $\mathbf{v}[n]$ se definește prin relația $\mathbf{v}[n] = \mathbf{Q}^H (\mathbf{w}_{opt} - \mathbf{w}[n])$ (matricea \mathbf{Q} are drept coloane *vectorii proprii* ai matricii de autocorelație \mathbf{R}).

Este ușor de observat că relația recursivă care exprimă modul de operare al algoritmului include matricile \mathbf{R} și \mathbf{p} . În multe situații practice, datorită imposibilității de a caracteriza corect din punct de vedere statistic procesele aleatoare implicate, aceste mărimi nu sunt disponibile, astfel încât suntem nevoiți să utilizăm valori *estimate* ale acestora. Cea mai simplă metodă de estimare constă în înlocuirea valorilor exacte ale matricilor menționate anterior – calculate prin acțiunea operatorului de mediere $E\{\cdot\}$ pe un ansamblu de realizări individuale ale proceselor aleatoare considerate – prin valorile lor *instantanee*. Procedeu stă la baza formulării algoritmului denumit *Least Mean-Squares* (LMS) și este prezentat în Tabelul 3.2.

Tabelul 3.1: Algoritmul de tip scădere după gradient (*gradient descent*)

Variabile și parametri:	
Vectorul de coeficienți la iterația n:	$\mathbf{w}[n] = \{w_0[n] \ w_1[n] \ w_2[n] \ \dots \ w_{M-1}[n]\}^T$
Vectorul de intrare la iterația n:	$\mathbf{u}[n] = \{u[n] \ u[n-1] \ \dots \ u[n-M+1]\}^T$
Răspunsul dorit la iterația n:	$d[n]$
Matricea de autocorelație a intrării:	$\mathbf{R} = E\{\mathbf{u}[n]\mathbf{u}^H[n]\}$
Vectorul de intercorelație intrare-ieșire:	$\mathbf{p} = E\{\mathbf{u}[n]d^*[n]\}$
Gradientul funcției de eroare la iterația n:	$\nabla J_w[n] = -2\mathbf{p} + 2\mathbf{R}\mathbf{w}[n]$
Constanta de adaptare:	$0 \leq \eta \leq 1$
Inițializare:	
Valoarea inițială a coeficienților:	$\mathbf{w}[0]$ (tipic = $\mathbf{0}$)
Calcul iterativ: n = 1, 2, ...	
Se adaptează valorile coeficienților:	$\mathbf{w}[n+1] = \mathbf{w}[n] + \eta\{\mathbf{p} - \mathbf{R}\mathbf{w}[n]\}$

Tabelul 3.2: Algoritmul *Least-Mean Squares* (LMS)

Variabile și parametri:	
Vectorul de coeficienți la iterația n:	$\mathbf{w}[n] = \{w_0[n] \ w_1[n] \ w_2[n] \ \dots \ w_{M-1}[n]\}^T$
Vectorul de intrare la iterația n:	$\mathbf{u}[n] = \{u[n] \ u[n-1] \ \dots \ u[n-M+1]\}^T$
Răspunsul dorit la iterația n:	$d[n]$
Răspunsul filtrului la iterația n:	$y[n] = \mathbf{w}^H[n]\mathbf{u}[n]$
Eroarea de estimare la iterația n:	$e[n] = d[n] - y[n]$
Valoarea estimată a matricii de autocorelație a intrării:	$\hat{\mathbf{R}}[n] = \mathbf{u}[n]\mathbf{u}^H[n]$
Valoarea estimată a vectorului de intercorelație intrare-ieșire dorită:	$\hat{\mathbf{p}}[n] = \mathbf{u}[n]d^*[n]$
Valoarea estimată a gradientului funcției de eroare la iterația n:	$\hat{\nabla} J_w[n] = -2\mathbf{u}[n]d^*[n] + 2\mathbf{u}[n]\mathbf{u}^H[n]\mathbf{w}[n]$
Constanta de adaptare:	$0 \leq \eta \leq 1$
Inițializare:	
Valoarea inițială a coeficienților:	$\mathbf{w}[0]$ (tipic = $\mathbf{0}$)
Calcul iterativ: n = 1, 2, ...	
Se adaptează valorile coeficienților:	$\mathbf{w}[n+1] = \mathbf{w}[n] + \eta\mathbf{u}[n]e^*[n]$

Avantajul principal al acestui algoritm constă în simplitatea sa, însă sunt necesare o serie de observații:

- spre deosebire de algoritmul de scădere după gradient, la care modificarea setului de coeficienți se face *determinist*, strict pe direcția gradientului funcției de eroare, în cazul algoritmului LMS apare un **zgomot de gradient** (datorat lucrului cu valorile *estimate* și nu cele exacte ale acestei mărimi), care imprimă modificărilor un caracter aleator - nu întotdeauna pe direcția gradientului! - deși **în medie** evoluția urmărește această direcție.
- din același motiv, valorile finale ale setului de coeficienți ai filtrului nu se mai stabilizează la \mathbf{w}_{opt} (soluția ecuațiilor Wiener-Hopf), ci **fluctuează permanent** în mod aleator în jurul acestora. Ca urmare, avem de-a face și cu o modificare a noțiunii de convergență a algoritmului. Astfel, se consideră următoarele 2 definiții:

$$\text{Convergență în valoare medie:} \quad E\{\mathbf{w}[n]\} \rightarrow \mathbf{w}_{opt} \quad (3.19)$$

$$\text{Convergență în valoare pătratică medie:} \quad J[n] \rightarrow J(\infty), J(\infty) > J_{min} \quad (3.20)$$

Pentru a asigura convergența conform acestor definiții, constanta de adaptare trebuie să satisfacă următoarele condiții [4]:

$$\begin{aligned} 0 < \eta < \frac{2}{\lambda_{max}} \\ \sum_{i=1}^M \frac{\eta \lambda_i}{2 - \eta \lambda_i} < 1 \end{aligned} \quad (3.21)$$

Se poate arăta că relațiile anterioare conduc la formularea unei condiții ușor de utilizat în practică, anume că valoarea constantă de adaptare η trebuie aleasă invers proporțional cu puterea semnalului aplicat la intrare [4].

- există un “preț” care se plătește pentru simplificarea majoră datorată estimării valorilor matricilor \mathbf{R} și \mathbf{p} : valoarea finală a erorii pătratice medii, notată $J(\infty)$, va fi mai mare decât cea corespunzătoare folosirii setului de coeficienți optim \mathbf{w}_{opt} :

$$J(\infty) = \frac{J_{min}}{1 - \sum_{i=1}^M \frac{\eta \lambda_i}{2 - \eta \lambda_i}} \quad (3.22)$$

Parametrul denumit *misadjustment* exprimă cantitativ această abatere (de regulă, în valori procentuale):

$$MA = \frac{J(\infty) - J_{min}}{J_{min}} = \frac{\sum_{i=1}^M \frac{\eta \lambda_i}{2 - \eta \lambda_i}}{1 - \sum_{i=1}^M \frac{\eta \lambda_i}{2 - \eta \lambda_i}} \approx \frac{\eta}{2} \sum_{i=1}^M \lambda_i = \frac{\eta M \lambda_{mediu}}{2} \quad (3.23)$$

Constanta de timp asociată procesului de convergență a algoritmului este:

$$\tau \approx \frac{1}{2\eta \lambda_{mediu}}, \quad \text{unde } \lambda_{mediu} = \frac{1}{M} \sum_{i=1}^M \lambda_i \quad (3.24)$$

Au fost elaborate și o serie de variante la algoritmul LMS standard, menite să corecteze unele dintre dezavantajele acestuia, printre care viteza redusă de

convergență, apariția unor probleme de natură numerică la implementarea *hard* a algoritmului datorate preciziei finite de lucru (în acest caz se manifestă efectul erorilor de cuantizare), zgomot de gradient apreciabil în cazul lucrului cu semnale de intrare cu gamă dinamică mare (în această situație variația valorilor coeficienților filtrului de la o iterație la alta este prea mare). În Tabelul 3.3 se prezintă câteva dintre variantele utilizate des în aplicații, iar mai jos se indică funcția MATLAB care implementează varianta standard a algoritmului.

Tabelul 3.3: Variante ale algoritmului LMS

LMS cu moment:	$\Delta \mathbf{w}[n] = \eta \mathbf{u}[n] e^*[n] + \alpha \Delta \mathbf{w}[n-1]$
LMS normalizat:	$\Delta \mathbf{w}[n] = \mathbf{w}[n+1] - \mathbf{w}[n] = \frac{\eta \mathbf{u}[n] e^*[n]}{\ \mathbf{u}[n]\ }$
LMS “cu pierderi” (<i>leaky-LMS</i>):	$\mathbf{w}[n+1] = (1 - \eta \alpha) \mathbf{w}[n] + \eta \mathbf{u}[n] e^*[n]$
LMS “cu semn”	$\Delta \mathbf{w}[n] = \eta \mathbf{u}[n] \text{sign}\{e[n]\}$

lms.m: Funcție MATLAB care implementează algoritmul LMS

```
function [W, MSE] = lms(u, d, mu, K, M);
% u - semnal de intrare
% d - semnal de iesire dorit
% mu - constanta de adaptare
% K - numar iteratii
% M - ordinul filtrului adaptiv
% W - coeficientii filtrului adaptiv
% MSE - eroarea patratica medie

MSE = zeros(1,K); % alocare memorie pentru salvarea erorii experimentului curent
W = zeros(M,1); % valori initiale ale coeficientilor filtrului
X = zeros(M,1); % initializarea ferestrei de date aplicate la intrarea filtrului

for k=1:K % bucla indexata de numarul de iteratii pentru un experiment
    X = [u(k); X(1:M-1)]; % vector de intrare la iteratia curenta
    y = W'*X; % iesirea filtrului la iteratia curenta
    e = d(k)-y; % eroarea de estimare la iteratia curenta
    W = W+2*mu*e*X; % actualizarea valorii coeficientilor
    MSE(k)=e^2; % eroarea patratica instantanee
end
```

3.2.2 Algoritmul Recurrent Least-Squares (RLS)

Problema de filtrare liniară optimală poate fi abordată și dintr-o perspectivă complementară celei specifice filtrului Wiener, renunțând la punctul de vedere statistic, bazat pe considerarea unui ansamblu de realizări individuale ale unor procese aleatoare cu rol de intrare și ieșire dorită a filtrului și înlocuindu-l cu o abordare *temporală*. În mod concret, acțiunea operatorului statistic de mediere $E\{\cdot\}$ este înlocuită cu simpla mediere aritmetică a valorilor unor realizări particulare *unice* ale proceselor aleatoare menționate anterior.

Ca și în paragrafele precedente vom considera în continuare un filtru discret cu răspuns finit la impuls de ordin M , având setul de coeficienți $\mathbf{w} = [w_0 \ w_1 \ \dots \ w_{M-1}]^T$, a cărui ieșire la un moment dat se poate calcula pe baza relației:

$$y[n] = \sum_{k=0}^{M-1} w^*[k] u[n-k] \quad (3.25)$$

Funcția de cost utilizată pentru determinarea coeficienților filtrului este dată de suma valorilor pătratice ale erorilor de estimare instantanee:

$$J(\mathbf{w}) = \sum_{n=n_1}^{n_2} |e[n]|^2, \text{ unde } e[n] = d[n] - y[n] \quad (3.26)$$

Valorile n_1 și n_2 desemnează limitele intervalului de timp pe care efectuează minimizarea. Urmând un mod de calcul oarecum similar celui care conduce la formularea ecuațiilor Wiener-Hopf se poate ajunge la un sistem de ecuații asemănător, cunoscut sub denumirea de **ecuații normale** [4, 9], cu particularitatea că matricea de autocorelație \mathbf{R} corespunzătoare intrării, precum și vectorul de intercorelație intrare-ieșire \mathbf{p} sunt înlocuite de variantele calculate prin mediere temporală, conform relațiilor următoare:

<i>Funcția de autocorelație temporală:</i>	$\phi[t, k] = \sum_{n=M}^N u[n-k] u^*[n-t], 0 \leq t, k \leq M-1 \quad (3.27)$
<i>Funcția de intercorelație intrare-ieșire dorită:</i>	$\theta[-k] = \sum_{i=M}^N u[n-k] d^*[n], 0 \leq k \leq M-1 \quad (3.28)$
<i>Sistemul de ecuații normale:</i>	$\mathbf{\Phi} \mathbf{w} = \mathbf{\Theta} \quad (3.29)$

În ultima relație intervin următoarele mărimi:

<i>Matricea de autocorelație temporală:</i>	$\Phi = \begin{bmatrix} \phi[0,0] & \phi[1,0] & \cdots & \phi[M-1,0] \\ \phi[0,1] & \phi[1,1] & \cdots & \phi[M-1,1] \\ \vdots & \vdots & \ddots & \vdots \\ \phi[0,M-1] & \phi[1,M-1] & \cdots & \phi[M-1,M-1] \end{bmatrix} \quad (3.30)$
<i>Vectorul de intercorelație temporală:</i>	$\Theta = \{\theta[0] \ \theta[-1] \ \cdots \ \theta[-M+1]\}^T \quad (3.31)$

Valorile optime ale setului de ponderi ale filtrului rezultă sub forma:

$$\mathbf{w}_{opt} = \Phi^{-1}\Theta \quad (3.32)$$

Rezolvarea ecuației matriciale de mai sus nu este, de regulă, o chestiune simplă. În particular, merită observat faptul că matricea de autocorelație temporală Φ nu este în general pătratică, astfel încât pentru calculul matricii Φ^{-1} trebuie să apelăm la ceea ce se numește *pseudoinversă* [11]. Un instrument elegant și eficient pentru calculul inversei unei matrici cu dimensiuni oarecare este oferit de metoda *Singular Value Decomposition* (SVD) [11]. O soluție alternativă este oferită de posibilitatea de a calcula valorile optime ale setului de coeficienți \mathbf{w} într-o manieră recursivă, astfel încât la apariția fiecărei noi perechi de date intrare-ieșire dorită să putem actualiza vechile rezultate. În acest scop, matricile care intervin în formularea setului de ecuații normale se rescriu sub forma:

$$\begin{aligned} \Phi[n] &= \sum_{i=1}^n \lambda^{n-i} \mathbf{u}[i] \mathbf{u}^H[i] = \lambda \Phi[n-1] + \mathbf{u}[n] \mathbf{u}^H[n] \\ \Theta[n] &= \sum_{i=1}^n \lambda^{n-i} \mathbf{u}[i] d^*[i] = \lambda \Theta[n-1] + \mathbf{u}[n] d^*[n] \end{aligned} \quad (3.33)$$

unde parametrul $0 < \lambda \leq 1$, denumit factor de “uitare” (*forgetting factor*), exprimă proprietatea potrivit căreia în evaluarea acestor matrici valorile cele mai recente au o pondere mai importantă decât cele vechi. Pasul următor îl reprezintă utilizarea unui rezultat algebric celebru denumit “lema de inversiune a matricilor” [4], care oferă posibilitatea de a calcula inversa matricii de autocorelație temporală $\Phi[n]$ pe baza unei relații de recurență. Se ajunge astfel la formularea algoritmului adaptiv

denumit *Recursive Least-Squares* (RLS), a cărei expresie matematică este prezentată în Tabelul 3.4, iar în continuare funcția MATLAB care indică implementarea algoritmului. În relațiile din tabel semnalul $\alpha[n]$ desemnează eroarea de estimare corespunzătoare utilizării “vechilor” valori ale setului de coeficienți ai filtrului (anterioare etapei de actualizare de la momentul n , motiv pentru care această mărime este denumită eroare *a priori*), iar parametrul δ reprezintă o valoare scalară pozitivă mică, utilizată în inițializarea matricii $\mathbf{P}[n] = \Phi^{-1}[n]$. Algoritmul RLS asigură o viteză de convergență mult mai mare decât algoritmul LMS, însă este mult mai laborios. În plus, după cum se va arăta în paragraful următor, acest algoritm este sensibil la aspectele de natură numerică datorate utilizării unei rezoluții finite în implementarea *hardware*.

Tabelul 3.4 Algoritmul RLS

Variabile și parametri:	
Vectorul de coeficienți la iterația n :	$\mathbf{w}[n] = \{w_0[n] w_1[n] w_2[n] \dots w_{M-1}[n]\}^T$
Vectorul de intrare la iterația n :	$\mathbf{u}[n] = \{u[n] u[n-1] \dots u[n-M+1]\}^T$
Răspunsul dorit la iterația n :	$d[n]$
Eroarea de estimare <i>a priori</i> la iterația n :	$\alpha[n] = d[n] - \mathbf{w}^H[n-1]\mathbf{u}[n]$
Vectorul de câștig (<i>gain</i>) la iterația n :	$\mathbf{k}[n]$
Inversa matricii de autocorelație temporală la iterația n :	$\mathbf{P}[n] = \Phi^{-1}[n]$
Inițializare:	
Valoarea inițială a matricii \mathbf{P} :	$\mathbf{P}[0] = \delta^{-1}\mathbf{I}$
Valoarea inițială a vectorului de coeficienți:	$\mathbf{w}[0]$ (tipic = $\mathbf{0}$)
Calcul iterativ: $n = 1, 2, \dots$	
Se adaptează valorile parametrilor de interes:	$\mathbf{k}[n] = \frac{\lambda^{-1}\mathbf{P}[n-1]\mathbf{u}[n]}{1 + \lambda^{-1}\mathbf{u}^H[n]\mathbf{P}[n-1]\mathbf{u}[n]}$ $\alpha[n] = d[n] - \mathbf{W}^H[n-1]\mathbf{u}[n]$ $\mathbf{W}[n] = \mathbf{W}[n-1] + \mathbf{k}[n]\alpha^*[n]$ $\mathbf{P}[n] = \lambda^{-1}\mathbf{P}[n-1] - \lambda^{-1}\mathbf{k}[n]\mathbf{u}^H[n]\mathbf{P}[n-1]$

rls.m: Funcție MATLAB care implementează algoritmul RLS

```
function [W] = rls(u, d, lambda, delta, K, M);
% u           - semnal de intrare
% d           - semnal de iesire dorit
% lambda, delta - constante din Tabelul 3.4
% K           - numar iteratii
% M           - ordinul filtrului adaptiv
% W           - coeficientii filtrului adaptiv

W = zeros(M, 1); % initializarea coeficientilor filtrului
P = eye(d)/delta; % initalizarea matricii P
X=zeros(M,1);    % initializarea ferestrei de date aplicate la intrarea filtrului

for k = 1:K
    X = [u(k); X(1:M-1)];
    kappa = lambda + u'*P*u;
    k_gain = P*u / kappa;
    alpha = d(k) - W'*X;
    W = W + k_gain * alpha;
    Pp = k_gain * u' * P;
    P = (P - Pp) / lambda;
end
```

3.2.3 Filtrul Kalman

Problema de filtrare liniară optimală a fost abordată pe parcursul paragrafelor anterioare din perspectiva teoriei filtrului Wiener, bazată pe minimizarea unei funcții de cost dependente de eroarea de estimare, calculată la rândul ei pornind de la ecuația care exprimă legătura intrare-ieșire a filtrului considerat. În plus, algoritmul RLS expus anterior a introdus principiul elegant al calculului iterativ al mărimilor de interes, cu avantaje majore asupra performanțelor algoritmului. Un alt punct de vedere în soluționarea problemei studiate este oferit de așa-numitul filtru Kalman, care păstrează ideea calculului recursiv, însă introduce descrierea sistemului adaptiv analizat prin *ecuații de stare*. O asemenea abordare permite extinderea comodă a teoriei la aplicații cu intrări/ieșiri multiple și, mai important, oferă cadrul de operare cu semnale reprezentând manifestări ale unor procese aleatoare *nestaționare*.

În cele ce urmează ne vom referi la varianta discretă a acestui algoritm, cu mențiunea că există analize teoretice și ale variantei analogice, mai puțin întâlnită în aplicații. Astfel, să considerăm vectorul notat $\mathbf{x}[n]$ ca reprezentând un set de M

parametri ai unui sistem dinamic liniar ce definesc starea sistemului la momentul de timp n . Sistemul furnizează un semnal de ieșire accesibil măsurătorii pentru un interval de timp cu durata de N eșantioane și poate fi caracterizat prin ecuațiile [4]:

$$\begin{aligned} \mathbf{x}[n+1] &= \mathbf{\Phi}[n+1, n] \mathbf{x}[n] + \mathbf{v}_1[n] \\ \mathbf{y}[n] &= \mathbf{C}[n] \mathbf{x}[n] + \mathbf{v}_2[n] \end{aligned} \quad (3.34)$$

în care $\mathbf{\Phi}[n+1, n]$ desemnează așa-numita *matrice de tranziție a stărilor*, $\mathbf{v}_1[n]$ este denumit *zgomot de proces* și este modelat ca un proces aleator de tip zgomot alb cu valoare medie nulă și matrice de autocorelație $\mathbf{Q}_1[n]$, $\mathbf{C}[n]$ este o matrice de dimensiune $N \times M$, iar $\mathbf{v}_2[n]$ este zgomotul de măsură, considerat de asemenea cu valoare medie nulă și matrice de autocorelație $\mathbf{Q}_2[n]$. Matricile $\mathbf{\Phi}[n+1, n]$ și $\mathbf{C}[n]$ se consideră cunoscute, iar procesele aleatoare $\mathbf{v}_1[n]$ și $\mathbf{v}_2[n]$ sunt statistic independente.

Să presupunem acum că dispunem de o *valoare precisă* a vectorului de stare la un moment dat n , notată cu $\hat{\mathbf{x}}[n]^-$, bazată pe totalitatea informației disponibile *înaintea* momentului n . În plus, se cunoaște și matricea de covarianță a erorii de estimare corespunzătoare momentului n (presupusă cu valoare medie nulă):

$$\mathbf{P}[n]^- = E\{\mathbf{e}[n]^- \mathbf{e}[n]^{-H}\} = E\{(\mathbf{x}[n] - \hat{\mathbf{x}}[n]^-)(\mathbf{x}[n] - \hat{\mathbf{x}}[n]^-)^H\} \quad (3.35)$$

Ideea de bază este de a îmbunătăți valoarea precisă $\hat{\mathbf{x}}[n]^-$ **folosind informația suplimentară adusă de valoarea măsurată la momentul n , anume $\mathbf{y}[n]$** . Valoarea *actualizată* a stării estimate la momentul n , notată $\hat{\mathbf{x}}[n]$, se alege sub forma unei combinații liniare între valoarea anterioară $\hat{\mathbf{x}}[n]^-$ și *eroarea de predicție* la momentul n :

$$\hat{\mathbf{x}}[n] = \hat{\mathbf{x}}[n]^- + \mathbf{G}[n](\mathbf{y}[n] - \mathbf{C}[n]\hat{\mathbf{x}}[n]^-) \quad (3.36)$$

unde factorul $\mathbf{G}[n]$ este denumit câștig Kalman (*Kalman gain*) și se determină în așa fel încât valoarea estimată $\hat{\mathbf{x}}[n]$ să fie optimă conform unui criteriu statistic precizat. În literatură este folosită eroarea pătratică medie drept criteriu de apreciere a optimalității soluției și în acest caz se demonstrează că valoarea optimă a factorului $\mathbf{G}[n]$ se poate scrie sub forma:

$$\mathbf{G}[n] = \Phi[n+1, n] \mathbf{P}[n]^{-1} \mathbf{C}[n]^H (\mathbf{C}[n] \mathbf{P}[n]^{-1} \mathbf{C}[n]^H + \mathbf{Q}_2[n])^{-1} \quad (3.37)$$

Un element de dificultate în ecuația anterioară este dat de necesitatea de a dispune de matricea de corelație $\mathbf{P}[n]^{-1}$. Aceasta se calculează în mod recursiv sub forma unei ecuații cu diferențe *de tip Ricatti*:

$$\mathbf{P}[n+1]^{-1} = \Phi[n+1, n] \mathbf{P}[n]^{-1} \Phi^H[n+1, n] + \mathbf{Q}_1[n] \quad (3.38)$$

în care matricea de corelație $\mathbf{P}[n]$ este descrisă de ecuația:

$$\mathbf{P}[n] = \mathbf{P}[n-1]^{-1} - \Phi[n, n+1] \mathbf{G}[n] \mathbf{C}[n] \mathbf{P}[n-1]^{-1} \quad (3.39)$$

Relațiile anterioare se aplică iterativ, permițând calcularea valorilor *prezise* ale mărimilor de interes la momentul $(n+1)$ pe baza informațiilor disponibile *pînă la momentul n inclusiv*, după cum se prezintă în Tabelul 3.5. Singurii parametri disponibili pentru a controla evoluția filtrului Kalman sunt valoarea inițială a matricii de covarianță a erorii de estimare $\mathbf{P}[0]$ și matricile de autocorelație $\mathbf{Q}_1[n]$ și $\mathbf{Q}_2[n]$. Alegerea unor valori mari ale acestora are efectul micșorării câștigului Kalman, operațiune oarecum asemănătoare micșorării constantei de adaptare din cazul algoritmilor de tip scădere după gradient.

Mecanismul de operare a filtrului Kalman poate fi reprezentat sugestiv prin schema-bloc din Fig. 3.5. Denumirea de “filtru” este oarecum nepotrivită în contextul evocat anterior, în realitate fiind vorba de *un algoritm de calcul* care urmărește minimizarea unei funcții de eroare și nu de interpretarea obișnuită de sistem cu răspuns în frecvență selectiv. Merită subliniată posibilitatea de a extinde acest algoritm și în cazul sistemelor neliniare, pe baza așa-numitului **filtru Kalman extins** (EKF), prin *liniarizarea ecuațiilor de stare în jurul punctului curent de funcționare*, înlocuind practic sistemul neliniar printr-unul liniar variabil în timp.

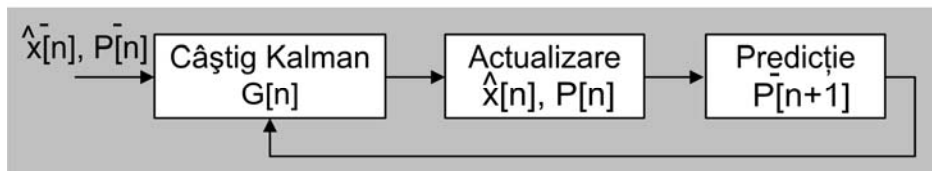


Fig. 3.5 Schema-bloc corespunzătoare filtrului Kalman

Tabelul 3.5: Filtrul Kalman

Variabile și parametri:	
Vector de stare la iterația n:	$\mathbf{x}[n]$
Matrice de tranziție a stării de la momentul n la iterația (n+1):	$\Phi[n+1,n]$
Vector de valori măsurate la iterația n:	$\mathbf{y}[n]$
Matrice corespunzătoare ecuației de măsură:	$\mathbf{C}[n]$
Matrice de autocorelație a zgomotului asociat procesului:	$\mathbf{Q}_1[n]$
Câștig Kalman la iterația n:	$\mathbf{G}[n]$
Eroare de predicție la iterația n:	$\boldsymbol{\alpha}[n]$
Matrice de autocorelație a zgomotului asociat ecuației de măsură:	$\mathbf{Q}_2[n]$
Inițializare:	
Stare inițială:	$\mathbf{X}[0]$
Valoare inițială a matricii de autocorelație a erorii de predicție:	$\mathbf{P}[0]$
Calcul iterativ: n = 1, 2, ...	
Se adaptează valorile parametrilor de interes:	$\mathbf{G}[n] = \Phi[n+1,n]\mathbf{P}[n]^{-1}\mathbf{C}[n]^H * (\mathbf{C}[n]\mathbf{P}[n]^{-1}\mathbf{C}[n]^H + \mathbf{Q}_2[n])^{-1}$ $\boldsymbol{\alpha}[n] = \mathbf{y}[n] - \mathbf{C}[n]\hat{\mathbf{x}}[n]^-$ $\hat{\mathbf{x}}[n] = \hat{\mathbf{x}}[n]^- + \mathbf{G}[n]\boldsymbol{\alpha}[n]$ $\mathbf{P}[n] = \mathbf{P}[n-1]^- - \Phi[n,n+1]\mathbf{G}[n]\mathbf{C}[n]\mathbf{P}[n-1]^-$ $\mathbf{P}[n+1]^- = \Phi[n+1,n]\mathbf{P}[n]\Phi^H[n+1,n] + \mathbf{Q}_1[n]$

3.3 Algoritmi adaptivi în domeniul frecvență

Analiza algoritmului LMS prezentată în paragraful anterior a arătat că valorile constantelor de timp asociate evoluției parametrilor unui filtru adaptiv pe parcursul procesului de convergență variază invers proporțional cu valorile proprii ale matricii de autocorelație a intrării \mathbf{R} . În consecință, pentru a obține o viteză mare de

kalman.m: Funcție MATLAB care implementează filtrul Kalman

```
function [x_new, P]=kalman_w(x, d, Phi, C, x_0, P_0, Q1, Q2, numIter);
% Semnificatia parametrilor de intrare este dată în Tabelul 3.5
% x_new: valoarea actualizata a vectorului de stare
% P: valoarea actualizata a matricii de corelație a erorii
% MSE: eroarea pătratică medie

Phat_old=P_0;
xhat_old=x_0;

for i=1:numIter
    S=C*Phat_old*C'+Q2;
    g=Phi*Phat_old*C'*inv(S);
    xhat_new=xhat_old+g*(d(i)-C*xhat_old);
    P=Phat_old-Phi*g*C*Phat_old;
    Phat_new=Phi*P*Phi'+Q1;
    xhat_old=xhat_new;
    Phat_old=Phat_new;
end

x_new=xhat_old;
P=Phat_old;
```

convergență este necesar ca valorile proprii să fie cât mai mari. Pe de altă parte, din considerente legate de stabilitatea algoritmului, valoarea proprie maximă limitează amplitudinea constantei de adaptare, care la rândul ei influențează performanțele de viteză ale sistemului. Rezultă din aceste observații că este de dorit ca “împrăștierea” acestor valori să fie cât mai mică, situația ideală corespunzând condiției ca *toate valorile proprii să fie identice*, caz în care matricea de autocorelație \mathbf{R} devine proporțională cu o matrice unitate. Ca urmare, variabilele de intrare ale filtrului adaptiv devin necorelate (ortogonale) și au puteri identice. Una dintre posibilitățile cele mai simple de a asigura atingerea acestui obiectiv îl constituie aplicarea unei transformări liniare adecvate, iar “candidatul” cel mai la îndemână este *Transformata Fourier Discretă* (DFT).

Schema-bloc a unui filtru adaptiv bazat pe utilizarea unei transformate liniare cu efect de decorelare (ortogonalizare) a intrărilor se prezintă în Fig. 3.6 [1]. Relațiile care descriu modul de operare al algoritmului LMS pentru un filtru care utilizează pentru ortogonalizare varianta DFT (denumit intuitiv filtru adaptiv în domeniul

frecvență) sunt prezentate în Tabelul 3.6, iar în continuare se indică funcția MATLAB corespunzătoare utilizării unui algoritm adaptiv de tip LMS cuplat cu *Transformata Cosinus Discretă*.

Observații:

- a) alegerea DFT este dictată în special de familiarizarea sporită a proiectanților cu acest instrument de analiză și nu de optimalitatea rezultatului. În fapt, se poate arăta că alte transformate liniare, precum *Transformata Cosinus Discretă* (DCT) sau *Transformata Wavelet Discretă* (DWT) asigură o reducere mai pronunțată a gamei dinamice a valorilor proprii ale matricii de autocorelație a intrării pentru clase largi de procese aleatoare staționare [1].
- b) asigurarea decorelării printr-una dintre transformatele enumerate anterior este independentă de natura particulară a semnalului prelucrat.

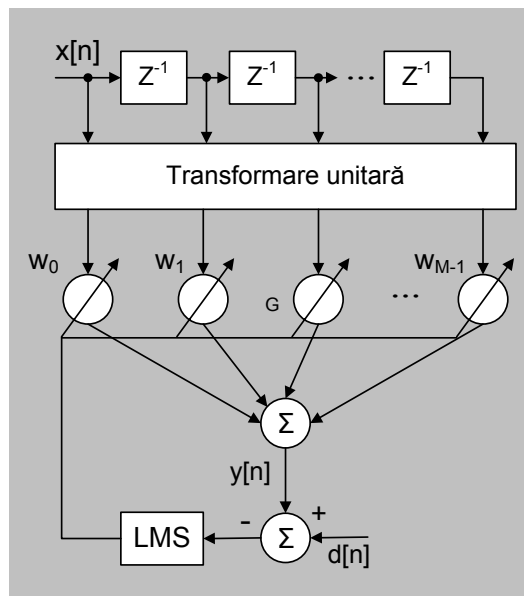


Fig. 3.6 Schema-bloc a unui filtru adaptiv în domeniul frecvență

Tabelul 3.6: Algoritmul LMS în domeniul frecvență

Variabile și parametri:	
Vectorul de coeficienți la iterația n:	$\mathbf{W}[n] = \{w_0[n] w_1[n] w_2[n] \dots w_{M-1}[n]\}^T$
Vectorul de intrare la iterația n:	$\mathbf{u}[n] = \{u[n] u[n-1] \dots u[n-M+1]\}^T$
Transformata FFT a intrării	$U[k,n] = \sum_{n=0}^{M-1} u[n] e^{-j \frac{2\pi nk}{M}}, k = 0 \dots M-1$
Normalizarea puterii*:	$U_{\text{norm}}[k,n] = \frac{U[k,n]}{\sqrt{P[k,n]}}, k = 0 \dots M-1$
Eroarea de estimare:	$e[k] = d[k] - y[k]$
Calcul iterativ: k = 1, 2, ...	
Adaptarea coeficienților:	$\mathbf{W}[k+1] = \mathbf{W}[k] + \eta U_{\text{norm}}[k] e^*[k]$

* Valorile individuale ale puterii pe fiecare componentă în parte se estimează pe baza relației: $P[k,n] = \alpha P[k,n-1] + (1-\alpha)u^2[k,n]$ [1].

O explicație intuitivă a motivului pentru care DFT reușește să asigure decorelarea informațiilor aplicate la intrare se poate obține făcând apel la interpretarea modului de operare al acestei transformate din perspectiva răspunsului în frecvență al filtrelor liniare. Astfel, considerînd un semnal discret $x[n]$ avînd un număr finit de eșantioane N , formula care definește DFT este:

$$\mathbf{X}[k] = \sum_{n=1}^N \mathbf{x}[n] e^{-j \frac{2\pi nk}{N}}, k = 1 \dots N \quad (3.40)$$

Un punct de vedere interesant este cel potrivit căruia oricare dintre eșantioanele $X[k]$ se obține ca rezultat al produsului de convoluție dintre semnalul $x[n]$ și un semnal de forma $h_k[n] = e^{j \frac{2\pi kn}{N}}$, cu **indicele k fixat** la una dintre valorile cuprinse între 1 și N . În aceste condiții semnalul $h_k[n]$ poate fi interpretat ca *funcția pondere a unui filtru discret* cu răspuns finit la impuls (FIR), avînd răspunsul în frecvență:

$$H_k(\omega) = \sum_{n=1}^N h_k[n] e^{-j\omega n} \quad (3.41)$$

dct_lms.m: Funcție MATLAB care implementează algoritmul DCT_LMS

```

function [W, MSE] = dct_lms(u, d, mu, alpha, K, M);
% u - semnal de intrare
% d - semnal de iesire dorit
% mu - constanta de adaptare
% alpha - constanta pentru estimarea puterii
% K - numar iteratii
% M - ordinul filtrului adaptiv
% W - coeficientii filtrului adaptiv
% MSE - eroarea patratica medie

MSE=zeros(1,K); % alocare memorie pentru salvarea erorii experimentului curent
W=zeros(M,1); % valori initiale ale coeficientilor filtrului
X=zeros(M,1);
TW=zeros(M,1);
sigma2=zeros(M,1);

% Matricea asociată transformatei DCT
for j=1:M
    T(1,j)=1/sqrt(M);
    for i=2:M
        T(i,j)=sqrt(2/M)*cos(pi*(i-1)*(2*j-1)/2/M);
    end
end

for k=1:K
    X=[u(k);X(1:N)];
    S=T*X; % transformata DCT aplicată intrării
    y=TW*S; % ieșirea (transformată) a filtrului
    sigma2=(1-alpha)*sigma2+alpha*S.*S; % estimarea puterii intrării (transformate)
    isigma2=1./(sigma2+gamma); % inversa puterii estimate
    e=d(k)-y; % eroare de estimare
    TW=TW+2*mu*isigma2.*(e*S); % actualizarea setului de coeficienți
    MSE(k)=e^2; % eroarea pătratică
end

```

În Fig. 3.7 este reprezentat grafic modulul acestei funcții, care corespunde unui filtru trece-bandă având frecvența centrală $2\pi k/N$. Rezultă de aici că obținerea tuturor celor N eșantioane care definesc DFT presupune de fapt trecerea semnalului de intrare printr-un ansamblu de filtre trece-bandă, ale căror frecvențe centrale sunt răspândite echidistant pe intervalul $[0, 2\pi]$. Interpretarea acestei observații în contextul discuției anterioare referitoare la decorelarea unui filtru adaptiv este imediată: dacă filtrele trece-bandă ar fi fost *ideale* (adică ar fi avut răspunsuri în

frecvență sub forma unor “ferestre” dreptunghiulare care nu se suprapun) atunci informațiile regăsite la ieșirea lor ar fi fost necorelate (intuitiv, “n-ar fi avut legătură” unele cu altele, extrăgând caracteristici distincte din semnalul comun de intrare). Cum însă răspunsurile în frecvență ale fiecăruia dintre filtre ocupă în realitate întreaga bandă de frecvență $[0, 2\pi]$, suprapunându-se parțial, rezultă că în realitate toate filtrele vor extrage caracteristici comune ale intrării, diferită fiind numai ponderea în care diversele caracteristici se regăsesc la ieșirea fiecărui filtru.

O explicație la fel de intuitivă se poate da și dintr-o perspectivă geometrică. Pentru aceasta, reamintim mai întâi că funcția de cost de tip eroare pătratică medie asociată unui filtru liniar adaptiv definește o hipersuprafață cu aspect elipsoid în sistemul de coordonate format de setul de coeficienți ai filtrului, ca în Fig. 3.8. Ținând cont de observația potrivit căreia o matrice \mathbf{Q} formată din coloane ortogonale și normă Euclidiană comună egală cu 1 – cum este cazul matricii corespunzătoare DFT obținute prin reunirea vectorilor $h_k[n]$ definiți anterior – este *unitară* (în sensul că $\mathbf{Q}^{-1} = \mathbf{Q}^H$) și de proprietatea că o transformare unitară permite rotația obiectului asupra căruia acționează fără să îi modifice **forma**, rezultă că aplicarea DFT conduce la situația reprezentată în Fig. 3.8 pentru cazul simplu al unui filtru cu numai 2 coeficienți, adică axele principale ale elipsei se apropie de axele de coordonate [1]. În același timp, devine evident și rolul etapei de normalizare a puterilor pe fiecare componentă a DFT, anume tendința de transformare a suprafeței cu aspect elipsoid într-una sferică (din punct de vedere matematic, efectul este de egalizare a valorilor proprii ale matricii de autocorelație a semnalului obținut în urma aplicării DFT).

Este important de subliniat că o analiză riguroasă a eficienței utilizării transformărilor ortogonale asupra vitezei de convergență a filtrelor adaptive liniare depinde de clasa semnalelor de intrare considerate. Un exemplu în acest sens este prezentat în [1] pentru așa-numitele *proces aleatoare de tip Markov* de ordinul I (obținute prin aplicarea unui semnal de tip zgomot alb la intrarea unui filtru trecejos de ordinul I), care indică reducerea semnificativă a împrăstierii valorilor proprii ale matricii de autocorelație a intrării folosind variantele DFT și DCT. Utilizarea *Transformatei Wavelet Discrete* (DWT) este abordată în [5].

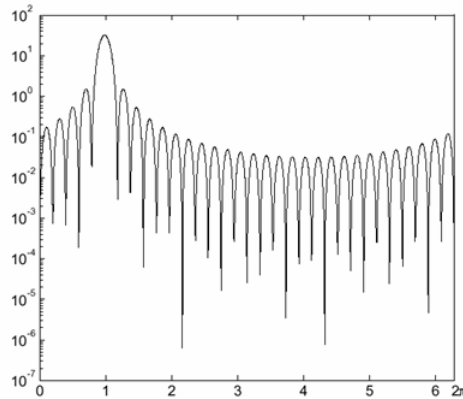


Fig. 3.7 Răspunsul în frecvență al unui filtru trece-bandă corespunzător FFT

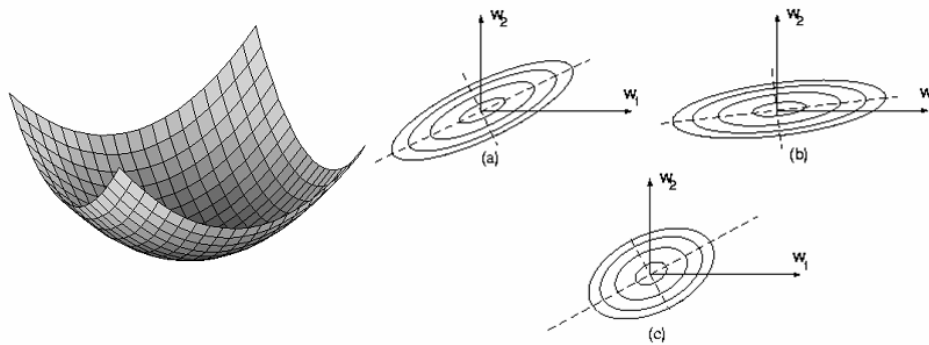


Fig. 3.8 Efectul transformării unitare și normalizării asupra funcției de eroare

3.4 Discuție asupra algoritmilor adaptivi

Există o serie de aspecte atât de ordin teoretic cât și practic care merită discutate în contextul utilizării algoritmilor adaptivi. Acestea vizează condiții, constrângeri, ipoteze și restricții care influențează în mod nemijlocit comportarea sistemelor bazate pe folosirea acestor instrumente de procesare a semnalelor și fixează cadrul adecvat funcționării corecte a aplicațiilor avute în vedere. În cele ce urmează vom trece în revistă o parte dintre aceste elemente, cu accent pe aspectele specifice algoritmilor LMS și RLS descriși în paragraful anterior, referitoare în special la performanțele acestora și la implementarea *hardware* folosind procesoare numerice de semnal.

Convergența algoritmului LMS

Utilizarea unor mărimi estimate în locul valorilor exacte ale parametrilor statistici implicați în formularea algoritmului de tip scădere după gradient are numeroase efecte asupra comportării algoritmului LMS, în particular asupra convergenței acestuia. Justificarea teoretică riguroasă a convergenței se face, de regulă, în condițiile așa-numitei *teorii a independenței* (statistice), care se bazează pe următoarele ipoteze simplificatoare [4]:

- “ferestrele” de timp succesive aplicate la intrarea filtrului, adică vectorii $\mathbf{u}[1]$, $\mathbf{u}[2]$, ... sunt considerați statistic independenți
- la momentul n , vectorul de intrare $\mathbf{u}[n]$ este statistic independent față de toate valorile anterioare ale semnalului dorit $d[1]$, $d[2]$, ..., $d[n-1]$
- la momentul n , semnalul dorit $d[n]$ este independent statistic față de toate valorile sale anterioare $d[1]$, $d[2]$, ..., $d[n-1]$
- la orice moment de timp, semnalul de intrare $\mathbf{u}[n]$ și cel dorit $d[n]$ respectă o distribuție normală (gaussiană).

Este interesant de menționat faptul că, în practică, aceste condiții nu sunt întotdeauna respectate cu exactitate (sau valabilitatea lor este dificil de verificat) și totuși algoritmul LMS funcționează cu bune rezultate. În particular, prima ipoteză simplificatoare este rareori valabilă, deoarece 2 ferestre de timp succesive vor avea în comun majoritatea eșantioanelor (dacă însă semnalele discrete aplicate la intrare nu reprezintă serii de *timp*, ci secvențe *spațiale* – de exemplu, ca în cazul antenelor adaptive – atunci condiția este mai ușor de îndeplinit). Există posibilitatea de a analiza convergența algoritmului LMS și fără a face apel la constrângerile anterioare, în condițiile în care constanta de adaptare are valori mici [4].

Un alt efect al lucrului cu valori estimate se manifestă în modalitatea de calcul a curbei care desemnează evoluția în timp a erorii pătratice medii $J_w(n)$ (așa-numita *curbă de învățare*). În cazul algoritmului de tip scădere după gradient expresia acestei mărimi este indicată în relația (3.14), în care intervine o operație de mediere prin intermediul acțiunii operatorului statistic $E\{\}$. Pe de altă parte, în cazul algoritmului LMS intervine zgomotul de gradient, astfel încât modificarea valorilor coeficienților filtrului nu se face întotdeauna strict în sensul descreșterii erorii.

Acest aspect impune ca, în practică, să fie necesare *experimente repetate* folosind aceeași valoare a constantei de adaptare și pornind din *aceleași valori inițiale* ale coeficienților filtrului, iar datele de intrare și de ieșire dorită să fie extrase ca realizări individuale ale unor acelorași procese aleatoare. Se obține astfel un **ansamblu de curbe de învățare zgomotoase**, care prin *mediere aritmetică* vor conduce la o curbă care, la limită, va corespunde cu evoluția funcției $J[n]$ din relația (3.18). Această concluzie este valabilă însă doar dacă valoarea constantei de adaptare utilizate este suficient de mică, în caz contrar apar diferențe între cele 2 funcții de eroare [6].

Efectul rezoluției finite

Studiul algoritmilor adaptivi, în particular analiza convergenței acestora, se face de regulă în condițiile în care valorile mărimilor implicate (semnale de intrare și de ieșire, coeficienți ai filtrului, rezultate intermediare) sunt considerate ca fiind numere reale, altfel spus precizia de lucru este considerată teoretic infinită. În practică însă, atunci când astfel de tehnici de prelucrare a semnalelor urmează a fi implementate folosind procesoare numerice de semnal (sau, după caz, microprocesoare de uz general) suntem nevoiți să folosim o *rezoluție finită*, cu alte cuvinte mărimile de interes vor fi reprezentate prin cuvinte binare având un număr finit de biți. Ca urmare, apar binecunoscutele erori de cuantizare, a căror identificare, modelare și tratare corectă vor avea efecte decisive asupra funcționării previzibile a algoritmului. Există două surse ale acestor erori: a) procesul de conversie analog-numerică a semnalelor prelucrate, în care aspectele esențiale vizează caracterul uniform sau neuniform al pasului de cuantizare, codul binar utilizat în reprezentarea datelor, precum și nivelul zgomotului de cuantizare generat; b) procesul de trunchiere/rotunjire a rezultatelor calculelor aritmetice, care conduce la erori ale căror proprietăți statistice sunt diferite de cele specifice conversiei analog-numerică (de exemplu, valoarea medie a erorilor de trunchiere/rotunjire poate fi nenulă). În cazurile cele mai neplăcute astfel de erori se pot acumula progresiv, conducând în final la saturarea valorilor mărimilor de interes. Această situație corespunde unei *instabilități numerice* a algoritmului considerat și, foarte important, *nu* poate fi evitată prin creșterea suplimentară a

rezoluției folosite. Pentru a minimiza efectele acestor erori, procesoarele de semnal moderne includ regiștri de lucru (în care se stochează rezultatele intermediare) de lungime mult mai mare decât rezoluția utilizată în reprezentarea rezultatelor finale (biții suplimentari se numesc *biți de gardă*).

Pe de altă parte, performanțele unui algoritm numeric stabil vor fi evident influențate de dimensiunea cuvintelor binare, iar parametrul denumit *acuratețe numerică* indică tocmai care este rezoluția necesară pentru ca dinamica și parametrii caracteristici ai algoritmului să se plaseze suficient de aproape de cei ai versiunii necuantizate. Din acest punct de vedere sunt cunoscute soluții constructive (de exemplu, arhitecturile de tip laticice) sau de sistem (algoritmul *leaky-LMS*) care oferă robustețe sporită în raport cu apariția acestor erori.

În final acestor considerații, să menționăm că procesul de cuantizare este în esență un *fenomen neliniar* și, ca urmare, analiza teoretică a convergenței, stabilității și dinamicii acestuia este mult mai dificilă decât în cazul versiunii liniare ideale.

În cazul algoritmului LMS, se demonstrează că setul de coeficienți ai filtrului reprezintă parametrii cei mai sensibili la apariția erorilor de cuantizare [4]. Se poate calcula o mărime de gen *misdjustment* care măsoară amplitudinea perturbației introduse prin considerarea valorilor cuantizate ale coeficienților în raport cu cele exacte, a cărei mărime, ca și în cazul altor caracteristici, va depinde atât de constanta de adaptare cât și de valorile proprii ale matricii de autocorelație a intrării. Una dintre cele mai des utilizate soluții de stabilizare a comportării acestui algoritm o reprezintă varianta *leaky-LMS*, care urmărește minimizarea simultană a erorii de estimare dar și a amplitudinii coeficienților filtrului. Funcția de cost folosită este în acest caz de forma:

$$J[n] = e^2[n] + \alpha \|\hat{\mathbf{w}}[n]\|^2 \quad (3.42)$$

unde α este un parametru pozitiv, iar minimizarea acesteia conduce la formularea următoarei expresii pentru modalitatea de actualizare a coeficienților:

$$\mathbf{w}[n+1] = (1 - \eta\alpha) \mathbf{w}[n] + \eta e[n] \mathbf{u}[n] \quad (3.43)$$

Excepție făcând de factorul care ponderează primul termen, relația anterioară este identică practic cu versiunea standard. Se poate arăta că includerea acestui factor

este echivalentă cu introducerea, peste semnalul de intrare original, a unui zgomot alb cu valoare medie nulă și dispersie α , denumit *dither* [4].

Există 2 tipuri de erori care pot fi întâlnite în cazul implementării *hardware* a algoritmului LMS [4, 9]:

a) fenomenul denumit *stalling* apare atunci când termenul de corecție a setului de coeficienți ai filtrului – calculat ținând cont de cuantizare! – are amplitudine prea mică (mai mică decât cel mai puțin semnificativ bit – LSB), adică $|\eta e[n] \mathbf{u}[n-i]| \leq \text{LSB}$. Implicația acestei observații este imediată: valoarea constantei de adaptare η nu poate fi aleasă oricât de mică, deși acest lucru ar fi fost de dorit din perspectiva minimizării zgomotului de gradient și a apropierei dinamicii de cea a algoritmului de tip scădere după gradient.

b) ”alunecarea parametrilor” (*parameter drift*) se referă la posibilitatea ca, în anumite situații, valorile coeficienților filtrului să tindă să crească oricât de mult (în practică valorile lor se saturează la nivelele maxime posibil de reprezentat folosind rezoluția aleasă), deși semnalele de intrare și eroarea de estimare rămân mărginite. Fără a intra în amănunte, să menționăm totuși că acest fenomen nedorit este cauzat de utilizarea drept intrări a unor clase de semnale particulare, iar constanta de timp specifică manifestării acestor erori este extrem de mare, astfel încât detectarea lor prin simulări tipice este greu de realizat (numărul de iterații necesar depășește cu mult valorile uzuale).

În cazul algoritmului RLS (ca, de altfel, și în cazul filtrului Kalman) se manifestă o instabilitate numerică generată de faptul că inversa matricii de autocorelație a erorii de estimare, notată $\mathbf{P}[n]$ în Tabelul 3.4 – care trebuie să fie o matrice pozitiv definită! – este calculată prin diferența a 2 matrici care *nu sunt* neapărat pozitiv definite (conform ultimei relații din tabel). O soluție care păstrează nealterat caracterul pozitiv-definit al matricii $\mathbf{P}[n]$ este indicată în [12], în care se calculează efectiv numai valorile matricii situate deasupra diagonalei principale, iar restul se completează prin simetrie. De asemenea, fenomenul de *stalling* amintit anterior poate apare și în cazul algoritmului RLS, mai ales dacă factorul “de uitare” λ folosit în actualizarea matricii $\mathbf{P}[n]$ este apropiat de valoarea 1. Soluția practică o

reprezintă calculul valorilor acestei matrici folosind un registru de acumulare cu o lungime mare, adică având un număr suficient de biți de gardă.

Filtre adaptive de tip IIR

Există aplicații practice care necesită utilizarea unor filtre FIR de ordin foarte mare. Întârzierea mare introdusă, ca și creșterea volumului de calcul sugerează posibilitatea folosirii ca alternativă a unui filtru adaptiv de tip IIR. Analiza teoretică a unor algoritmi adaptivi aplicați unor astfel de filtre a fost făcută în literatură în special în cazul aplicațiilor de tip identificare de sistem [10]. S-au utilizat cu precădere modele statistice de tip ARMA (autoregresiv cu medie alunecătoare), la care ieșirea filtrului este descrisă de o relație de forma:

$$y[n] = - \sum_{i=1}^N a[k] y[n-i] + \sum_{j=0}^M b[j] u[n-j] \quad (3.44)$$

unde $a[k]$ și $b[j]$ sunt coeficienți numere reale. Au fost formulate 2 metode de adaptare a valorilor acestor coeficienți [10]:

- metoda *output-error* aplică o procedură de tip scădere după gradient asupra unei funcții de cost de tip eroare pătratică medie. Dezavantajele acestei soluții sunt legate pe de o parte de instabilitatea potențială (nu putem garanta că, în decursul procesului de adaptare, polii filtrului vor rămâne întotdeauna în domeniul de stabilitate, astfel încât sunt necesare măsuri speciale de garantare a stabilității), iar pe de altă parte de faptul că funcția de cost nu mai prezintă o valoare minimă unică, așa cum se întâmpla în cazul filtrelor FIR, ci va avea o multitudine de valori minime locale, astfel încât inițializarea coeficienților va avea o importanță hotărâtoare în fixarea performanțelor acestuia.

- metoda *equation-error* elimină acest ultim neajuns, înlocuind în membrul drept al ecuației anterioare valorile propriu-zise ale ieșirii filtrului cu valorile întârziate ale *semnalului dorit*:

$$y[n] = - \sum_{i=1}^N a[k] y[n-i] + \sum_{j=0}^M b[j] d[n-j] \quad (3.45)$$

În acest fel însă filtrul obținut nu mai este identic cu cel inițial, motiv pentru care ecuația anterioară, însoțită de un algoritm de tip scădere după gradient pentru

determinarea coeficienților, este folosită în special în faza de început a procesului de adaptare, după care se revine practic la o funcționare de tip *output-error*.

O soluție superioară este furnizată de o clasă de filtre care “împrumută” caracteristicile optime ale ambelor tipuri de filtre discrete. Acestea respectă în continuare o arhitectură de tip FIR, însă înlocuiesc acțiunea operatorului clasic de întârziere prin operatori particulari, corespunzători unor filtre IIR. Un exemplu în acest sens îl reprezintă filtrele de tip Laguerre [9], a căror arhitectură se prezintă în Fig. 3.9. Funcțiile de transfer implicate sunt descrise prin relațiile:

$$L(z, \alpha) = \sqrt{1 - \alpha^2} \frac{z^{-1} - \alpha}{1 - \alpha z^{-1}} ; \quad L_0(z, \alpha) = \frac{\sqrt{1 - \alpha^2}}{1 - \alpha z^{-1}} \quad (3.46)$$

Se observă astfel că operatorul de întârziere z^{-1} este înlocuit printr-un filtru de tip trece-tot de ordinul I (excepție face funcția de transfer $L_0(z, \alpha)$, care corespunde unui filtru trece-jos), al cărui unic pol este plasat în domeniul de stabilitate dacă este îndeplinită condiția $|\alpha| < 1$.

Un alt caz particular îl reprezintă așa-numitele filtre gamma [8], la care operatorul de întârziere este înlocuit prin funcția de transfer:

$$\Gamma(z) = \frac{\mu}{z - (1 - \mu)} \quad (3.47)$$

stabilitatea fiind asigurată pentru domeniul de valori $0 < \mu < 2$.

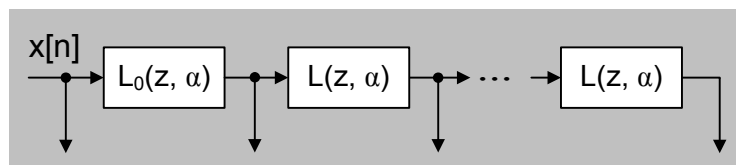


Fig. 3.9 Filtru de tip Laguerre

Analiza comparativă LMS-RLS

Deși nu au fost prezentate demonstrații riguroase ale convergenței celor 2 exemple importante de algoritmi adaptivi descriși în cuprinsul acestui paragraf, merită să menționăm totuși o serie de concluzii raportate în literatură, cu implicații semnificative în utilizarea practică a acestora:

- evoluția în decursul procesului de adaptare a erorii (pătratică) de estimare este descrisă de regulă prin așa-numita “curbă de învățare” (*learning curve*), care se trasează separat pentru fiecare experiment în parte, urmând ca prin mediere aritmetică să se obțină o caracteristică apropiată de valoarea teoretică a erorii pătratică medii (experimentele individuale au în comun același set de valori inițiale ale coeficienților filtrului adaptiv, precum și caracteristicile statistice ale semnalelor de intrare și ieșire dorită). Viteza de convergență a algoritmului RLS este mult mai mare decât a algoritmului LMS, tipic cu un ordin de mărime, însă volumul de calcul necesar pentru efectuarea unei singure iterații este de asemenea mult sporit.
- teoretic, algoritmul RLS poate converge către soluțiile optime corespunzătoare rezolvării setului de ecuații normale (3.32), cu alte cuvinte parametrul *misadjustment* se poate anula.
- definiția noțiunii de convergență în cazul ambilor algoritmi face distincția între convergența în valoare medie și cea în valoare pătratică medie. Spre deosebire de LMS, în cazul RLS aceste definiții nu sunt influențate de împrăștierea valorilor proprii ale matricii de autocorelație (temporală) a intrării.
- există numeroase aplicații practice importante în care un filtru adaptiv evoluează într-un mediu *nestaționar*, când proprietățile statistice ale semnalelor de intrare și de ieșire dorită *variază în timp*. În aceste condiții, algoritmului utilizat i se cere nu numai să asigure convergența coeficienților filtrului către valorile optime, dar să fie capabil și să urmărească modificarea permanentă a poziției acestor valori optime. Performanțele de urmărire (*tracking*) sunt puternic dependente de natura aplicației considerate și de particularitățile mărimilor statistice implicate. De multe ori algoritmul LMS are totuși o evoluție mai robustă decât RLS în astfel de situații, astfel încât acest argument, alături de simplitatea sporită, constituie motive pentru a-l prefera în aplicații.

Filtre adaptive neliniare

În cuprinsul acestui paragraf ne-am ocupat numai de clasa algoritmilor adaptivi utilizați pentru determinarea valorilor optime ale coeficienților unor filtre discrete liniare, capabili să rezolve probleme pentru care nu beneficiem de informații exacte

privind mărimile statistice de interes sau pentru care rezolvarea directă, de regulă prin metode algebrice, presupune un volum mare de calcul și sensibilitate la aspectele de natură numerică. Există însă și posibilitatea de a extinde aceste principii de operare și la sisteme cu caracter neliniar, exemplul cel mai elocvent fiind oferit de așa-numitele *rețele neurale artificiale*. Deși nu există o definiție general acceptată, majoritatea cercetătorilor sunt de acord că acestea reprezintă ansambluri de elemente de procesare simple, interconectate prin canale de comunicații prin care se propagă informație numerică. Din perspectivă istorică, multe dintre ideile vehiculate în acest context sunt motivate de dorința de a construi sisteme capabile să rezolve cu succes sarcini uzuale pentru creierul uman precum înțelegerea vorbirii sau recunoașterea formelor. În fapt, această abordare s-a dovedit utilă în special pentru probleme dificil de formalizat sub forma unui algoritm (adică a unei “rețete” care să garanteze rezultatul), situație care presupune o înțelegere profundă a aplicației considerate. Majoritatea rețelelor neurale utilizează mecanisme pe baza cărora intensitatea legăturilor dintre neuroni sunt ajustate în funcție de calitatea răspunsului la stimuli externi. Ajungem astfel la principala trăsătură a acestor sisteme, anume capacitatea de *a învăța pe bază de exemple*, folosind “experiența” anterioară pentru a-și îmbunătăți permanent performanțele, dar și de a oferi un anumit grad de generalizare, care se traduce printr-un răspuns adecvat la informații de intrare care nu au fost folosite în faza de “antrenare”.

Se pot identifica două direcții distincte înspre care este canalizată atenția cercetătorilor din domeniul rețelelor neurale. Prima o reprezintă identificarea unor modele plauzibile din punct de vedere biologic pentru neuronii elementari și structura de interconexiuni dintre aceștia. Interesul este justificat de preocupările pentru studierea creierelor naturale și de nivelul tehnologic actual, în speranța că într-o zi vom putea reproduce artificial performanțele remarcabile ale acestora. Cea de-a doua, pe care am putea-o denumi “inginerească”, își propune un scop mai puțin ambițios dar la fel de necesar, anume **identificarea unor principii de procesare suficient de simple și robuste, dependente de un număr relativ restrâns de parametri și care să poată fi folosite pentru rezolvarea unor probleme concrete.**

Gama aplicațiilor în care se utilizează rețelele neurale artificiale este extrem de vastă, extinzându-se mult în afara preocupărilor legate de tehnică în general și de electronică în particular. În ultimii ani au fost raportate rezultate foarte încurajatoare privind folosirea acestora în medicină, finanțe sau construcția de automobile și viitorul va demonstra cu siguranță înmulțirea și diversificarea acestor exemple. Această abordare s-a dovedit utilă și în cazul unor probleme "clasice", ca de exemplu conversia analog-numerică sau calculul de transformate liniare.

Rețelele neurale artificiale sunt caracterizate de 3 elemente: modelul adoptat pentru elementul de procesare individual (neuronul), structura particulară de interconexiuni (arhitectura) și mecanismul de ajustare a legăturilor dintre neuroni (algoritmul de învățare). Aspectul neliniar intervine în mod nemijlocit în descrierea neuronului elementar, pentru care așa-numitul *model aditiv*, reprezentat în Fig. 3.10, este de departe cea mai frecventă opțiune. Caracterul adaptiv se manifestă prin acțiunea algoritmului de învățare, pentru care au fost formulate numeroase clase de metode, unele dintre cele mai cunoscute fiind bazate, de exemplu, din nou pe mecanismul de scădere după gradient.

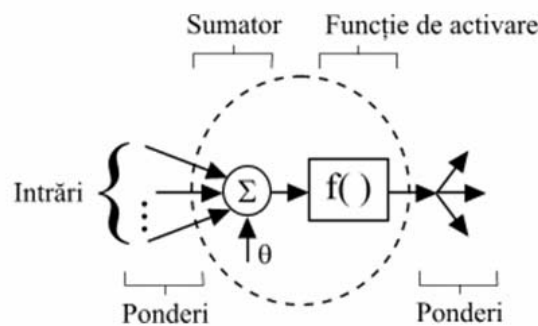


Fig. 3.10 Modelul aditiv pentru neuronul elementar

Tipul de rețea neurală care se apropie cel mai mult de categoria filtrelor liniare analizate anterior îl reprezintă cele denumite *Radial Basis Functions* (RBF) [3], a căror arhitectură se prezintă în Fig. 3.11.

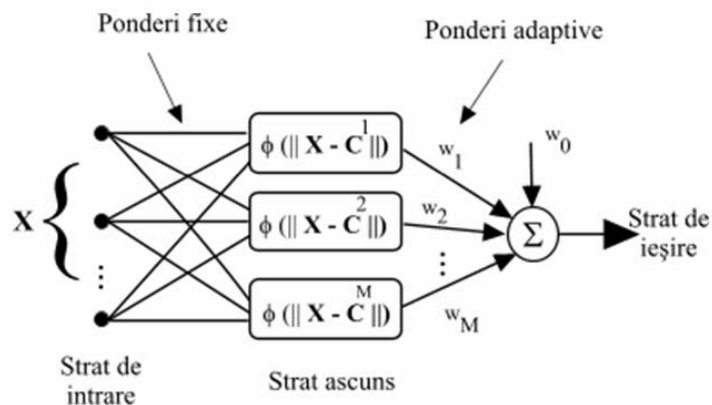


Fig. 3.11 Arhitectura de tip *Radial Basis Functions*

Relația care descrie legătura intrare-ieșire este formulată astfel:

$$f(\mathbf{X}) = w_0 + \sum_i w_i \phi_i(\|\mathbf{X} - \mathbf{C}^i\|) \quad (3.48)$$

în care: $\|\cdot\|$ reprezintă distanța (în general, Euclidiană) între doi vectori aparținând unui spațiu multidimensional, ϕ_i reprezintă funcții neliniare alese dintr-un set de funcții tipice, vectorii \mathbf{C}^i (având aceeași dimensiune cu cei de intrare) se numesc *centri*, iar w_i sunt coeficienți scalari. Mecanismul de selectare a centrilor este ghidat de principiul ca numărul și pozițiile acestora să ofere o aproximare cât mai fidelă a densității de repartiție a datelor de intrare. Odată alese acești parametri, valorile setului de coeficienți w_i rezultă în urma aplicării unui algoritm adaptiv din “arsenalul” celor specifici filtrelor liniare, de regulă chiar algoritmul LMS. De altfel, în raport cu semnalele de la ieșirea neuronilor având funcțiile de activare ϕ_i , ne putem imagina o rețea neurală de tip RBF ca fiind un simplu filtru FIR, cu aceleași avantaje privind viteza sporită de convergență și unicitatea soluției finale (se poate arăta că în cazul altor modele de rețele neurale, neliniaritatea funcției de activare implică un aspect puternic neregulat al erorii pătratice medii, caracterizat de prezența a numeroase valori minime locale). O posibilitate de generalizare a arhitecturii o reprezintă înlocuirea valorilor scalare ale setului de coeficienți w_i prin filtre discrete, de regulă de tip FIR.

3.5 Aplicații ale filtrelor adaptive

În cele ce urmează prezentăm o serie de exemple concrete de utilizare a algoritmilor adaptivi prezentați anterior în aplicații de procesare de semnal, ilustrând diferențele dintre performanțele acestora și influența diverșilor parametri specifici asupra vitezei de convergență, dinamicii și calității soluției finale obținute.

Predicție liniară

În acest tip de aplicații rolul filtrului adaptiv este de a aproxima cât mai bine valoarea unui semnal la un moment dat pe baza unui număr finit de valori *anterioare* ale acestuia. Ideea fundamentală care justifică atingerea unui asemenea obiectiv constă în supoziția că valorile succesive ale semnalului analizat respectă în mod obiectiv o dependență funcțională (în cazul cel mai simplu, liniară) dependentă de un număr limitat de parametri, ale căror valori pot fi estimate folosind un algoritm adaptiv adecvat. În cazul liniar, modelele considerate se aleg de regulă dintre următoarele 3 variante: autoregresiv (AR), cu medie alunecătoare (MA), respectiv combinația acestora (ARMA). Numărul de parametri care descriu modelul (și care definesc *ordinul* acestuia) se estimează folosind criterii statistice consacrate. În unele situații informația de ieșire este dependentă nu numai de valorile anterioare ale semnalului analizat, ci și ale altor semnale. În plus, natura acestei dependențe poate varia în timp, sistemul adaptiv fiind forțat să asigure pe de o parte convergența rapidă a valorilor parametrilor și pe de altă parte urmărirea modificărilor apărute în procesul fizic analizat. Exemple practice de aplicații sunt tehnica LPC (*Linear Predictive Coding*) utilizată în prelucrarea semnalelor vocale, metoda ADPCM (*Adaptive Differential Pulse Code Modulation*) folosită în transmisiuni de date, predicția seriilor financiare.

Ca exemplu, să considerăm un proces aleator de tip AR de ordinul 2, generat pe baza ecuației cu diferențe:

$$u[n] + \alpha u[n-1] + \beta u[n-2] = v[n] \quad (3.49)$$

unde $v[n]$ este un zgomot alb cu valoare medie nulă și dispersie σ_v^2 , iar α și β reprezintă parametri scalari reali, aleși astfel încât rădăcinile ecuației caracteristice asociate acestei ecuații cu diferențe să fie plasate în interiorul cercului de rază

unitate. În mod concret, să presupunem $\alpha = \beta = -0.5$, $\sigma_v^2 = 1$. Urmărim să determinăm valorile succesive ale semnalului $u[n]$ (care va juca acum rolul semnalului dorit $d[n]$ utilizat în formularea ecuațiilor Wiener-Hopf) folosind drept semnal de intrare vectorul bidimensional $\{u[n-1], u[n-2]\}$. În acest scop folosim algoritmul de tip scădere după gradient, iar rezultatele experimentale se prezintă în Fig. 3.12. Se pot trage următoarele concluzii: a) cu cât valoarea constantei de adaptare este mai mare cu atât crește viteza de convergență (către o aceeași valoare minimă a funcției de eroare!); b) cu cât împrăștierea valorilor proprii ale matricii de autocorelație a intrării este mai mare cu atât viteza de convergență scade; c) curbele de contur constant corespunzătoare erorii pătratice medii sunt de forma unor elipse (în figură este trasată câte o elipsă pentru fiecare actualizare a setului de coeficienți ai filtrului). Dacă se alege o valoare prea mare a constantei de adaptare atunci algoritmul nu mai converge.

Egalizarea canalelor de transmisiuni de date

Egalizarea canalelor de transmisiuni reprezintă o aplicație tipică de modelare inversă (deconvoluție), conform schemei-bloc din Fig. 3.3c. Presupunând ca sistemul necunoscut este liniar și are funcția de transfer $P(z)$, filtrul adaptiv ar trebui să realizeze în mod ideal funcția de transfer $H(z) = \frac{1}{P(z)}$. Din punct de vedere practic trebuie să ținem cont de următoarele aspecte:

- sistemul ce urmează a fi modelat introduce de regulă întârzieri, care trebuie compensate decalând în mod corespunzător semnalul dorit de la ieșirea filtrului adaptiv. Pentru filtrele FIR cu fază liniar variabilă se cunoaște că această întârziere este egală cu jumătate din ordinul filtrului.
- întotdeauna vor apărea zgomote suprapuse peste semnalele utile, acestea vor afecta în sens negativ determinarea modelului invers (rezultate satisfăcătoare se obțin dacă zgomotul este considerat alb, cu valoare medie nulă).
- riguros vorbind, un filtru FIR poate constitui modelul invers al unui sistem necunoscut doar dacă acesta are numai poli, nu și zerouri (altfel spus, un filtru IIR nu poate fi modelat perfect folosind un filtru FIR de ordin finit).

- semnalul de intrare trebuie să fie capabil să excite toate modurile (frecvențele naturale) corespunzătoare funcției de transfer a sistemului modelat, în caz contrar unele moduri nu vor fi observabile și deci nu vor putea fi modelate. Din acest motiv se preferă aplicarea la intrarea sistemului modelat a unui semnal cu spectru bogat, de tip zgomot alb.

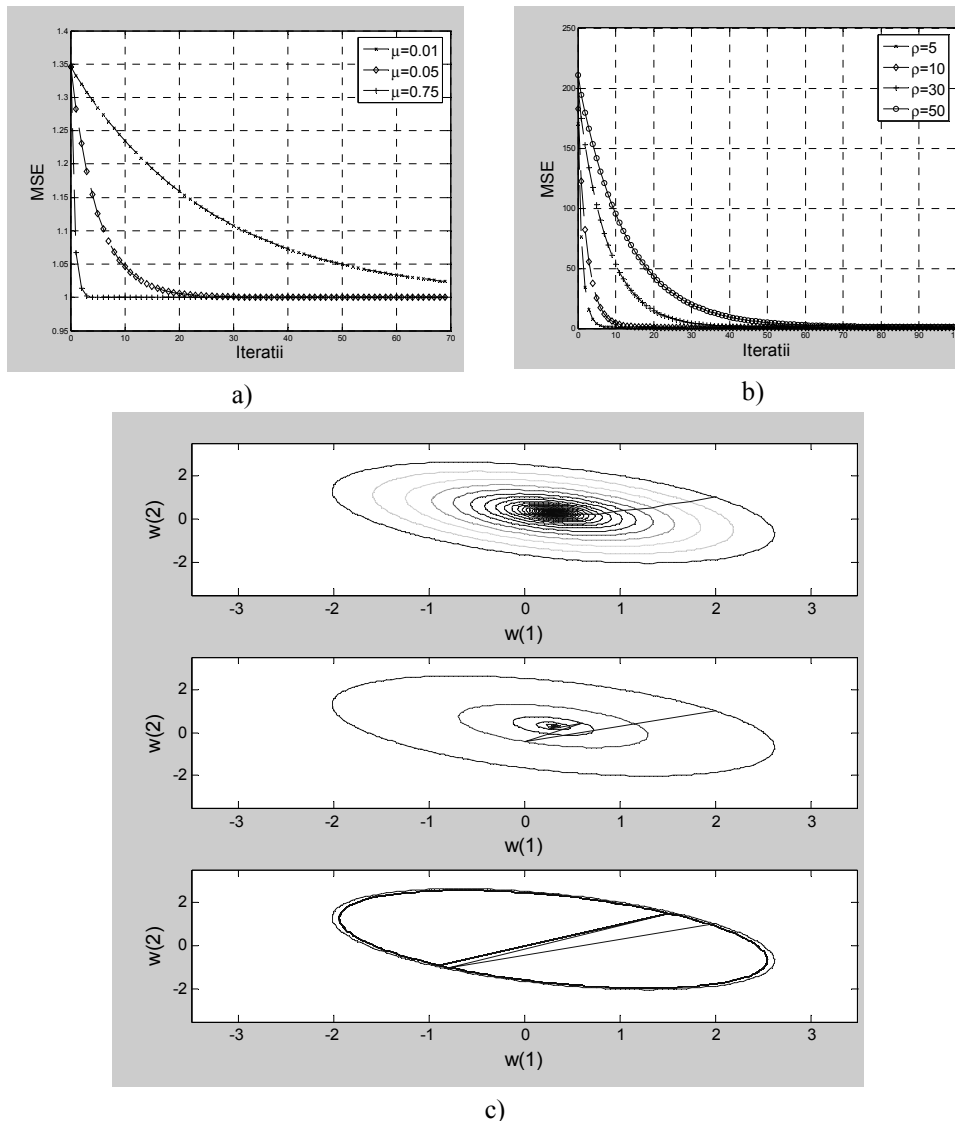
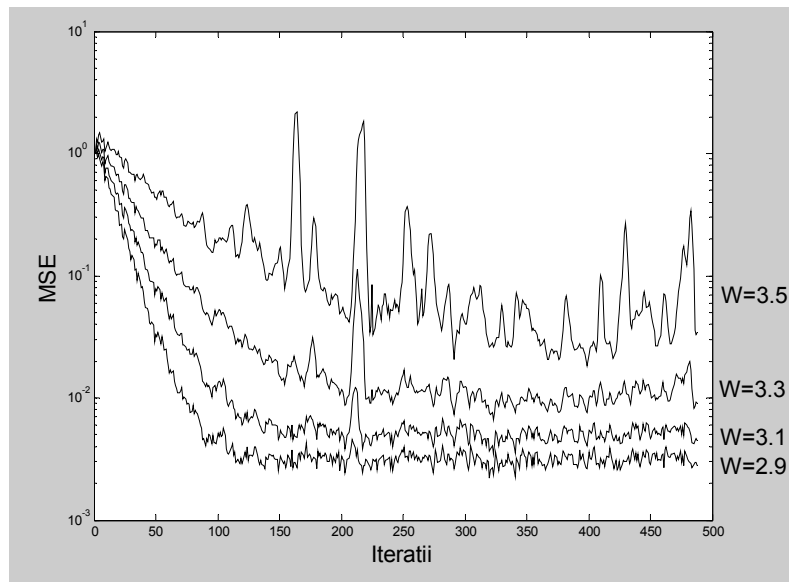
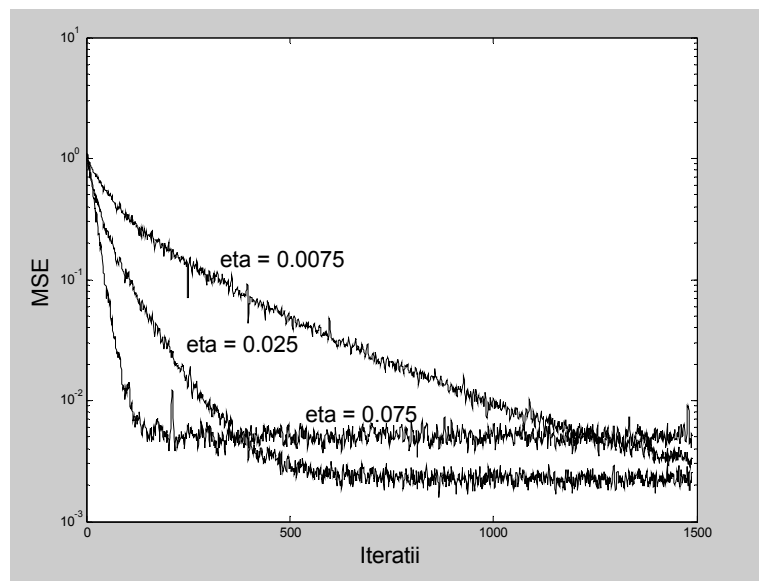


Fig. 3.12 Predicție liniară: a) evoluția erorii pătratice medii în funcție de constanta de adaptare; b) evoluția erorii pătratice medii în funcție de împrăștierea valorilor proprii ale matricii R ($\rho=\lambda_{\max}/\lambda_{\min}$); c) curbe de contur constant ale erorii pătratice medii și evoluția setului de coeficienți ai filtrului adaptiv



a)



b)

Fig. 3.14 Evoluția erorii pătratice medii: a) în raport cu împrăștierea valorilor proprii ale matricii de autocorelație a intrării; b) în raport cu valoarea constantei de adaptare

Bibliografie

- [1] Beaufays, F., "Orthogonalizing adaptive algorithms: RLS, DFT/LMS, and DCT/LMS", in *Adaptive Inverse Control*, NJ: Prentice Hall, 1995
- [2] Ciocoiu, I.B., *Rețele neurale artificiale*, Iași: Cantes, 2001
- [3] Haykin, S., *Neural Networks - A Comprehensive Foundation*, IEEE Press, 1994
- [4] Haykin, S., *Adaptive Filter Theory*, 4th ed., New Jersey: Prentice-Hall, 2002
- [5] Hosur, S., Tewfik, A.H., "Wavelet Transform Domain Adaptive FIR Filtering", *IEEE Trans. on Signal Processing*, vol. 45, no. 3, pp. 617-629, 1997
- [6] Nascimento, V.H., Sayed, A.H., "On the learning mechanism of adaptive filters", *IEEE Trans. on Signal Processing*, vol. 48, no. 6., pp. 1609-1625, 2000
- [7] Papoulis, A., *Probability, Random Variables, and Stochastic Processes*, New York: McGraw-Hill, 1984
- [8] Principe, J.C., B. de Vries, P.G. de Oliveira, "The Gamma Filter-A New Class of Adaptive IIR Filters with Restricted Feedback", *IEEE Trans. Sign. Process.*, vol. 41, 1993
- [9] Sayed, A.H., *Fundamentals of adaptive filtering*, New Jersey: Wiley, 2003
- [10] Shynk, J.J., "Adaptive IIR filtering", *IEEE ASSP Mag.*, pp. 4-21, 1989
- [11] Strang. G., *Introduction to linear algebra*, Wellesley: Cambridge Press, 1993
- [12] Yang, B., "A note on the error propagation analysis of recursive least squares algorithms", *IEEE Trans. on Signal Processing*, vol. 42, pp. 3523-3525, 1994