

**MINISTERUL EDUCAȚIEI NAȚIONALE
UNIVERSITATEA "POLITEHNICA" TIMIȘOARA
FACULTATEA DE ELECTRONICĂ ȘI TELECOMUNICAȚII**

**METODE ȘI ALGORITMI DE PRELUCRARE
ADAPTIVĂ A SEMNALELOR**

TEZĂ DE DOCTORAT

Conducător științific
Prof. dr. ing. CRIȘAN SEVER

624.493
181 Δ

Autor
Ing. PĂNOIU CAIUS

BIBLIOTECA CENTRALĂ
UNIVERSITATEA "POLITEHNICA"
TIMIȘOARA

-1998-

CUPRINS

Introducere.....	1
1. Conceptul de prelucrare adaptivă a semnalelor.....	3
1.1. Filtre liniare.....	3
1.2. Filtre neliniare.....	4
1.3. Filtre adaptive.....	4
1.4. Aplicații ale filtrelor adaptive.....	5
2. Filtre adaptive liniare.....	8
2.1. Eroarea medie pătratică.....	8
2.2. Calculul coeficienților filtrului Wiener.....	10
2.3. Metoda pantei descendente maxime.....	13
2.3.1. Evaluarea directă a ecuației diferențiale.....	16
2.3.2. Algoritmul gradientului stocastic (LMS).....	21
2.3.3. Algoritmul LMS cu pas variabil.....	25
2.4. Concluzii.....	32
3. Implementarea filtrelor adaptive IIR utilizând structuri lattice.....	34
3.1. Tipuri de structuri de filtre lattice fără reacție.....	34
3.1.1. Structura bazată pe algoritmul lui Durbin.....	34
3.1.2. Structura bazată pe metoda gradientului adaptiv.....	38
3.1.3. Structura bazată pe metoda gradientului adaptiv normalizat.....	41
3.1.4. Structura bazată pe utilizarea conceptului de spații vectoriale.....	42
3.1.5. Structura bazată pe reactualizarea coeficientului unghiular.....	46
3.2. Tipuri de structuri de filtre lattice cu reacție.....	50
3.2.1. Structura generală a celulei lattice cu reacție.....	50
3.2.2. Structura lattice feedback cu două multiplicări.....	51
3.2.3. Structura lattice Feedback normalizată.....	52
3.3. Filtre adaptive IIR.....	53
3.3.1. Arhitectura filtrelor adaptive IIR utilizând structuri lattice.....	53
3.3.2. Algoritmi de implementare a filtrelor adaptive IIR.....	56
3.3.2.1. Algoritm bazat pe utilizarea gradientului.....	56
3.3.2.2. Algoritmul Steiglitz-McBride.....	57
3.3.2.3. Algoritmul SHARF.....	58
3.4. Concluzii.....	59
4. Filtre adaptive neliniare.....	60
4.1. Algoritmi OSLMS.....	60
4.1.1. Influența perturbațiilor cu caracter negaussian asupra filtrării LMS.....	60
4.1.2. Principiul filtrării OSLMS.....	62

4.1.3. Tipuri de filtre OSLMS.....	62
4.1.3.1. Filtrul median-LMS (MLMS).....	62
4.1.3.2. Filtrul medie-LMS (ALMS).....	63
4.1.3.3. Algoritmul “mediei ordonate” LMS.....	64
4.1.4. Studiul parametrilor filtrelor OSLMS.....	65
4.1.5. Comutarea algoritmilor OSLMS în funcție de natura zgomotelor perturbatoare.....	67
4.1.6. Filtre adaptive MALMS.....	71
4.2. Filtre IIR-OSLMS.....	73
4.2.1. Definirea filtrelor IIR-OSLMS.....	73
4.2.2. Studiul caracteristicilor filtrelor IIR-OSLMS.....	74
4.3. Filtre combinaționale.....	75
4.3.1. Definirea filtrelor combinaționale.....	75
4.3.2. Filtre C cu parametrii de trunchiere variabili (VTC).....	83
4.3.3. Filtre C generalizate (GC).....	84
4.3.4. Implementarea filtrelor C adaptive.....	84
4.4. Filtre bazate pe criteriul erorii medii absolute.....	86
4.4.1. Filtre WOS.....	86
4.4.2. Filtre FIR-Median-Hibride (FMH).....	86
4.4.3. Filtre FIR-WOS-Hibride (FWH).....	87
4.4.4. Arhitectura și implementarea filtrelor WOS.....	88
4.4.5. Implementarea filtrelor adaptive rapide WOS și FMH.....	89
4.5. Concluzii.....	90
5. Reglarea adaptivă a temperaturii la cuptorul cu rezistență electrică.....	92
5.1. Metoda de identificare on-line a parametrilor proceselor lente cu timp mort....	92
5.2. Metoda adaptivă de identificare a parametrilor procesului.....	96
5.2.1. Aproximări ale modelului funcției de transfer a sistemului.....	96
5.2.2. Determinarea funcției sistem a sistemului discret.....	96
5.2.3. Studiul algoritmilor de identificare și a formelor de implementare.....	102
5.2.4. Validarea alegerii funcției sistem aproximativă.....	110
5.3. Reglatoare adaptive.....	112
5.3.1. Alegerea și acordarea reglatoarelor pentru procese lente.....	112
5.3.2. Implementarea reglatoarelor adaptive.....	115
5.4. Sistemul de reglare adaptivă al temperaturii la cuptorul cu rezistență electrică .	116
5.4.1. Structura sistemului de reglare al temperaturii.....	116
5.4.2. Caracteristicile plăcii de achiziție de date ADA-3100.....	117
5.4.3. Măsurarea parametrilor cuptorului cu rezistență electrică prin metoda de identificare on-line.....	119
5.4.4. Prezentarea metodei propuse de reglare adaptivă a temperaturii.....	120
5.5. Eliminarea zgomotelor cu caracter negaussian.....	121
5.6. Rezultate experimentale.....	122
5.7. Concluzii.....	129
6. Concluzii și contribuții originale.....	131
7. Bibliografie.....	135
8. Anexe.....	141

INTRODUCERE

Multe probleme de zi cu zi care apar în tehnica comunicațiilor precum și în prelucrarea semnalelor implică eliminarea zgomotelor și distorsiunilor ce pot apărea în procesele fizice variabile în timp sau a celor cu parametri necunoscuți. Aceste tipuri de procese reprezintă unele din cele mai dificile probleme de rezolvat în transmiterea și recepția informației.

Aria tehnicilor de prelucrare adaptivă a semnalelor furnizează atât metode deosebit de puternice în eliminarea zgomotelor și distorsiunilor în comunicații cât și în extragerea informațiilor referitoare la procesele fizice necunoscute.

Lucrarea de față se înscrie pe linia aprofundării științifice a tehnicilor de prelucrare adaptivă a semnalelor, propunându-și pe de o parte să realizeze un studiu asupra metodelor și algoritmilor adaptivi utilizați în prelucrarea semnalelor, iar pe de altă parte să prezinte o metodă originală de determinare, utilizând tehnicile adaptive, a parametrilor unui sistem necunoscut.

Teza a fost elaborată în intervalul 1992-1998 la Facultatea de Electronică și Telecomunicații a Universității "Politehnica" din Timișoara.

Lucrarea conține 6 capitole, are un număr de 189 de pagini, 78 de figuri și 17 tabele. Au fost utilizate 75 de referințe bibliografice dintre care 7 sunt lucrări la elaborarea cărora a participat și autorul.

Capitolul 1 are ca scop prezentarea conceptului de prelucrare adaptivă a semnalelor, prezentând caracteristicile filtrelor adaptive și clasele de aplicații ale acestora.

Capitolul 2 prezintă un studiu referitor la posibilitățile de implementare a filtrelor adaptive liniare. Pentru realizarea comparației între diferiți algoritmi autorul a simulat pe calculator comportarea unui canal de comunicație influențat de zgomot gaussian după care, cunoscând caracteristicile canalului și ale zgomotului, a calculat coeficienții optimi de predicție ai filtrului Wiener. Concluziile referitoare la performanțele fiecărui algoritm au putut fi desprinse prin medierea statistică a aceluiași număr de evenimente, fiecare eveniment conținând același număr de eșantioane.

Capitolul 3 prezintă un studiu referitor la performanțele filtrelor adaptive lattice. Performanțele structurilor lattice fără reacție au fost studiate în mod similar cu cel utilizat în capitolul precedent, autorul obținând concluzii importante referitoare la aceste structuri. Deoarece forma directă de implementare a filtrelor IIR prezintă dezavantaje în ceea ce privește stabilitatea, s-au arătat noi posibilități de implementare a filtrelor IIR utilizând structura lattice cu reacție, prezentând totodată și principalele structuri de filtre lattice cu reacție. În încheierea capitolului sunt prezentați trei algoritmi de identificare a parametrilor unui sistem necunoscut, algoritmi utilizați în situația în care funcția sistem a sistemului necunoscut poate fi aproximată de un filtru IIR cu structură lattice cu reacție.

Posibilitățile de eliminare a perturbațiilor care nu prezintă caracter gaussian utilizând filtre neliniare sunt prezentate în capitolul 4. În primul paragraf al acestui capitol sunt prezentate filtrele OSLMS, autorul studiind performanțele acestora în ceea ce privește eliminarea influențelor perturbațiilor cu caracter de impuls cu amplitudine mare și durată scurtă. Bazat pe concluziile obținute, în paragraful următor autorul a introdus o nouă categorie de filtre denumite IIR-OSLMS, definind relațiile de reactualizare a coeficienților și realizând un studiu referitor la performanțele acestor filtre. În încheierea capitolului se face o prezentare a altor tipurilor de filtre neliniare precum și a performanțelor acestora, fiind de asemenea prezentate posibilitățile de utilizare a filtrelor neliniare ca filtre neliniare adaptive.

În capitolul 5 este prezentată o metodă originală de a determina parametrii unui sistem cu o constantă de timp și timp mort utilizând un filtru adaptiv IIR al cărui numitor al funcției de transfer a fost implementat cu o structură lattice. Validarea rezultatelor experimentale a fost făcută prin comparație cu o metodă de identificare on-line a parametrilor proceselor cu timp mort.

Lucrarea se încheie cu capitolul 6 în care sunt prezentate concluziile și contribuțiile autorului la rezolvarea problemei abordate.

Pe întreaga perioadă de elaborare a lucrării, autorul a beneficiat de îndrumarea competentă și plină de înțelegere a conducătorului științific, domnul profesor doctor inginer Sever Crișan, căruia îi mulțumește din suflet.

Pentru încurajările și sfaturile primite pe parcursul redactării lucrării ca și pentru întreaga sa formație profesională autorul aduce cele mai sincere mulțumiri domnului profesor doctor inginer Liviu Toma.

Nu în ultimul rând, autorul mulțumește părinților, precum și în mod deosebit soției pentru sprijinul deosebit acordat pe toată perioada elaborării lucrării și fără de care lucrarea nu s-ar fi putut finaliza.

1. CONCEPTUL DE PRELUCRARE ADAPTIVĂ A SEMNALELOR

Subiectul (domeniul) filtrării adaptive a semnalelor s-a dezvoltat intens în ultima perioadă de timp, astfel că în prezent constituie o parte importantă a prelucrării statistice a semnalelor [1],[2],[12],[15]. Ori de câte ori este necesară o prelucrare a semnalelor rezultate dintr-un proces cu statistică necunoscută utilizarea unui filtru adaptiv oferă o soluție atractivă de rezolvare a problemei precum și o îmbunătățire substanțială a performanțelor comparativ cu situația utilizării unui filtru fix realizat prin metode convenționale. Mai mult decât atât, utilizarea unui filtru adaptiv furnizează noi posibilități de prelucrare a semnalelor care nu erau posibile altfel. Filtrele adaptive au fost aplicate până în prezent cu succes în diverse domenii ca de exemplu: comunicațiile, controlul proceselor, radar, seismologie, inginerie biomedicală etc. [15],[23],[26].

1.1. Filtre liniare

Termenul de filtrare este adesea utilizat pentru a descrie un dispozitiv ce poate fi o parte a unei structuri hardware și/sau un program software la care se aplică un set de date de intrare influențate de zgomot și furnizează la ieșire informații despre mărimea de interes. Zgomotul poate apărea de la o varietate de surse. Se poate utiliza un filtru pentru a realiza trei operații de bază de prelucrare a informației [2]:

1. **Filtrarea** - reprezintă extragerea informației referitoare la mărimea de interes la momentul t utilizând datele măsurate până la momentul de timp t inclusiv.
2. **Netezirea** - constă în obținerea de informații privitoare la valoarea mărimii de interes de la momentul de timp $t + \tau$, pentru $\tau > 0$, utilizând inclusiv datele măsurate după momentul de timp t în obținerea acestor informații. Aceasta înseamnă că în cazul netezirii există o întârziere în obținerea mărimii de interes dar, pe de altă parte, deoarece se pot utiliza și datele obținute ulterior momentului de timp t este de așteptat ca rezultatele obținute să prezinte o mai mare acuratețe decât în cazul procesului de filtrare.
3. **Predicția** - constă în obținerea de informații privitoare la valoarea mărimii de interes de la momentul de timp $t + \tau$, pentru $\tau > 0$, utilizând datele măsurate până la momentul de timp t .

Filtrul este liniar dacă filtrarea, netezirea sau predicția mărimii de interes este un operator liniar de datele aplicate la intrarea filtrului.

În determinarea soluției problemei filtrării liniare se utilizează anumiți parametri statistici. media și funcția de corelație, pentru a separa semnalul util de zgomotul nedorit. Este astfel necesară realizarea unui filtru liniar care să minimizeze efectul zgomotelor de la intrare. O aproximare utilă este de a minimiza valoarea mediei pătratice a semnalului de eroare definit ca diferența între răspunsul dorit și ieșirea filtrului.

Pentru semnale staționare de intrare soluția rezultantă este cunoscută ca filtru Wiener care se spune că este optim în sensul mediei pătratice.

În situații de nestaționaritate a semnalului de intrare, filtrul optim are o formă variabilă în timp. O soluție de succes este oferită de filtrul Kalman [2].

Atât filtrele Wiener cât și Kalman au fost dezvoltate în literatură atât pentru semnale în timp continuu cât și pentru cele discrete în timp.

Istoric, cele mai frecvent utilizate prelucrări ale semnalelor au fost cele de tip liniar. Există un număr important de motive de utilizare a tehnicilor liniare printre care cel mai important este existența unei bogate și elegante teorii referitoare la sistemele liniare. Din nefericire, în ciuda acestei teorii, multe probleme de prelucrare a semnalelor nu pot fi rezolvate satisfăcător prin utilizarea sistemelor liniare.

1.2. Filtre neliniare

Filtrul este neliniar dacă filtrarea, netezirea sau predicția mărimii de interes este un operator neliniar de datele aplicate la intrarea filtrului.

Filtrele adaptive uzuale bazate pe metoda gradientului și obținute pe baza implementării algoritmului LMS utilizează o estimare instantanee asupra gradientului la suprafața erorii medii patratice pentru a reactualiza coeficienții filtrului. O astfel de strategie implică faptul că algoritmul LMS este extrem de vulnerabil la influența impulsurilor. Din acest punct de vedere filtrele neliniare permit obținerea unor performanțe ale filtrelor adaptive mai apropiate de cerințele optime și într-o gamă mult mai largă în ceea ce privește domeniul evenimentelor de la intrarea filtrului începând cu zgomotul gaussian pînă la impulsurile de amplitudine mare și durată scurtă.

1.3. Filtre adaptive

Construcția unui filtru Wiener necesită informații apriori asupra statisticii datelor ce urmează a fi prelucrate. Filtrul este optim doar dacă caracteristicile statistice ale datelor de intrare se potrivesc cu informația apriori pe baza căreia a fost construit filtrul [1],[2].

Există două etape de determinare a parametrilor filtrului: prima în care se face o estimare a parametrilor statistici ai semnalului și a doua în care se determină parametrii filtrului printr-o formulă nerecursivă de calcul.

O metodă eficientă este aceea de a utiliza un filtru adaptiv, înțelegând prin aceasta că filtrul se autogenerază printr-un algoritm recursiv, fapt ce face posibil ca filtrul să funcționeze satisfăcător fără completa cunoaștere a caracteristicilor esențiale ale semnalului.

În literatură s-a dezvoltat o mare varietate de algoritmi recursivi utilizați la implementarea filtrelor adaptive[1],[2],[4],[11],[38]. Alegerea unui algoritm este determinată de numeroși factori:

1. **Viteza de convergență** - definită ca numărul de iterații necesar algoritmului, în răspunsul la un semnal de intrare staționar, pentru a converge suficient de mult la soluția Wiener optimă în sensul erorii medii pătratice. O viteză mare de convergență permite algoritmului o adaptare rapidă la evenimente staționare cu statistică necunoscută.

2. **Eroarea de ajustare** - reprezintă o măsură cantitativă a diferenței între valoarea finală a erorii medii pătratice, mediată pe un ansamblu de filtre adaptive și eroarea medie pătratică produsă de filtrul Wiener.

3. **Urmărirea** - constă în faptul că atunci când un algoritm de intrare operează pentru evenimente nestaționare este necesar ca algoritmul să urmărească variațiile statistice ale evenimentului. Performanțele algoritmului în ceea ce privește urmărirea sunt influențate de două caracteristici contradictorii: viteza de convergență și zgomotul algoritmului.

4. **Robustetea** - se referă la abilitatea algoritmului de a opera satisfăcător cu date de intrare a căror valori nu respectă o anumită caracteristică.

5. **Volumul de calcul** - reprezintă o caracteristică a algoritmului care include :

a. -numărul de operații (multiplicări , diviziuni, adunări, scăderi) necesar pentru realizarea unei iterații complete a algoritmului;

b. -dimensiunea memoriei necesare pentru a stoca datele și programul;

c. -investiția necesară programării algoritmului pe un calculator.

6. **Structura** - se referă la structura care se obține în situația implementării hardware a algoritmului.

7. **Proprietățile numerice** - se referă la neacuratețea algoritmului datorată erorilor de cuantizare. Acestea pot apărea fie datorită conversiei analog numerice a datelor de la intrare, fie datorită reprezentării interne a lor în memoria calculatorului.

1.4. Aplicații ale filtrelor adaptive

Deși aplicațiile filtrării adaptive sunt în general extrem de diferite, ele au totuși o caracteristică comună: un vector de date de intrare și un răspuns dorit sunt utilizați pentru calculul estimativ al erorii, utilizată la rândul ei la controlul coeficienților filtrului. Acești coeficienți, în funcție de structura filtrului, pot fi: coeficienți pondere în cazul filtrelor transversale, coeficienți de reflexie în cazul filtrelor lattice, respectiv parametri de rotație în cazul matricei sistolice.

Există patru clase de aplicații ale filtrării adaptive, prezentate în figura 1.1. Notațiile utilizate în figuri sunt:

x - semnalul aplicat la intrarea filtrului adaptiv;

u - semnalul aplicat la intrarea sistemului;

y - semnalul de la ieșirea filtrului adaptiv;

d - răspunsul dorit;

e = d - y - eroarea estimată.

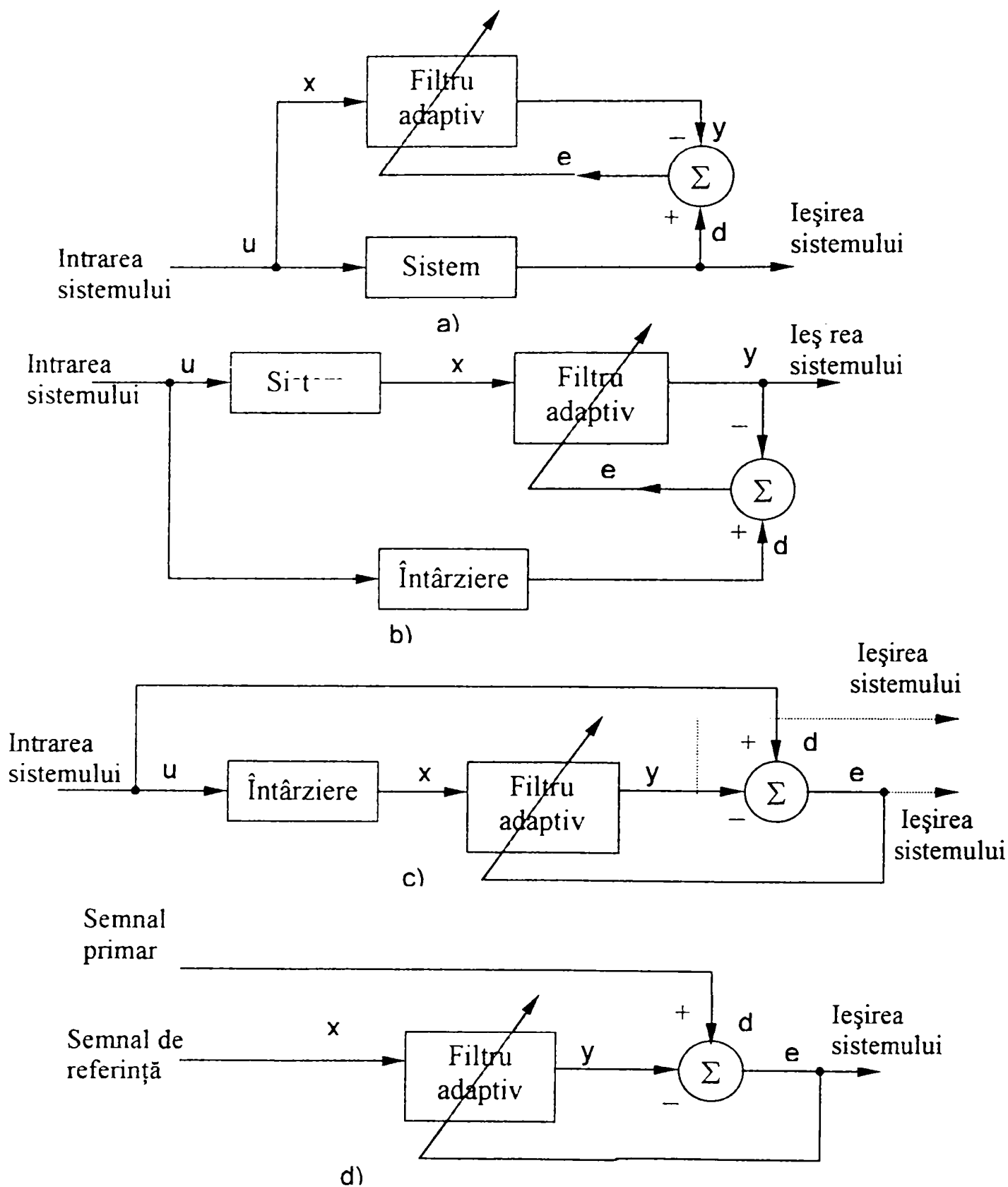


Fig. 1.1. Principalele categorii de aplicații a filtrelor adaptive : a) identificarea; b) modelarea inversă; c) predicția; d) eliminarea interferențelor.

Funcțiile celor patru clase de bază de aplicații ale filtrelor adaptive prezentate în [2] sunt:

I. Identificarea (fig.1.1.a). În categoria aplicațiilor care se ocupă cu identificarea, un filtru adaptiv este utilizat pentru a furniza un model linear care reprezintă cea mai bună aproximare a sistemului necunoscut. Sistemul și filtrul adaptiv au la intrare același semnal, sistemul furnizând la ieșire răspunsul dorit a se obține și la ieșirea filtrului. Dacă sistemul are natură dinamică, modelul obținut va fi variabil în timp.

II. Modelarea inversă (fig. 1.1.b). În această categorie de aplicații, funcția filtrului adaptiv este de a furniza un model invers care să reprezinte cea mai bună aproximare a unui sistem necunoscut. În mod ideal, modelul invers are funcția de transfer egală cu inversa funcției de transfer a sistemului. Răspunsul dorit al filtrului adaptiv îl reprezintă versiunea întârziată a semnalului de intrare.

III. Predicția (Fig.1.1.c). În această categorie, funcția filtrului adaptiv este de a furniza cea mai bună predicție a valorii curente a unui semnal necunoscut. Valoarea curentă a semnalului servește ca răspuns dorit al filtrului. În funcție de aria de interes, ca ieșire a sistemului poate fi utilizată fie ieșirea filtrului adaptiv, fie eroarea de predicție. În primul caz sistemul poartă denumirea de predictor, iar în al doilea caz poartă denumirea de predictor al erorii de filtrare.

IV. Eliminarea interferențelor (fig.1.1.d). În această categorie de aplicații filtrul adaptiv este utilizat pentru eliminarea interferențelor necunoscute conținute în semnalul primar, eliminarea fiind optimizată din punct de vedere al unui criteriu. Semnalul primar servește ca răspuns dorit în timp ce un semnal de referință este adus la intrarea filtrului adaptiv.

2. FILTRE ADAPTIVE LINIARE

În acest capitol autorul prezintă un studiu referitor la posibilitățile de implementare a filtrelor adaptive liniare. Pentru realizarea comparației între diferiții algoritmi prezentați autorul a simulat pe calculator comportarea unui canal de comunicație influențat de zgomot gaussian. Cunoscând parametrii canalului simulat, precum și ai zgomotului care influențează canalul, autorul a calculat pe baza metodelor prezentate în literatură coeficienții optimi de predicție ai filtrului Wiener. Apoi, utilizând diferiți algoritmi de determinare a coeficienților filtrului Wiener și efectuând același număr de rulări în cazul fiecărui algoritm studiat, autorul a obținut concluzii referitoare la principalii factori care determină alegerea unui algoritm (viteza de convergență, eroarea de ajustare, volumul de calcule, etc).

2.1. Eroarea medie pătratică

Așa cum s-a prezentat anterior, expresia erorii de predicție este dată de relația

$$e(n) = d(n) - y(n), \quad (2.1)$$

unde $d(n)$ este ieșirea sistemului, iar $y(n)$ reprezintă ieșirea filtrului de predicție.

O posibilă măsură a erorii de predicție prezentată în [1] este valoarea absolută a erorii de predicție $|e(n)|$. Această măsură în analiza matematică duce de multe ori la rezultate neutilizabile în tentativa analizei algoritmilor rezultați.

Măsura eroare pătratică de predicție $e^2(n)$ se pretează mult mai bine analizei matematice. Eroarea pătratică este dată de relația

$$e^2(n) = [d(n) - y(n)]^2.$$

Eroarea medie pătratică se definește ca o medie pe ansamblu a secvenței pătratului erorii. Deoarece eroarea medie pătratică este o medie pe ansamblu și nu o medie în timp se definește eroarea medie pătratică ca și o funcție de timp dată de relația

$$e(n) = E\{e^2(n)\}, \quad (2.2)$$

unde cu $E\{\cdot\}$ s-a notat operatorul de mediere.

Considerând vectorul datelor de intrare $\mathbf{x}(n)$, se definește filtrul adaptiv caracterizat prin setul de N coeficienți $w_i(n)$, acel filtru pentru care predicția semnalului dorit se determină potrivit relației

$$y(n) = \sum_{i=0}^{N-1} w_i(n) \cdot x(n-i) = \mathbf{w}_N^T(n) \cdot \mathbf{x}_N(n), \quad (2.3)$$

unde

$$\mathbf{w}_N^T(n) = [w_0(n), \dots, w_{N-1}(n)] \quad (2.3 a)$$

$$\mathbf{x}_N^T(n) = [x(n), \dots, x(n-N+1)] \quad (2.3 b)$$

relații în care N reprezintă numărul de componente ale vectorului și n reprezintă argumentul timp.

Pentru fiecare valoare a vectorului coeficienților $\mathbf{w}_N(n)$ rezultă o valoare corespunzătoare a erorii medii pătratice a predicției, această dependență funcțională notându-se $\varepsilon(\mathbf{w}_N)$. S-a arătat în [1] că eroarea medie pătratică este minimă dacă setul de coeficienți \mathbf{w}_N verifică ecuația

$$\mathbf{R}_{NN} \cdot \mathbf{w}_N^* = \mathbf{p}_N, \quad (2.4)$$

ecuație denumită ecuația normală și în care s-au notat :

$$\mathbf{p}_N = E\{y(n)\mathbf{x}_N(n)\} \quad \text{vectorul de intercorelație;}$$

$$\mathbf{R}_{NN} = E\{\mathbf{x}_N(n)\mathbf{x}_N^T(n)\} \quad \text{matricea de autocorelație.}$$

Vectorul de intercorelație se poate determina pe baza relației

$$\mathbf{p}_N = \begin{bmatrix} E\{y(n)x(n)\} \\ E\{y(n)x(n-1)\} \\ \dots \\ E\{y(n)x(n-N+1)\} \end{bmatrix} = \begin{bmatrix} \varphi_{xy}(0) \\ \varphi_{xy}(1) \\ \dots \\ \varphi_{xy}(N-1) \end{bmatrix},$$

unde φ_{xy} sunt coeficienții de intercorelație și sunt definiți de relația

$$\varphi_{xy}(m) = E\{x(n)y(n-m)\} = E\{y(n)x(n-m)\}.$$

Matricea de autocorelație se poate determina pe baza relației

$$\mathbf{R}_{NN} = \begin{bmatrix} \varphi(0) & \varphi(1) & \dots & \varphi(N-1) \\ \varphi(1) & \varphi(0) & \dots & \varphi(N-2) \\ \dots & \dots & \dots & \dots \\ \varphi(N-1) & \varphi(N-2) & \dots & \varphi(0) \end{bmatrix},$$

unde φ_x sunt coeficienții de autocorelație ai semnalului staționar $x(n)$ definiți de relația

$$\varphi_x(m-k) = \varphi_x(k-m) = E\{x(n-m)x(n-k)\}.$$

Pentru rezolvarea ecuației matriciale (2.4) este necesară estimarea vectorului de intercorelație și a matricei de autocorelație. În [3] sunt prezentați principalii estimatori utilizați în literatură :

a) pentru estimarea dispersiei σ_y^2 a semnalului $y(n)$ se poate utiliza estimatorul deplasat

$$\hat{\sigma}_y^2 = \frac{1}{N} \cdot \sum_{i=0}^{N-1} [y(i) - \hat{m}_y]^2,$$

unde \hat{m}_y reprezintă estimatorul mediei semnalului $y(n)$ pentru care se poate folosi relația

$$m_1 = \frac{1}{N} \cdot \sum_{i=1}^{N-1} y(i) ;$$

b) estimarea coeficienților de intercorelație se face utilizând datele disponibile cu relația

$$\hat{\varphi}_{xx}(m) = \frac{1}{N} \cdot \sum_{i=0}^{N-|m|-1} x(n-i)y(n-|m|-i) ,$$

unde m reprezintă parametrul funcției de intercorelație ;

c) estimarea coeficienților de autocorelație, în mod asemănător, se face cu relația

$$\hat{\varphi}_{xx}(m) = \frac{1}{N} \cdot \sum_{i=0}^{N-|m|-1} x(n-i)x(n-|m|-i)$$

Rezolvarea ecuației normale (2.4) permite obținerea setului coeficienților de predicție \mathbf{w}_N , pentru care eroarea medie pătratică este minimă, filtrul realizat cu acești coeficienți purtând denumirea de filtru Wiener. Dacă matricea coeficienților de autocorelație este inversabilă, setul de coeficienți \mathbf{w}_N^* se poate obține pe baza relației

$$\mathbf{w}_N^* = \mathbf{R}_{NN}^{-1} \cdot \mathbf{p}_N \quad (2.5)$$

Relația (2.5) se utilizează foarte rar în practică datorită faptului că este necesar calculul inversei matricei de autocorelație \mathbf{R}_{NN} , fiind deci necesar un volum de calcule mare. În principal s-au dezvoltat două categorii de metode de rezolvare a ecuației (2.4), cunoscute ca metoda pantei descendente maxime și algoritmul lui Durbin, fiecare dintre acestea conducând la tehnici de prelucrare a semnalelor des utilizate.

2.2. Calculul coeficienților filtrului Wiener

Pentru exemplificarea modului de determinare a coeficienților filtrului transversal s-a ales răspunsul dorit $d(n)$ ca fiind semnalul obținut la ieșirea unui proces autoregresiv de ordin 1. Acest semnal a fost produs aplicând la intrarea unui filtru cu un pol a unui zgomot alb $v_1(n)$ de medie nulă și dispersie $\sigma_v^2 = 0,28$, (fig.2.1.a). S-a ales funcția de transfer a filtrului dată de ecuația

$$H_1(z) = \frac{1}{1 + 0,7943 \cdot z^{-1}}$$

Semnalul $d(n)$ este aplicat canalului de comunicație modelat de funcția de transfer

$$H_2(z) = \frac{1}{1 - 0,9443 \cdot z^{-1}}$$

Coefficienții care intervin în expresiile funcțiilor de transfer din relațiile anterioare au fost astfel aleși încât ecuația cu diferențe a procesului autoregresiv de ordin 2 să aibă coeficienții c_1 și c_2 de valori $c_1 = -0,15$, respectiv $c_2 = -0,75$. Aceste valori nu sunt restrictive în studiul ce se va face ulterior, ele influențând doar valorile finale ale coeficienților optimi ai filtrului Wiener, nu și performanțele algoritmilor studiați.

Autorul a verificat experimental că, prin alegerea altor valori ale coeficienților c_1 și c_2 , respectiv a valorilor dispersiilor zgomotelor, valorile finale ale coeficienților optimi ai filtrului Wiener se modifică, dar performanțele comparative ale algoritmilor rămân aceleași.

Canalul de comunicație este influențat de zgomotul gaussian $v_2(n)$ de medie nulă și dispersie $\sigma_2^2 = 0,1$, (fig.2.1.b), necorelat cu $v_1(n)$.

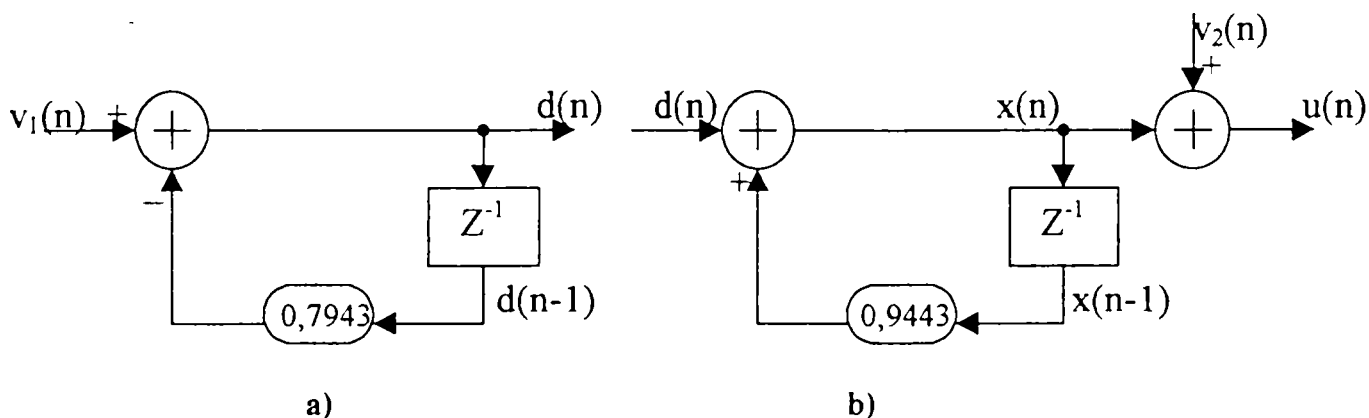


Fig.2.1. a) modelul autoregresiv al semnalului dorit $d(n)$;
b) modelul canalului de comunicații zgomotos.

Semnalul recepționat va fi dat de ecuația

$$u(n) = x(n) + v_2(n). \quad (2.6)$$

Rolul filtrului Wiener este de a furniza cea mai bună aproximare a semnalului de la intrarea canalului de comunicație, $d(n)$, utilizând semnalul recepționat $u(n)$.

Pentru determinarea coeficienților filtrului se începe cu analiza ecuației cu diferențe ce permite generarea răspunsului dorit $d(n)$

$$d(n) + a_1 \cdot d(n-1) = v_1(n),$$

unde $a_1 = 0,7943$. Dispersia procesului $d(n)$ este dată de ecuația

$$\sigma_d^2 = \frac{\sigma_1^2}{1 - a_1^2} = \frac{0,28}{1 - (0,7943)^2} = 0,7586.$$

Deoarece semnalul $d(n)$ acționează la intrarea canalului, semnalul $x(n)$ este dat de ecuația

$$x(n) + b_1 \cdot x(n-1) = d(n), \quad (2.7)$$

unde b_1 rezultă din funcția de transfer $H_2(z)$ a filtrului, $b_1 = -0,9443$.

Se observă că semnalul $x(n)$ poate fi obținut și prin aplicarea semnalului $v_1(n)$ la intrarea unui filtru cu doi poli a cărui funcție de transfer este

$$H(z) = H_1(z) \cdot H_2(z) = \frac{1}{(1 + 0,7943 \cdot z^{-1}) \cdot (1 - 0,9443 \cdot z^{-1})},$$

astfel că semnalul $x(n)$ este un proces autoregresiv de ordin 2 descris de ecuația cu diferențe

$$x(n) + c_1 \cdot x(n-1) + c_2 \cdot x(n-2) = v_1(n),$$

unde $c_1 = -0,15$; $c_2 = -0,75$.

Pentru determinarea coeficienților filtrului autorul a rezolvat ecuația (2.5) fapt ce a necesitat cunoașterea:

1. Matricei de corelație \mathbf{R}_{NN} a semnalului recepționat $u(n)$;
2. Vectorului de intercorelație \mathbf{p}_N între semnalul $u(n)$ și răspunsul dorit $d(n)$. În exemplul ales $N = 2$.

Semnalul recepționat $u(n)$, de la ieșirea canalului, constă în însumarea semnalului $x(n)$ cu zgomotul aditiv $v_2(n)$. Deoarece $x(n)$ și $v_2(n)$ sunt necorelate, se poate scrie

$$\mathbf{R}_{NN} = \mathbf{R}_x + \mathbf{R}_2 \quad (2.8)$$

unde \mathbf{R}_x este matricea de autocorelație a semnalului $x(n)$, iar \mathbf{R}_2 este matricea de autocorelație a lui $v_2(n)$.

S-a arătat în [2] că:

$$r_x(0) = \sigma_x^2 = \frac{1+c_2}{1-c_2} \cdot \frac{\sigma_1^2}{(1+c_2)^2 - c_1^2} = \frac{1-0,75}{1+0,75} \cdot \frac{0,28}{(1-0,75)^2 - (0,15)^2} = 1,$$

$$r_x(1) = \frac{-c_1}{1+c_2} = \frac{0,15}{1-0,75} = 0,6,$$

rezultate care permit scrierea matricei de autocorelație \mathbf{R}_x sub forma

$$\mathbf{R}_x = \begin{bmatrix} r_x(0) & r_x(1) \\ r_x(1) & r_x(0) \end{bmatrix} = \begin{bmatrix} 1 & 0,6 \\ 0,6 & 1 \end{bmatrix} \quad (2.9)$$

Deoarece semnalul v_2 este un proces necorelat de medie nulă și dispersie $\sigma_1^2 = 0,1$, matricea de autocorelație \mathbf{R}_2 poate fi scrisă sub forma

$$\mathbf{R}_2 = \begin{bmatrix} 0,1 & 0 \\ 0 & 0,1 \end{bmatrix} \quad (2.10)$$

Înlocuind (2.9) și (2.10) în (2.8) se obține

$$\mathbf{R}_{NN} = \begin{bmatrix} 1,1 & 0,6 \\ 0,6 & 1,1 \end{bmatrix}.$$

Vectorul de intercorelație conține funcțiile de intercorelație între semnalele $d(n)$ și $u(n)$ pentru întârzieri de 0, respectiv 1 eșanțon. Deoarece procesele sunt reale $p(k) = p(-k)$, unde

$$p(-k) = E\{u(n-k)d(n)\}. \quad (2.11)$$

Înlocuind (2.6) și (2.7) în (2.11) și ținând seama că $x(n)$ este necorelat cu $v_2(n)$ se obține

$$p(k) = r_x(k) + b_1 \cdot r_x(k-1),$$

unde $b_1 = -0,9443$

Se obțin pe rând:

$$p(0) = r_x(0) + b_1 \cdot r_x(-1) = 1 - 0,9443 \cdot 0,6 = 0,4334,$$

$$p(1) = r_x(1) + b_1 \cdot r_x(0) = 0,6 - 0,9443 \cdot 1 = -0,3443,$$

deci

$$\mathbf{p}_N = \begin{bmatrix} 0,4334 \\ -0,3443 \end{bmatrix}.$$

Determinarea coeficienților filtrului Wiener se face pe baza relației (2.5) unde

$$\mathbf{R}_{NN}^{-1} = \begin{bmatrix} r(0) & r(1) \\ r(1) & r(0) \end{bmatrix}^{-1} = \frac{1}{r(0)^2 - r(1)^2} \begin{bmatrix} r(0) & -r(1) \\ -r(1) & r(0) \end{bmatrix} = \begin{bmatrix} 1,2941 & -0,7058 \\ -0,7058 & 1,2941 \end{bmatrix}.$$

Din (2.5) se obține

$$\mathbf{w}_2^* = \begin{bmatrix} 1,2941 & -0,7058 \\ -0,7058 & 1,2941 \end{bmatrix} \cdot \begin{bmatrix} 0,4334 \\ -0,3443 \end{bmatrix} = \begin{bmatrix} 0,8038 \\ -0,7516 \end{bmatrix}. \quad (2.12)$$

În [1],[2] s-a prezentat expresia erorii medii pătratice în funcție de setul de coeficienți de predicție \mathbf{w}_N

$$\varepsilon(\mathbf{w}_N) = \sigma_d^2 - 2 \cdot \mathbf{w}_N^T \cdot \mathbf{p}_N + \mathbf{w}_N^T \cdot \mathbf{R}_{NN} \cdot \mathbf{w}_N \quad (2.13)$$

Se observă că eroarea medie pătratică depinde de cei N coeficienți w_i , deci reprezentarea în spațiul $N + 1$ dimensional va da o suprafață care pe axa $\varepsilon(\mathbf{w}_N)$ va prezenta un minim pentru valori ale coeficienților w_i identici cu cei ai funcției răspuns la impuls a sistemului. În modelul prezentat în care există doi coeficienți de predicție w_0 și w_1 , expresia erorii medii pătratice este dată de relația

$$\varepsilon(w_0, w_1) = 0,7586 - 2 \cdot 0,4334 \cdot w_0 + 2 \cdot 0,3443 \cdot w_1 + 1,2 \cdot w_0 \cdot w_1 + 1,1 \cdot w_0^2 + 1,1 \cdot w_1^2. \quad (2.14)$$

Variația erorii medii pătratice funcție de coeficienții w_0, w_1 , este prezentată în graficul din figura 2.2.a), iar în figura 2.2.b) sunt reprezentate curbele de nivel ale suprafeței erorii medii pătratice.

Setul de coeficienți optimi pentru care se obține eroarea medie pătratică minimă se obține pe baza ecuațiilor

$$\left[\frac{\partial}{\partial w_i} \cdot \varepsilon(\mathbf{w}_N) \right]_{\mathbf{w}_N = \mathbf{w}_N^*} = 0 \quad i = 0, \dots, N-1, \quad (2.15)$$

ecuații care permit obținerea relațiilor (2.5).

Graficele din figura 2.2.c) și figura 2.2.d) reprezintă proiecția suprafeței erorii medii pătratice pe un plan format din axa erorii și axa coeficientului w_0 , respectiv w_1 . Se observă că valoarea minimă a erorii care se obține este pentru valorile coeficienților de predicție date de cele din relația (2.12). **Programele scrise de autor** în MATLAB ce permit obținerea graficelor din figura 2.2 sunt prezentate în Anexa II.1.

Valoarea minimă a erorii medii pătratice este prezentată în [1] și [2]

$$\varepsilon_{\min} = \varepsilon(\mathbf{w}_N^*) = \sigma_d^2 - \mathbf{w}_N^T \cdot \mathbf{p}_N = 0,7586 - [0,4334, -0,3443] \cdot \begin{bmatrix} 0,8038 \\ -0,7516 \end{bmatrix} = 0,1515,$$

valoare ce corespunde cu graficele din figura 2.2.

2.3. Metoda pantei descendente maxime

Soluția ecuației (2.5), în afara calculului direct, se poate obține și printr-o metodă iterativă cunoscută ca metoda pantei descendente maxime. Datorită relațiilor obținute, această metodă este rar utilizată în prelucrarea adaptivă a semnalelor, în schimb aproximații ale acesteia sunt foarte des utilizate. Metoda s-a dezvoltat pe baza interpretării geometrice a suprafeței erorii medii pătratice. Considerând situația prezentată în figura 2.3 pentru doi coeficienți de predicție w_0 și w_1 , suprafața erorii medii pătratice prezintă un minim la punctul (w_0^*, w_1^*) . Alegând coeficienții inițiali $\mathbf{w}_N(0)$, ($N=2$), punctul corespunzător de pe suprafața erorii medii pătratice se caracterizează prin eroarea medie pătratică $\varepsilon(0)$.

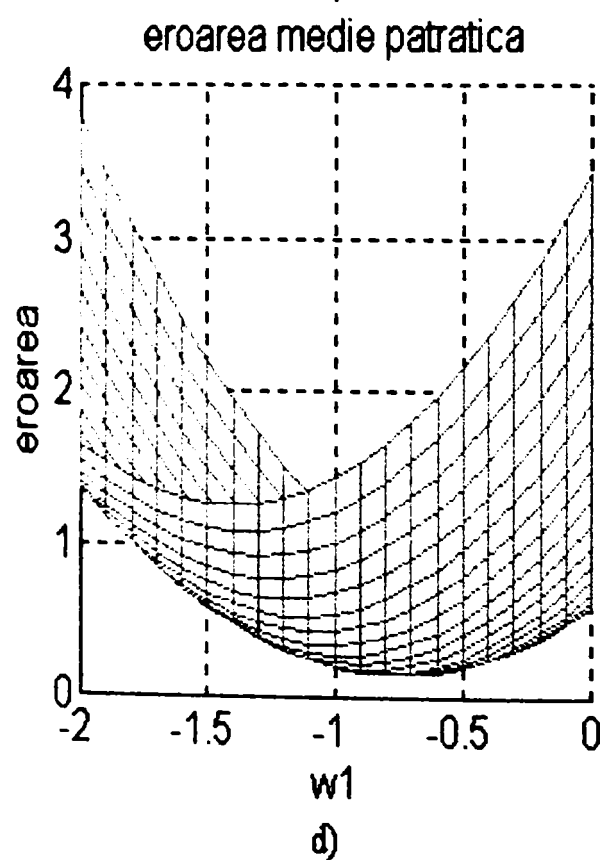
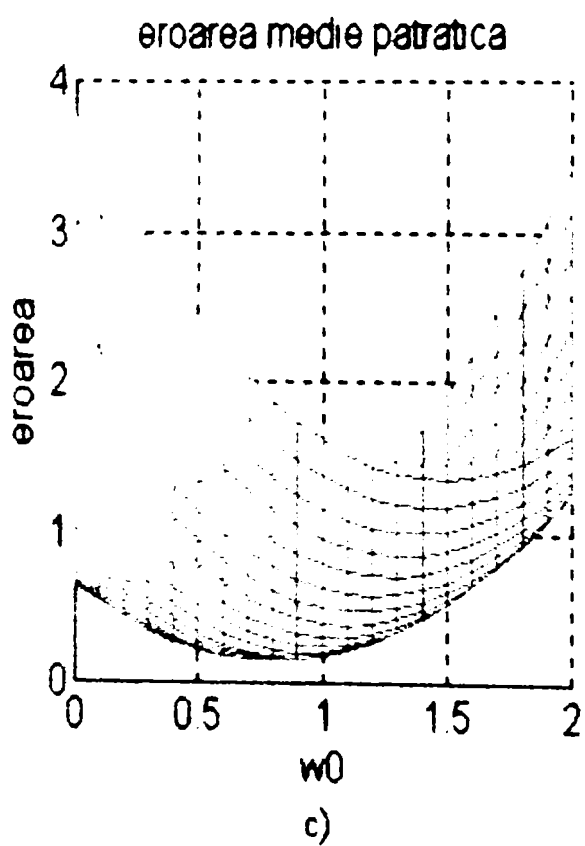
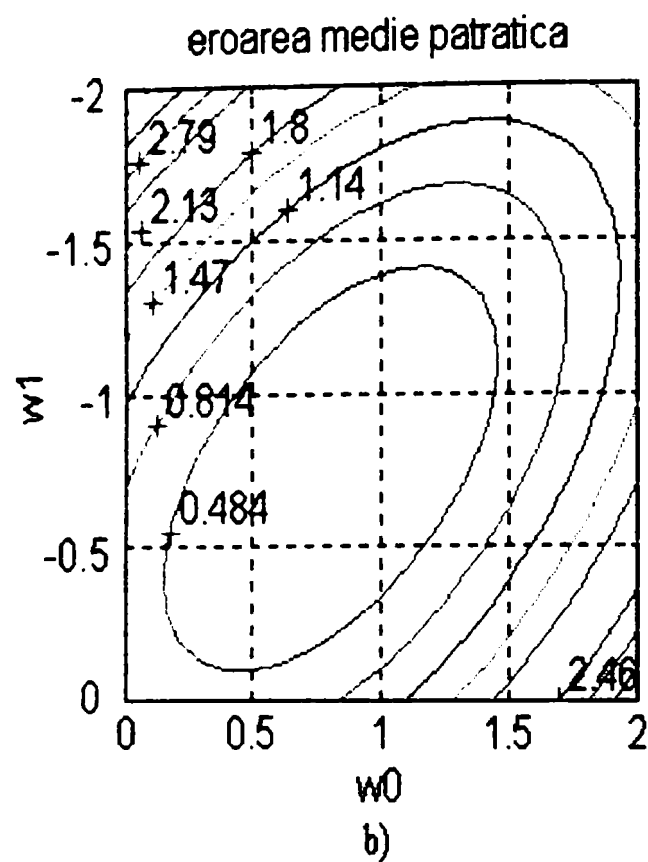
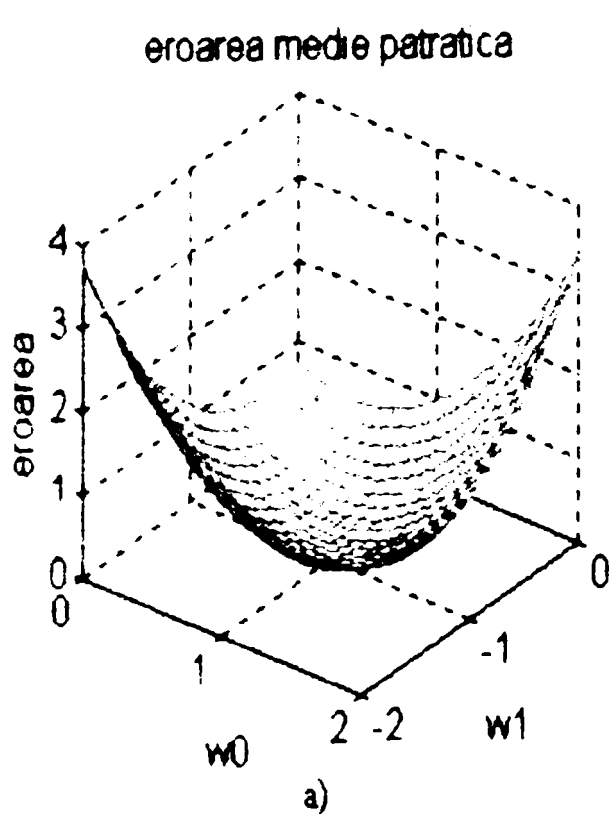


Fig. 2.2. Variația erorii medii pătratice pentru doi coeficienți de predicție: a) reprezentare tridimensională; b) proiecția erorii prin curbe de nivel ; c) dependența erorii de coeficientul w_0 ; d) dependența erorii de coeficientul w_1 .

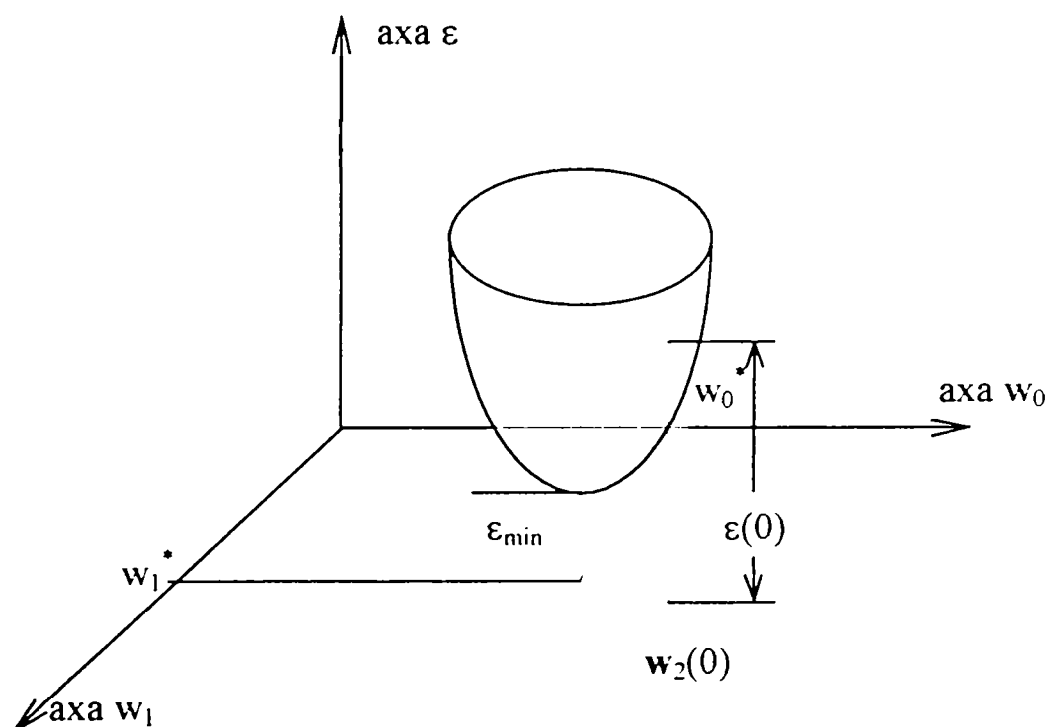


Fig. 2.3. Influența alegerii valorilor inițiale ale coeficienților de predicție asupra erorii medii pătratice.

În acest punct suprafața poate fi descrisă prin planul tangent la suprafață, care cuantifică rata modificărilor pe suprafața erorii medii pătratice raportată la axele de coordonate w_N . Tangentele la suprafață în punctul respectiv, obținute prin intersecția dintre planul tangent la suprafață și plane formate din axa erorii medii pătratice și axa unui coeficient sunt date de derivatele parțiale $\partial\varepsilon/\partial w_i$ calculate în punctul $\varepsilon(0)$. Se notează $\varepsilon = \varepsilon(n)$ evaluarea erorii medii pătratice în punctul de coordonate $w_N(n)$, unde n reprezintă numărul iterației.

Se definește gradientul suprafeței erorii ca fiind vectorul format de derivatele parțiale

$$\nabla_w [\varepsilon] = \frac{\partial \varepsilon}{\partial w_N} \quad (2.16)$$

Din definiție gradientul dat de relația (2.16) este orientat în direcția maximei modificări a suprafeței erorii medii pătratice, în sensul creșterii erorii medii pătratice. Presupunând că în punctul de coordonate $(w_N(0), \varepsilon(0))$ s-ar pune o bilă imaginară, dacă bilei i s-ar da drumul, aceasta s-ar rostogoli înspre punctul de minim al suprafeței pe direcția axei erorii, iar inițial va porni în direcția pantei descendente maxime a suprafeței. Acest sens este opus gradientului suprafeței. Bila eliberată poate trece de poziția de minim a suprafeței dar în timp se va stabiliza în poziția de minim, sugerând că un algoritm bazat pe metoda modificării valorilor coeficienților filtrului în funcție de valoarea curentă a gradientului la suprafața erorii medii pătratice este convergent.

Acest concept privește problema erorii medii pătratice din punct de vedere geometric (și gravitațional) și este similar cu mecanismul metodei pantei descendente maxime. Metoda rezolvă ecuația normală printr-un algoritm matematic ce descrie rostogolirea bilei. Considerând poziția bilei la intervale discrete de timp, aceasta se va mișca într-o manieră dependentă de gradientul erorii și de punctul în care se găsește. Echivalentul matematic al acestei proprietăți

este că, fiind dată poziția bilei pe suprafața erorii medii pătratice la momentul de timp n (sau la iterația n), poziția bilei la iterația $n+1$ este considerată a fi în sensul opus gradientului. Acest lucru poate fi scris formal astfel: dacă poziția inițială a bilei era punctul de coordonate $(\mathbf{w}_N(0), \varepsilon(0))$, poziția următoare va fi punctul $(\mathbf{w}_N(1), \varepsilon(1))$ exprimată prin relația

$$\begin{bmatrix} \mathbf{w}_N(1) \\ \varepsilon(1) \end{bmatrix} = \begin{bmatrix} \mathbf{w}_N(0) - \mu \nabla_{\mathbf{w}} [\varepsilon(0)] \\ \varepsilon(0) - \delta \end{bmatrix} \quad (2.17)$$

unde μ este o constantă de adaptare, iar δ este variația erorii medii pătratice între cele două iterații

Prin generalizarea relației (2.17) se obține formula de recurență

$$\mathbf{w}_N(n+1) = \mathbf{w}_N(n) - \mu \nabla_{\mathbf{w}} [\varepsilon(n)] \quad (2.18)$$

Această relație exprimă noua valoare a vectorului \mathbf{w}_N ca o funcție de vechea valoare și de un termen de corecție care depinde de proprietățile suprafeței erorii și de poziția punctului precedent.

Scopul metodei este de a rezolva ecuația normală deci de a găsi coeficienții \mathbf{w}_N^* .

Aplicarea metodei în cazul filtrării în sensul erorii medii pătratice minime constă în calculul gradientului suprafeței erorii în funcție de matricea de autocorelație, \mathbf{R}_{NN} , și de vectorul de intercorelație \mathbf{p}_N .

Eroarea medie pătratică se modifică în funcție de numărul iterației și de aceea se notează cu $\varepsilon(n)$. Gradientul lui $\varepsilon(n)$ este dat în [1]

$$\nabla_{\mathbf{w}} [\varepsilon(n)] = -2\mathbf{p}_N + 2\mathbf{R}_{NN} \cdot \mathbf{w}_N(n) \quad (2.19)$$

Înlocuind (2.19) în (2.18) se obține

$$\mathbf{w}_N(n+1) = \mathbf{w}_N(n) + 2\mu[\mathbf{p}_N - \mathbf{R}_{NN}\mathbf{w}_N(n)]$$

și deoarece μ este o constantă se notează $\alpha = 2\mu$ obținându-se relația

$$\mathbf{w}_N(n+1) = \mathbf{w}_N(n) + \alpha[\mathbf{p}_N - \mathbf{R}_{NN}\mathbf{w}_N(n)] \quad (2.20)$$

care este relația utilizată de metoda pantei descendente maxime.

Notând cu \mathbf{I}_{NN} matricea unitate, relația (2.20) mai poate fi scrisă sub forma

$$\mathbf{w}_N(n+1) = [\mathbf{I}_{NN} - \alpha \cdot \mathbf{R}_{NN}] \mathbf{w}_N(n) + \alpha \cdot \mathbf{p}_N \quad (2.21)$$

Procedura de aplicare directă a metodei prezentate se bazează pe estimarea matricei de autocorelație și a vectorului de intercorelație pe baza datelor disponibile, după care sunt efectuate iterații succesive utilizând matricea \mathbf{R}_{NN} și vectorul \mathbf{p}_N . Această metodă este utilizată rar în practică datorită complexității calculului care nu mai pot fi simplificate din punct de vedere matematic.

În [1] s-a demonstrat că metoda pantei descendente maxime este convergentă și setul de coeficienți converge la coeficienții optimi \mathbf{w}_N^* .

2.3.1. Rezolvarea directă a ecuației cu diferențe

Pomind de la relația de recurență (2.21), în [1] s-a dedus o relație care permite exprimarea coeficienților $\mathbf{w}_N(n)$ în funcție de coeficienții de la care se pornește inițial $\mathbf{w}_N(0)$ și de numărul iterației n

$$\mathbf{w}_N(n) = [\mathbf{I}_{NN} - \alpha \cdot \mathbf{R}_{NN}]^n \cdot \mathbf{w}_N(0) + \alpha \cdot \mathbf{p}_N \cdot \sum_{j=0}^{n-1} [\mathbf{I}_{NN} - \alpha \cdot \mathbf{R}_{NN}]^j \quad (2.22)$$

Această relație permite calculul vectorului $\mathbf{w}_N(n)$ la iterația n dar această evaluare nu este foarte viabilă deoarece trebuie calculate puterile unor matrice, procedeu matematic relativ complex.

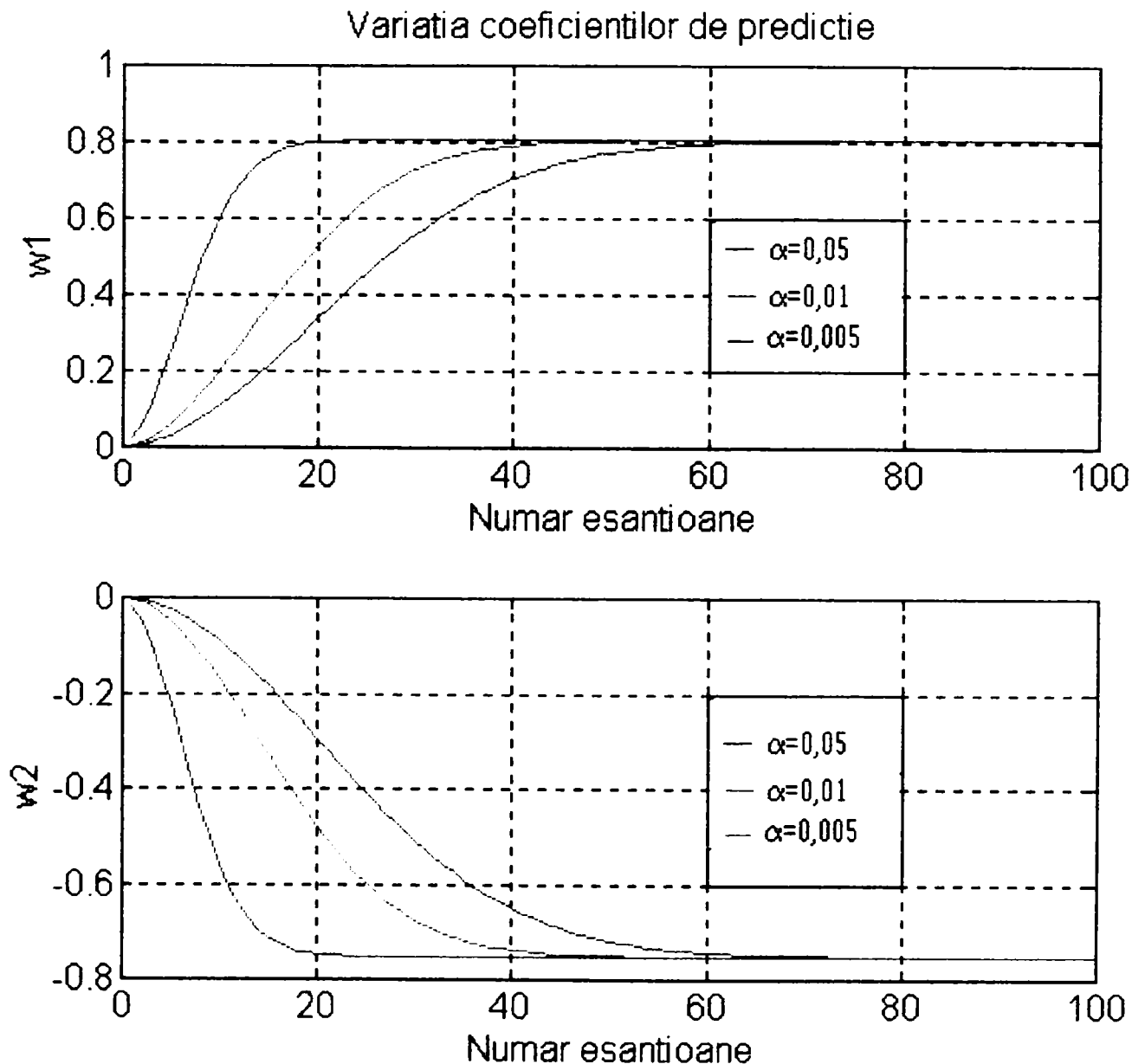


Fig. 2.4. Variația coeficienților de predicție pe baza evaluării directe a ecuației cu diferențe.

În Anexa II.2 este prezentat **programul realizat de autor care permite reprezentarea variației coeficienților de predicție**, prezentată în figura 2.4, **în funcție de diferite valori ale parametrului α** , considerând că matricea de autocorelație și vectorul de intercorelație sunt cele determinate pe cale analitică așa cum s-a prezentat anterior.

Se observă că **valorile coeficienților de predicție care se obțin sunt practic cele deduse pe cale teoretică**. De asemenea, odată cu creșterea valorii parametrului α se obține o mai rapidă convergență a valorilor coeficienților de predicție.

Relația (2.22) poate fi simplificată prin efectuarea unei translații din spațiul de coordonate w_N în spațiul de coordonate v_N , translație care translatează conturul intersecției suprafeței erorii medii pătratice cu plane paralele cu planul coeficienților, din conturul din figura 2.5.a. în conturul din figura 2.5.b.

Relația de translație a coordonatelor este

$$v_N = w_N - w_N^{\circ}$$

relație care duce la relația de recurență în planul v_N

$$v_N(n+1) = [I_{NN} - \alpha \cdot R_{NN}] \cdot v_N(n), \text{ sau}$$

$$v_N(n) = [I_{NN} - \alpha \cdot R_{NN}]^n \cdot v_N(0),$$

iar în spațiul w_N

$$w_N(n) = w_N^{\circ} + [I_{NN} - \alpha \cdot R_{NN}]^n \cdot v_N(0). \quad (2.23)$$

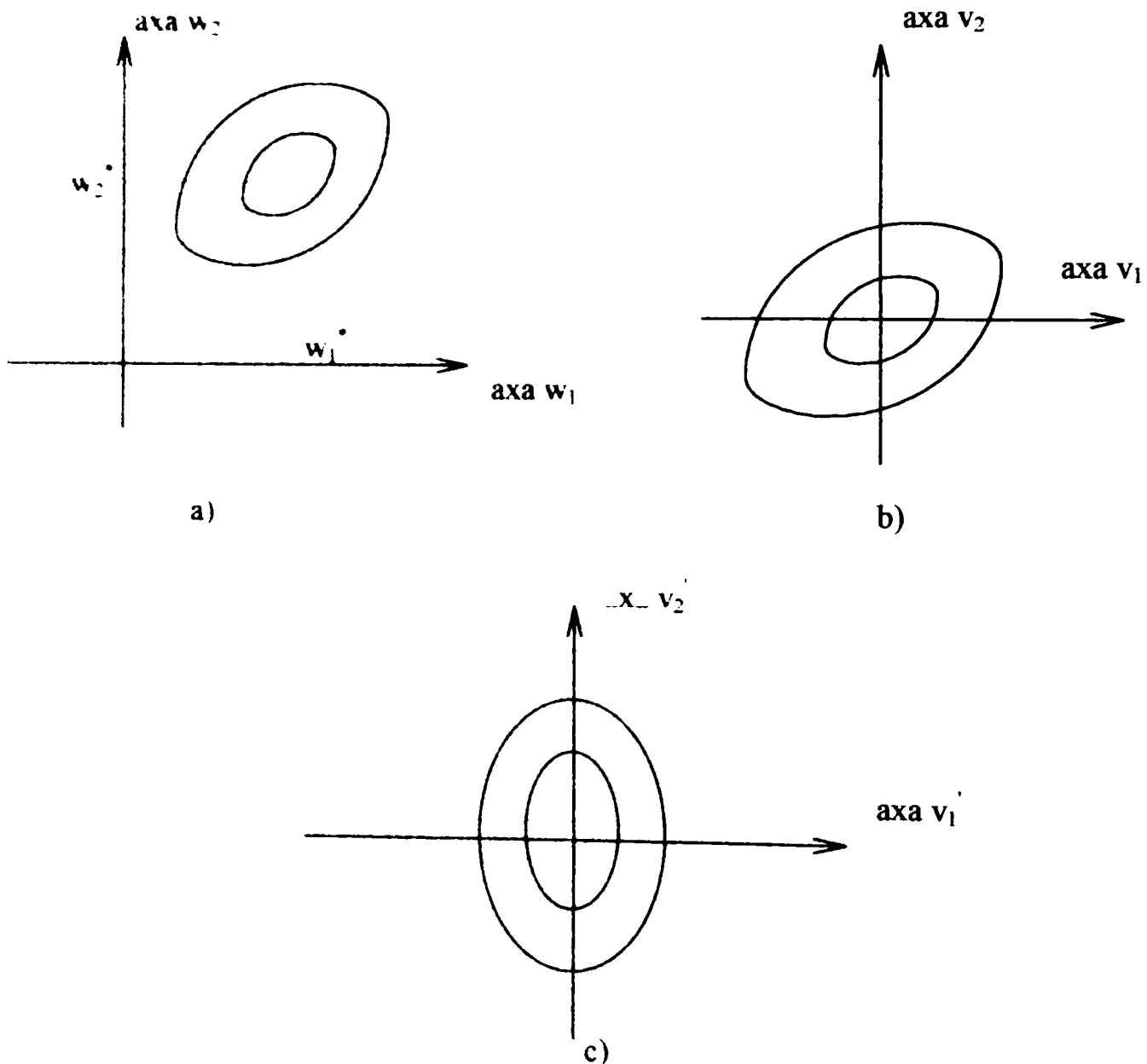


Fig. 2. 5.

Contururile suprafeței erorii medii pătratice în planele w_N , v_N și v_N' .

Pentru simplificarea în continuare a relațiilor (2.22) și (2.23) se poate efectua o rotație a axelor spațiului v_N obținându-se axele în spațiul v'_N prin înmulțirea vectorului v'_N cu o matrice A_{NN} . Această rotație a axelor spațiului v_N permite ca în spațiul v'_N intersecția suprafeței erorii medii pătratice cu planele paralele cu axele v_N să fie cea prezentată în figura 2.5 c). Relațiile care dau transformările ortogonale sunt

$$\begin{aligned} v_N &= A_{NN} \cdot v'_N \\ v'_N &= A_{NN}^T \cdot v_N \end{aligned}$$

unde A_{NN} este matricea care permite obținerea transformării ortogonale. Matricea A_{NN} se obține din matricea de autocorelație (sau din estimata sa) R_{NN} pe baza relației

$$A_{NN} = M_{NN}^T \cdot R_{NN} \cdot M_{NN},$$

unde matricea M_{NN} este o matrice a cărei coloane conțin vectorii proprii ai matricei R_{NN} , matricea M_{NN} fiind numită și matricea modală a lui R_{NN} . Matricea A_{NN} este o matrice diagonală cu valorile proprii λ_i ale lui R_{NN} .

S-a demonstrat în [1] că valorile coeficienților de predicție obținute în spațiul v'_N sunt date de relația

$$w_N^* = A_{NN}^{-1} \cdot p_N^* \quad (2.24)$$

relație în care matricea A_{NN}^{-1} este o matrice diagonală cu elementele

$$\left(A_{NN}^{-1} \right)_{ii} = \frac{1}{\lambda_i}.$$

Din relația (2.24), prin identificare, rezultă relațiile individuale

$$w_i^* = \frac{p_i^*}{\lambda_i}$$

și setul de ecuații liniar independente

$$w_i^*(n+1) = (1 - \alpha \cdot \lambda_i) w_i^*(n) + \alpha p_i^* \quad i = 0, 1, \dots, N-1.$$

În figura 2.6 s-a prezentat modul de variație al coeficienților de predicție pentru diferite valori ale parametrului de adaptare α , utilizând **programul realizat de autor** și prezentat în Anexa II.3. Se observă că **odată cu creșterea valorii parametrului α coeficienții de predicție își ating valoarea optimă după un număr mai mic de iterații.**

Așa cum s-a arătat în [1] și [2] alegerea **valorilor inițiale** ale coeficienților de predicție **nu influențează valoarea finală a acestora**, fapt ilustrat în figura 2.7. Astfel, deși valorile inițiale ale coeficienților sunt $w_1(0) = -1$, respectiv $w_2(0) = 1,5$, se observă că valorile finale atinse sunt aceleași ca în figura 2.6.

Un alt factor care influențează performanțele filtrului este **convergența**. Pentru ca setul de coeficienți să convergă este necesar să fie îndeplinită condiția

$$0 < \alpha < \frac{2}{\lambda_i} \quad i = 0, 1, \dots, N-1.$$

Deoarece această condiție trebuie îndeplinită de către toate ecuațiile sistemului, condiția generală pentru parametrul α este

$$0 < \alpha < \frac{2}{\lambda_{\max}}$$

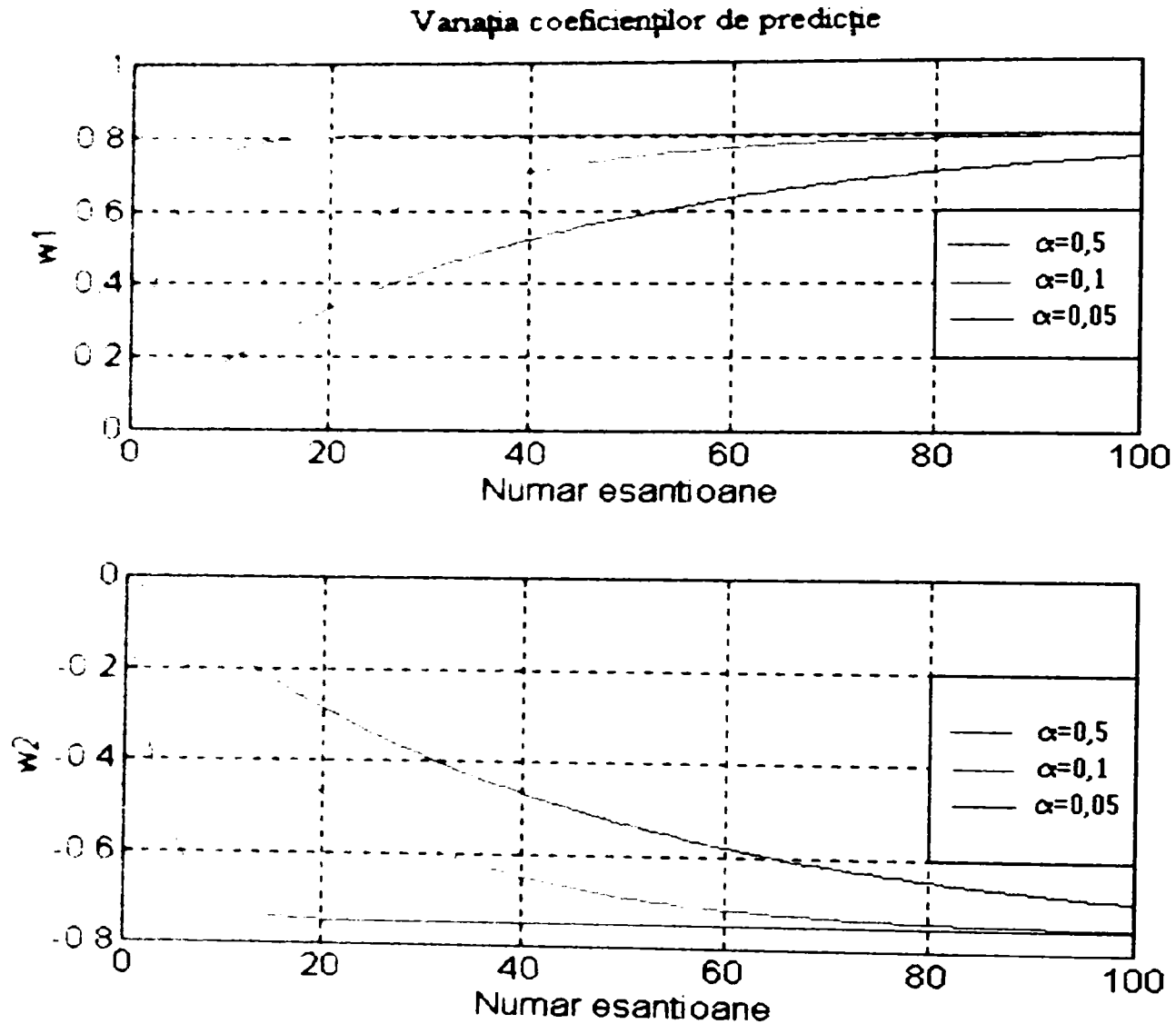


Fig. 2.6.
Variația coeficienților de predicție utilizând transformări ortogonale.

În figura 2.7 este prezentată de asemenea variația coeficienților de predicție pentru mai multe valori ale parametrului α , observând că **pentru valori mari ($\alpha = 1,175$) setul de valori prezintă oscilații și pentru valori mai mari obținându-se un șir divergent, nereprezentat în figură**

Analizând graficul din figura 2.7, se observă că se poate pune problema determinării unei valori optime a parametrului α pentru care forma de variație a coeficienților de predicție să fie similară regimului critic amortizat. **Autorul a efectuat mai multe încercări, ajungând la concluzia că această valoare optimă depinde de natura canalului și de parametrii zgomotului ce influențează canalul, fiind posibilă determinarea valorii ei prin mai multe încercări.** În exemplul prezentat această valoare este $\alpha = 0,05$.

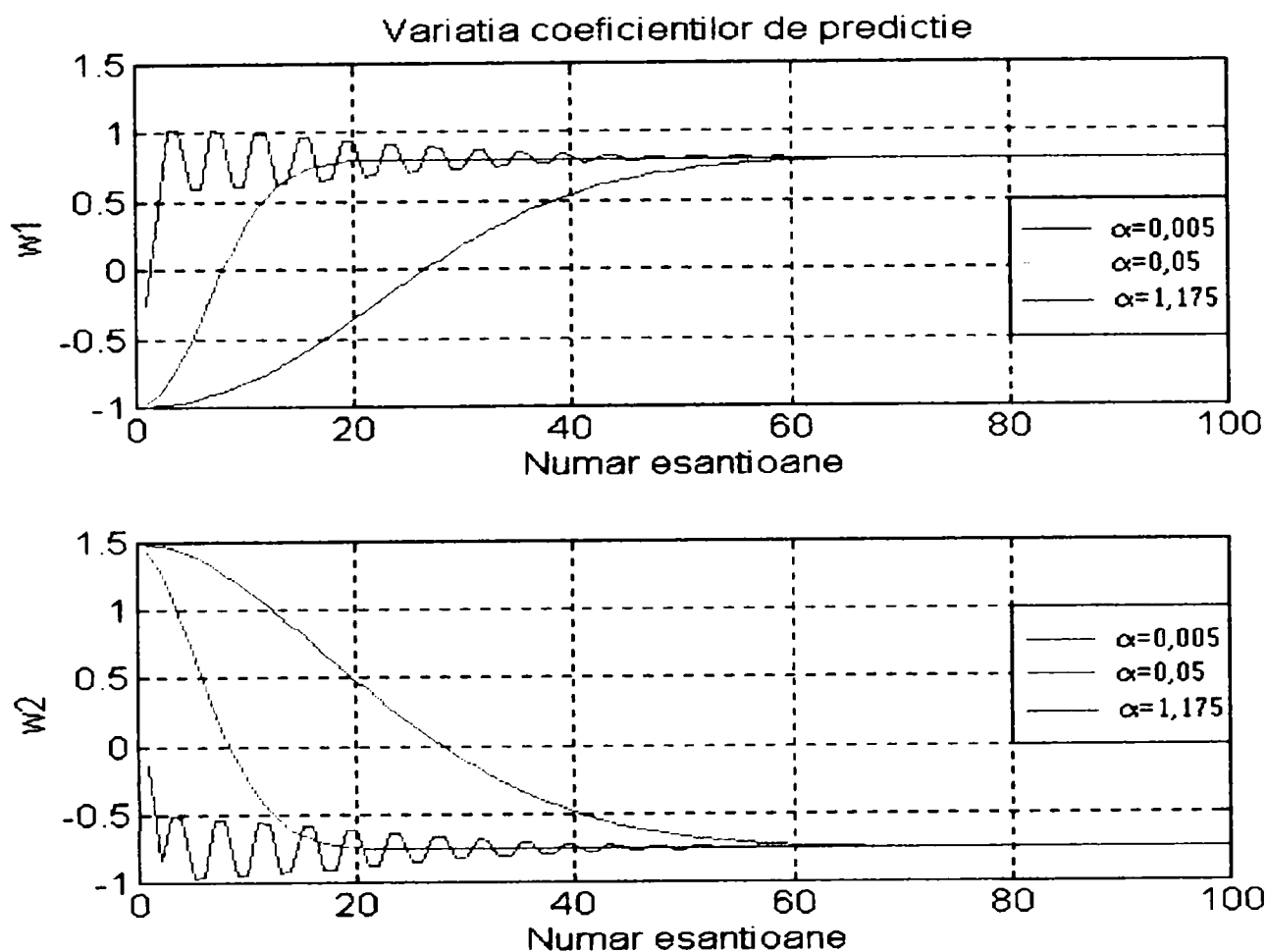


Fig. 2.7.
Influența alegerii valorilor inițiale și a parametrului de adaptare asupra coeficienților de predicție.

2.3.2. Algoritmul gradientului stocastic (LMS)

Acest algoritm, în literatura de limba engleză se întâlnește sub prescurtarea “LMS” (least mean square). Din cele prezentate la metoda pantei descendente maxime s-a văzut că pentru aplicarea metodei trebuiau cunoscute matricea de autocorelație \mathbf{R}_{NN} , respectiv vectorul de intercorelație \mathbf{p}_N . În majoritatea aplicațiilor practice matricea \mathbf{R}_{NN} și vectorul \mathbf{p}_N pot fi doar estimate din datele disponibile, dar această metodă prezintă două dezavantaje: primul constă în faptul că estimarea matricelor poate introduce erori datorită estimatorului folosit, care se pot propaga în continuare la calculul coeficienților de predicție \mathbf{w}_N , iar al doilea dezavantaj constă în faptul că reduce aria de aplicații în care se dorește prelucrarea datelor în timp real datorită volumului de calcule mare. De multe ori pot apărea ambele dezavantaje.

Algoritmul LMS, datorită simplității sale este cel mai utilizat algoritm implementat în sistemele actuale, fiind o formă simplificată obținută prin simpla aproximare a estimării funcției de intercorelație cu valoarea instantanee a produsului $e(n)x(n)$, adică

$$E\{e(n)\mathbf{x}_N(n)\} = e(n)\mathbf{x}_N(n) ,$$

care duce la relația pe care se bazează algoritmul LMS

$$\mathbf{w}_N(n+1) = \mathbf{w}_N(n) + \alpha e(n)\mathbf{x}_N(n). \quad (2.25)$$

Algoritmul LMS mai este denumit **gradient zgomotos** sau **algoritm de aproximare a gradientului** [1].[9]

Popularitatea algoritmului LMS rezidă în simplitatea sa deosebită din punct de vedere al structurii de calcul, necesități minime ale capacității de memorare precum și marea ușurință cu care poate fi analizat din punct de vedere matematic. Relația vectorială (2.25) se poate dezvolta exprimând fiecare componentă a vectorului \mathbf{w}_N sub forma

$$w_i(n+1) = w_i(n) + \alpha \cdot e(n) \cdot x(n-i) , \quad 0 \leq i \leq N-1. \quad (2.26)$$

Datorită proprietăților statistice de ortogonalitate dintre eroarea de predicție și datele filtrate metoda pantei descendente maxime este convergentă, în timp ce ecuația (2.25) impune condiții mult mai restrictive pentru ca $\mathbf{w}_N(n)$ să convergă la \mathbf{w}_N^* și anume

$$e(n) \cdot x(n-i) = 0 , \quad 0 \leq i \leq N-1.$$

Această condiție trebuie îndeplinită pentru fiecare iterație la orice moment de timp pentru ca $\mathbf{w}_N(n)$ să convergă la valorile optime \mathbf{w}_N^* , fapt care lasă impresia că algoritmul LMS este suboptimal. S-a demonstrat în [1] că în cazul convergenței, valorile instantanee ale coeficienților de predicție nu sunt exact egale cu valorile optime \mathbf{w}_N^* , dar media coeficienților de predicție converge la valorile optime.

Principala diferență între metoda pantei descendente maxime și algoritmul LMS este că prima este o metodă deterministă necesitând cunoașterea valorii exacte a gradientului, în timp ce algoritmul LMS este un algoritm stocastic recursiv utilizând doar o aproximare a gradientului

O problemă importantă constă în determinarea erorii medii pătratice suplimentare care apare datorită utilizării algoritmului LMS în locul metodei pantei descendente maxime. Considerând că în cazul algoritmului LMS valorile coeficienților de predicție $\mathbf{w}_N(n)$ fluctuează în jurul mediei lor $\bar{\mathbf{w}}_N$, care tinde într-adevăr la valorile optime \mathbf{w}_N^* , pentru a măsura diferența între eroarea obținută cu algoritmul LMS și cea cu metoda pantei descendente maxime se poate scrie că

$$\mathbf{w}_N(n) = \bar{\mathbf{w}}_N(n) + \mathbf{z}_N(n), \quad (2.27)$$

unde $\mathbf{z}_N(n)$ reprezintă o componentă stocastică.

Pentru a calcula eroarea medie pătratică în cazul algoritmului LMS se pornește de la relația

$$\varepsilon(n) = E\{e^2(n)\} = E\left\{\left[d(n) - \mathbf{w}_N^T(n) \cdot \mathbf{x}_N(n)\right]^2\right\},$$

care dezvoltată permite obținerea relației

$$\varepsilon(n) = E\{d^2(n)\} - 2E\{\mathbf{w}_N^T(n) \mathbf{x}_N(n) d(n)\} + E\{\mathbf{w}_N^T(n) \mathbf{x}_N(n) \mathbf{x}_N^T(n) \mathbf{w}_N(n)\}. \quad (2.28)$$

Introducând (2.27) în (2.28) se obține

$$\begin{aligned} \varepsilon(n) = & E\{d^2(n)\} - 2E\{\tilde{\mathbf{w}}_N^T(n) \mathbf{x}_N(n) d(n)\} + E\{\tilde{\mathbf{w}}_N^T(n) \mathbf{x}_N(n) \mathbf{x}_N^T(n) \tilde{\mathbf{w}}_N(n)\} \\ & - 2E\{\mathbf{z}_N^T(n) \mathbf{x}_N(n) d(n)\} + E\{\mathbf{z}_N^T(n) \mathbf{x}_N(n) \mathbf{x}_N^T(n) \tilde{\mathbf{w}}_N(n)\} + \\ & + E\{\tilde{\mathbf{w}}_N^T(n) \mathbf{x}_N(n) \mathbf{x}_N^T(n) \mathbf{z}_N(n)\} + E\{\mathbf{z}_N^T(n) \mathbf{x}_N(n) \mathbf{x}_N^T(n) \mathbf{z}_N(n)\}. \end{aligned} \quad (2.29)$$

Considerând că $\tilde{\mathbf{w}}_N$ tinde la \mathbf{w}_N^* pentru n suficient de mare și utilizând rezultatele de la metoda pantei descendente maxime se poate scrie că

$$\begin{aligned} E\{d^2(n)\} - 2E\{\tilde{\mathbf{w}}_N^T(n) \mathbf{x}_N(n) d(n)\} + E\{\tilde{\mathbf{w}}_N^T(n) \mathbf{x}_N(n) \mathbf{x}_N^T(n) \tilde{\mathbf{w}}_N(n)\} = \\ = \varepsilon_{\min} + \mathbf{v}_N^T(n) \mathbf{A}_{NN} \mathbf{v}_N'(n). \end{aligned}$$

Pentru obținerea unei expresii a erorii medii pătratice mai simple se pot face următoarele presupuneri simplificatoare, îndeplinite în practică de cele mai multe ori:

1° Semnalul de date $x(n)$ și coeficienții de predicție rezultați în urma algoritmului LMS $w_i(n)$ sunt necorelate.

2° Media componentelor vectorului de zgomot $z_i(n)$ este nulă

$$E\{z_i(n)\} = 0.$$

3° Componentele zgomotului sunt necorelate cu dispersia σ_z^2 , adică

$$E\{z_i(n) z_j(n)\} = \sigma_z^2 \delta(i - j),$$

unde $\delta(k)$ este semnalul discret impuls unitar.

4° Zgomotul este necorelat în timp

$$E\{z_i(n - n_1) z_i(n - n_2)\} = 0, \quad n_1 \neq n_2$$

5° Zgomotul este necorelat cu coeficienții de predicție LMS

$$E\{z_i(n) \tilde{w}_j(n)\} = E\{z_i(n)\} \cdot E\{\tilde{w}_j(n)\} = 0.$$

6° Zgomotul este necorelat cu semnalul de date

$$E\{z_i(n) x(n)\} = E\{z_i(n)\} \cdot E\{x(n)\} = 0.$$

Aceste presupuneri duc la concluzia că termenii 4,5 și 6 din relația erorii medii pentru algoritmul LMS sunt nuli. S-a demonstrat în [1] că ultimul termen al relației (2.29) este dat de relația

$$E\{\mathbf{z}_N^T(n) \mathbf{x}_N(n) \mathbf{x}_N^T(n) \mathbf{z}_N(n)\} = \sigma_x^2 \cdot \text{Tr}\{E\{\mathbf{z}_N(n) \mathbf{z}_N^T(n)\}\},$$

unde

$$\text{Tr}\left\{E\left\{\mathbf{z}_N(n)\mathbf{z}_N^T(n)\right\}\right\} \cong \sum_{i=1}^N \frac{\alpha \cdot \varepsilon_{\min}}{(2 - \alpha \cdot \lambda_i)} .$$

Înlocuind rezultatele parțiale obținute în (2.29) se obține relația simplificată a erorii medii pătratice în cazul utilizării algoritmului LMS

$$\varepsilon(n) = \varepsilon_{\min} + \mathbf{v}_N^T \cdot \mathbf{A}_{NN} \cdot \mathbf{v}_N(n) + \sigma_N^2 \cdot \alpha \cdot \varepsilon_{\min} \cdot \sum_{i=1}^N \frac{1}{2 - \alpha \cdot \lambda_i} , \quad (2.30)$$

în care ultimul termen reprezintă eroarea suplimentară introdusă de algoritmul LMS și poartă denumirea de zgomot suplimentar sau eroare în exces. Se observă că zgomotul suplimentar depinde de date precum și de parametrii filtrului.

În figura 2.8 este prezentat modul de variație a coeficienților filtrului și a erorii medii pătratice în cazul utilizării algoritmului LMS pentru setul de date obținut la ieșirea sistemului implementat în paragrafele precedente. **Programul realizat de autor este prezentat în Anexa B.4, graficele fiind obținute în urma medierii statistice a 200 de evenimente, fiecare eveniment conținând un număr de 1000 de eșantioane ale semnalului de ieșire. Timpul necesar rulării programului pe un calculator Pentium /100 MHz este de 173 secunde.**

Se observă că odată cu creșterea valorii parametrului α coeficienții își ating mai rapid valorile optime Wiener, w_N^* , iar eroarea medie pătratică este de asemenea mai rapid descrescătoare spre o aceeași valoare care depinde de valorile parametrului α .

În comparație cu metoda evaluării directe a ecuației cu diferențe (figura 2.4) sau a utilizării transformărilor ortogonale (figura 2.6) se observă că **algoritmul LMS este mult mai lent convergent însă valorile spre care tind coeficienții sunt aceleași ca și în cazul celorlalte două metode.**

Ca și în cazul metodelor precedente, s-a arătat în [1],[9] că valorile inițiale ale coeficienților filtrului nu influențează valorile finale atinse, coeficienții filtrului tinzând spre aceleași valori optime, ele influențând doar numărul de eșantioane după care aceste valori sunt atinse.

În [4] s-a arătat că algoritmul LMS este convergent doar dacă este îndeplinită condiția

$$0 < \alpha < \frac{1}{N \cdot E\{x^2(n)\}} , \quad (2.31)$$

unde N reprezintă numărul de coeficienți ai filtrului iar $E\{\}$ reprezintă operatorul de estimare statistică a mediei.

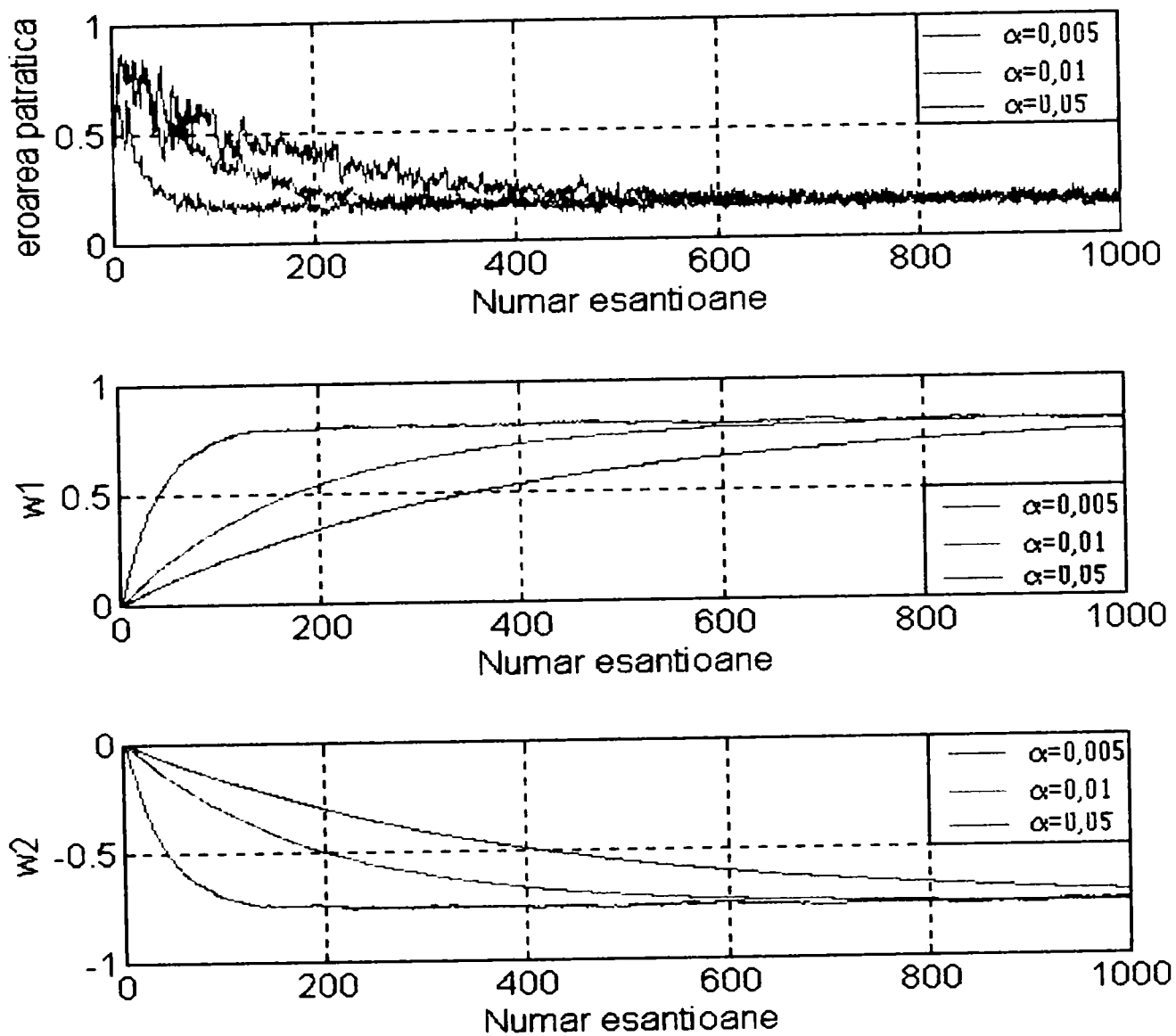


Fig. 2.8.

Variația erorii medii pătratice și a coeficienților filtrului în cazul utilizării algoritmului LMS.

2.3.3. Algoritmul LMS cu pas variabil.

În relația (2.25), în mod uzual, se notează $\alpha = 2 \mu$, unde μ reprezintă un parametru pozitiv a cărui alegere influențează atât convergența cât și eroarea algoritmului. S-a demonstrat în [13], [14] că o valoare mare a lui μ implică o convergență rapidă dar însoțită de erori mari și reciproc. Domeniul în care poate lua valori μ este dat de relația

$$0 < \mu < \frac{\lambda_{med}}{\lambda_{max}}$$

unde λ_i reprezintă valorile proprii ale matricei de autocorelație $\hat{\mathbf{R}}_{NN}(n)$.

Din cele prezentate rezultă că este eficientă utilizarea unui parametru μ variabil, astfel încât prin modificarea valorii sale să se poată realiza atât o convergență bună (pentru valori mai mari), cât și o eroare mică (pentru valori mai mici).

În acest fel, algoritmul LMS cu pas variabil poate fi descris de relația

$$\mathbf{w}_N(n+1) = \mathbf{w}_N(n) + 2 \cdot \mathbf{M}_N(n) \cdot e(n) \cdot \mathbf{x}_N(n),$$

unde $\mathbf{M}_N(n)$ este o matrice diagonală conținând valorile parametrului μ , de forma

$$\mathbf{M}_N(n) = \begin{bmatrix} \mu_1(n) & 0 & \dots & 0 \\ 0 & \mu_2(n) & \dots & 0 \\ & & \dots & \dots \\ 0 & 0 & \dots & \mu_N(n) \end{bmatrix}.$$

Există mai multe metode pentru a alege forma de variație a parametrului μ . În [13] s-au prezentat două metode de alegere a parametrului $\mu_i(n)$ pe baza erorii obținute $e(n) \cdot x_i(n)$.

Ambele metode se bazează pe scăderea parametrului $\mu_i(n)$ dacă apare un număr specificat m_0 de schimbări de semn a produsului $e(n) \cdot x_i(n)$, respectiv creșterea parametrului $\mu_i(n)$ dacă un număr specificat m_1 de valori consecutive a produsului $e(n) \cdot x_i(n)$ nu-și schimbă semnul. Se observă că este implicat numai bitul de semn al produsului $e(n) \cdot x_i(n)$. Parametrii m_0 și m_1 sunt denumiți "număr limită a schimbărilor de semn consecutive", respectiv "număr limită a valorilor cu același semn".

Domeniul în care se poate modifica parametrul μ este cuprins între o valoare minimă μ_{\min} determinată de erorile de calcul, respectiv o valoare maximă μ_{\max} care trebuie să fie în interiorul domeniului pentru care algoritmul LMS converge.

a) prima metodă, numită metoda VS, permite creșterea parametrului μ printr-o operație de multiplicare cu un factor $\alpha' > 1$, respectiv scăderea printr-o operație de împărțire cu același factor potrivit relațiilor

$$\mu_i(n) = \alpha' \cdot \mu_i(n-1),$$

sau

$$\mu_i(n) = \mu_i(n-1) / \alpha'$$

În implementările uzuale α' s-a ales de obicei egal cu 2, astfel încât modificarea parametrului μ se poate face printr-o simplă operație de deplasare.

În figura 2.9. este prezentată forma de variație a erorii medii pătratice precum și a coeficienților filtrului adaptiv în cazul utilizării metodei VS pentru un parametru de modificare constant a pasului și diferite valori inițiale ale pasului μ .

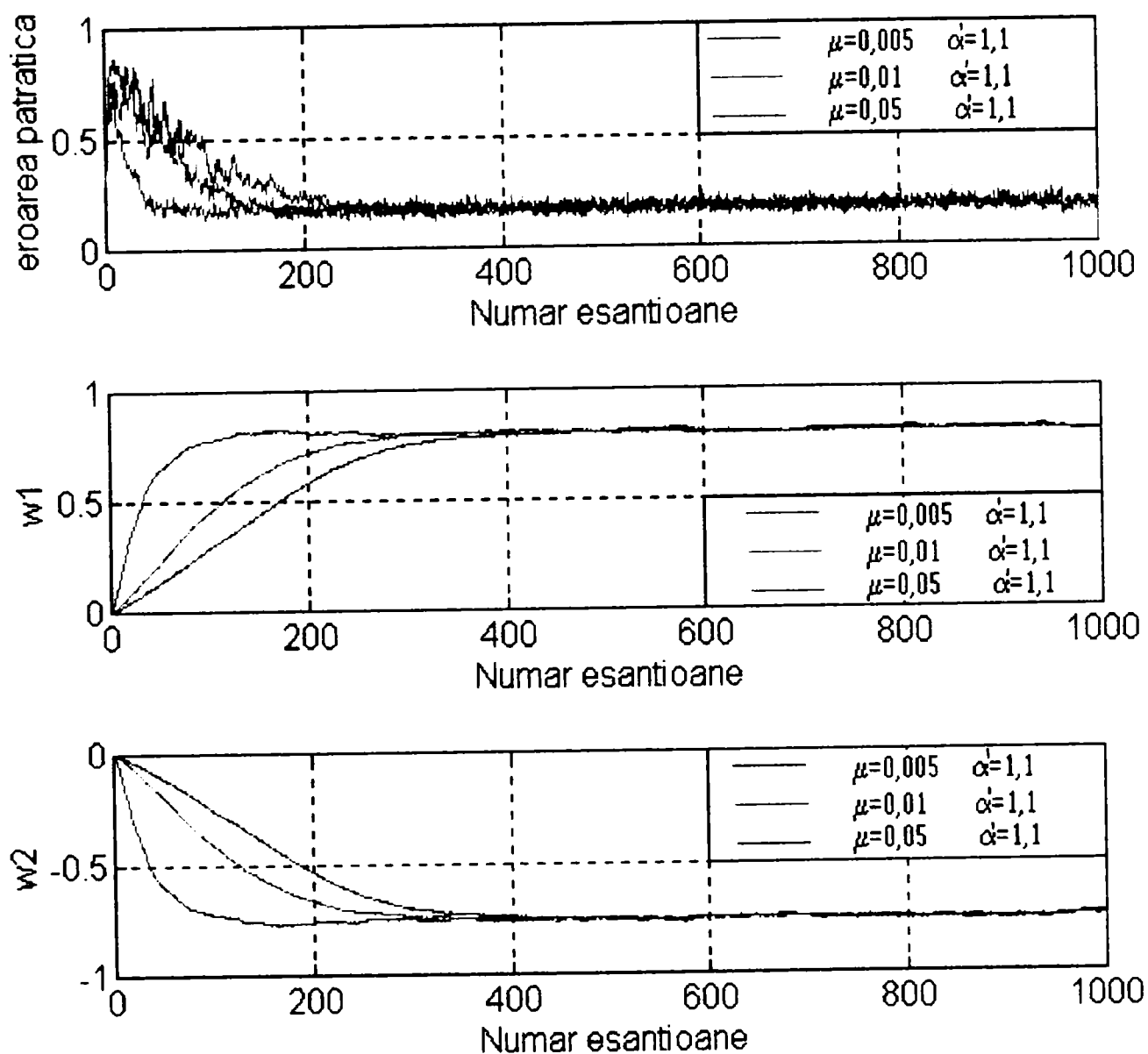


Fig. 2.9. Variația erorii medii pătratice și a coeficienților în cazul metodei VS pentru factor de modificare a pasului constant.

Se observă că algoritmul este cu atât mai convergent cu cât μ este mai mare în sensul că eroarea medie pătratică ajunge mai repede la valoarea minimă iar coeficienții își ating mai repede valorile optime.

Programul realizat de autor care permite obținerea setului de valori a erorii medii pătratice și a coeficienților filtrului este prezentat în Anexa II.5. Graficele, în mod similar cu cele din figura 2.8 au fost obținute în urma medierii statistice a 200 de evenimente, fiecare eveniment conținând un număr de 1000 de eșantioane. **Timpul necesar rulării** pe un calculator Pentium la 100 MHz **a fost de 272 secunde**, mai mare datorită operațiunilor de comparare suplimentare care apar în plus față de utilizarea algoritmului LMS.

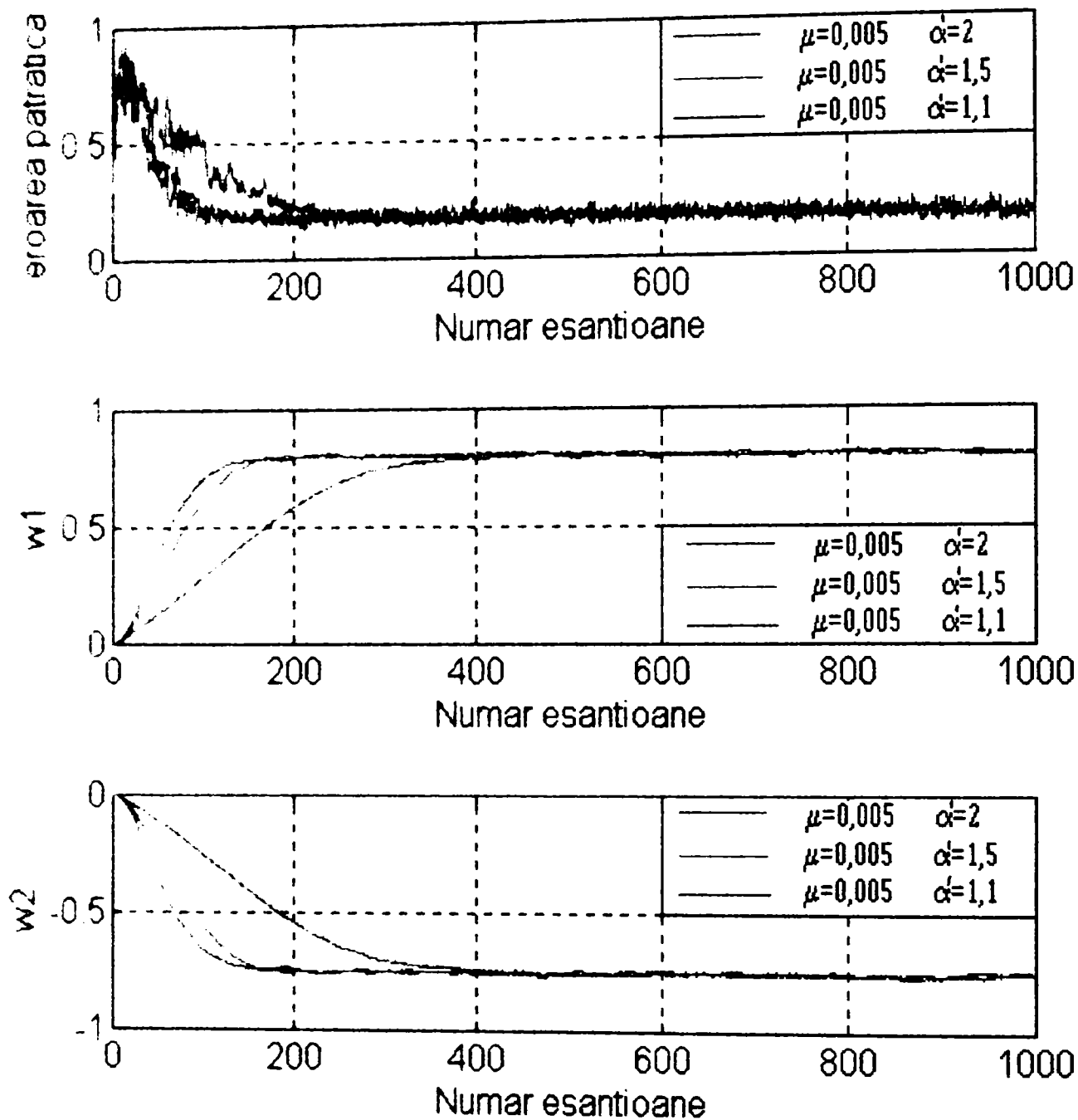


Fig.2.10. Variația erorii medii pătratice și a coeficienților în cazul metodei VS pentru factor de modificare a pasului variabil

Alegerea unui parametru variabil de modificare a pasului permite obținerea graficelor prezentate în figura 2.10., în care este de asemenea prezentată forma de variație a erorii medii pătratice precum și a coeficienților filtrului pentru metoda VS.

Se observă că **pentru o valoare inițială constantă a pasului μ , algoritmul este mai rapid convergent odată cu creșterea factorului de modificare a pasului α' , graficele fiind de asemenea obținute în urma medierii statistice a 200 de evenimente.**

b) A doua metodă, numită și metoda VSA , permite creșterea parametrului μ printr-o operație de adunare a unei constante $\alpha' > 0$, respectiv o descreștere prin scăderea aceleiași constante , potrivit relațiilor

$$\mu_i(n) = \mu_i(n-1) + \alpha'$$

sau

$$\mu_i(n) = \mu_i(n-1) - \alpha'$$

Deși această metodă este mai greu de implementat decât metoda VS, prezentată pentru $\alpha = 2$, creșterea complexității este compensată de o mai mare flexibilitate.

În figura 2.11. este prezentată forma de variație a erorii medii pătratice precum și a coeficienților filtrului în cazul utilizării metodei VSA.

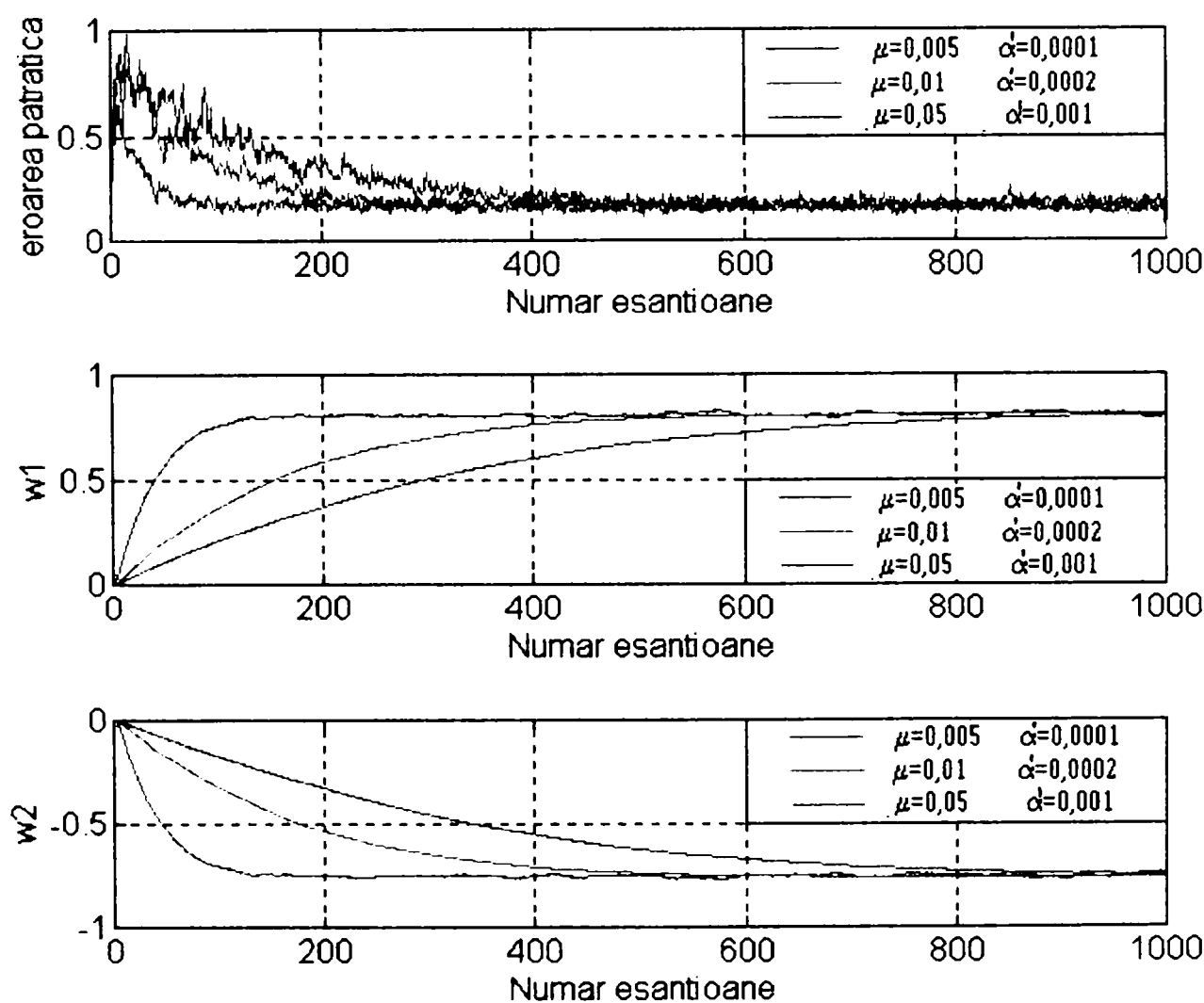


Fig.2.11. Variația erorii medii pătratice și a coeficienților în cazul metodei VSA.

O diagramă care permite implementarea atât a metodei VS cât și a metodei VSA este prezentată în figura 2.12. Parametrul r_0 reprezintă indicatorul de schimbări de semn, iar r_1 este numărul iterațiilor în care nu se schimbă semnul.

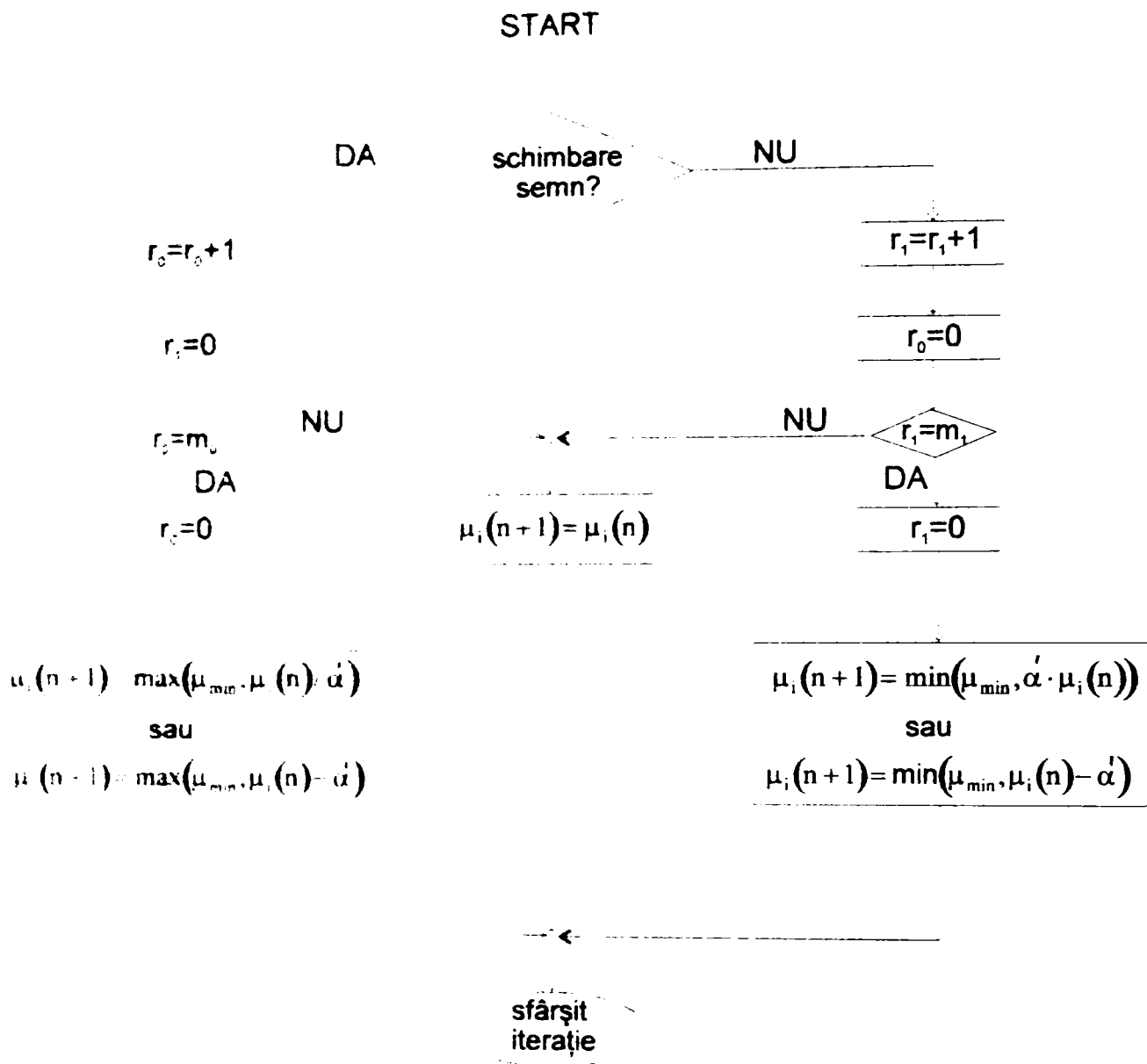


Fig. 2.12. Ordinograma metodelor VS și VSA.

c) A treia metodă de modificare a parametrului μ a fost prezentată în [15] și pornește de la relația de recurență

$$\mathbf{w}_N(n+1) = \mathbf{w}_N(n) - \mu \cdot \mathbf{R}_{NN}^{-1}(n) \cdot \nabla(n),$$

unde

$$\nabla(n) = 2 \cdot e(n) \cdot \mathbf{x}_N(n)$$

reprezintă estimatorul vectorului gradient, iar $\mathbf{R}_{NN}^{-1}(n)$ reprezintă estimata inversei matricei de autocorelație a semnalului de intrare.

În cazul utilizării unui coeficient μ variabil, s-a demonstrat că algoritmul de determinare a coeficientului μ la momentul n este dat de

$$\mu(n) = b \cdot \alpha(n)$$

unde b este o constantă fixă pozitivă cu valoare aleatoare iar $\alpha(n)$ este dat de relația

$$\alpha(n) = \frac{1}{1 - \tau(n) + 2 \cdot \tau(n) \cdot b}$$

unde

$$\tau(n) = \mathbf{x}_N^T(n) \cdot \hat{\mathbf{R}}_{NN}^{-1}(n) \cdot \mathbf{x}_N(n)$$

d) A patra metodă de modificare a pasului prezentată în [14] reactualizează pasul atât în funcție de valoarea precedentă cât și în funcție de pătratul erorii obținute la pasul precedent pe baza relației

$$\mu(n+1) = \alpha \cdot \mu(n) + \gamma \cdot e^2(n)$$

unde $0 < \alpha < 1$ iar $\gamma > 0$.

Valoarea lui α trebuie să fie subunitară pentru a introduce un factor de uitare. Se observă că dacă $\gamma = 0$ se poate obține algoritmul VS alegând $\alpha' = \frac{1}{\alpha}$. În mod uzual $\alpha = 0,97$, iar $\gamma = 5 \cdot 10^{-4}$.

O problemă importantă în alegerea unui algoritm o reprezintă volumul de calcule caracterizat prin numărul de operații (înmulțiri și adunări) necesare pentru parcurgerea unui pas. Aceasta se repercutează asupra timpului de lucru, de multe ori dorindu-se reducerea pe cât posibil a acestuia.

Unele aplicații ale filtrelor adaptive necesită un răspuns la impuls cu un spectru larg, astfel că filtrele utilizate necesită utilizarea unui mare număr de coeficienți. De exemplu, eliminarea ecoului în acustică necesită filtre cu un număr de 4000 de coeficienți. Acest fapt face să apară necesitatea reducerii volumului de calcule în cazul utilizării algoritmului LMS. Pentru un set de N coeficienți, obținerea a două valori succesive ale semnalului de la ieșirea filtrului necesită un număr de $4N$ operații de înmulțire și $4N$ operații de adunare.

Notând:

$$\begin{aligned} \mathbf{A}_0 &= [x(n) \quad x(n-2) \quad \dots \quad x(n-N+2)] \\ \mathbf{A}_1 &= [x(n-1) \quad x(n-3) \quad \dots \quad x(n-N+1)] \\ \mathbf{A}_2 &= [x(n-2) \quad x(n-4) \quad \dots \quad x(n-N)] \\ \mathbf{W}_0(n-1) &= [w_0 \quad w_1 \quad \dots \quad w_{N-2}]^T(n-1) \\ \mathbf{W}_1(n-1) &= [w_1 \quad w_2 \quad \dots \quad w_{N-1}]^T(n-1) \end{aligned} ,$$

s-a demonstrat în [7] că un algoritm echivalent algoritmului LMS (considerând N coeficienți) este dat de relațiile

$$\begin{bmatrix} e(n-1) \\ e(n) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -s(n) & 1 \end{bmatrix} \left\{ \begin{bmatrix} y(n-1) \\ y(n) \end{bmatrix} - \begin{bmatrix} \mathbf{A}_1(\mathbf{w}_0 + \mathbf{w}_1) + (\mathbf{A}_2 - \mathbf{A}_1)\mathbf{w}_1 \\ \mathbf{A}_1(\mathbf{w}_0 + \mathbf{w}_1) - (\mathbf{A}_1 - \mathbf{A}_0)\mathbf{w}_0 \end{bmatrix} (n-1) \right\}$$

$$\begin{bmatrix} \mathbf{w}_0 \\ \mathbf{w}_1 \end{bmatrix} (n+1) = \begin{bmatrix} \mathbf{w}_0 \\ \mathbf{w}_1 \end{bmatrix} (n-1) + \alpha \cdot \begin{bmatrix} \mathbf{A}_1^t(e(n-1) + e(n)) - (\mathbf{A}_1 - \mathbf{A}_0)^t e(n) \\ \mathbf{A}_1^t(e(n-1) + e(n)) + (\mathbf{A}_2 - \mathbf{A}_1)^t e(n-1) \end{bmatrix}$$

unde $s(n)$ poate fi dedus recursiv potrivit relației

$$s(n) = s(n-2) + \alpha [x(n-1)(x(n-1) + x(n-2)) - x(n-N-1)(x(n-N) + x(n-L-2))]$$

Acest algoritim denumit **BLMS** permite reducerea numărului de operații de înmulțire la $3N+2$ în timp ce crește la $4N+8$ numărul de operații de adunare, dar prezintă dezavantajul unei mai slabe convergențe față de algoritmul **LMS**.

2.4. Concluzii

În acest capitol au fost prezentați algoritmi adaptivi bazați pe metoda pantei descendente maxime. Pentru aplicarea metodei trebuiau cunoscute din datele de la intrarea filtrului adaptiv matricele de autocorelație \mathbf{R}_{NN} , respectiv vectorul de intercorelație \mathbf{p}_N . În aplicațiile practice acestea sunt estimate din datele de la intrare. Acest fapt prezintă două dezavantaje, introducerea erorilor datorită estimatorului folosit și timp de lucru mare datorită volumului mare de calcule, însă au avantajul unei convergențe foarte bune a coeficienților spre valorile optime.

Reducerea volumului de calcule se poate face utilizând algoritmul **LMS** care însă nu prezintă o la fel de bună convergență.

Comparând figurile 2.4, respectiv 2.8 se observă că în timp ce valorile optime ale coeficienților filtrului sunt atinse după câteva zeci de eșantioane, în cazul algoritmului **LMS** sunt necesare chiar sute de eșantioane. Din acest motiv utilizarea algoritmilor cu pas variabil permite o mai bună convergență a algoritmului **LMS**, fiind studiați algoritmi **VS** și **VSA**.

În figura 2.13. se poate observa că algoritmul **VS** ($\alpha=2$) are proprietățile cele mai bune în ceea ce privește convergența și eroarea medie pătratică, în timp ce algoritmul **LMS** are proprietăți mai slabe.

În ceea ce privește timpul de lucru necesar unei iterații s-a arătat că utilizarea algoritmului **LMS** este mai eficientă deoarece nu necesită operații de comparare suplimentare, deci din acest punct de vedere utilizarea lui este mai eficientă. Rulările au fost făcute prin medierea aceluiași număr de evenimente, cuprinzând același număr de eșantioane.

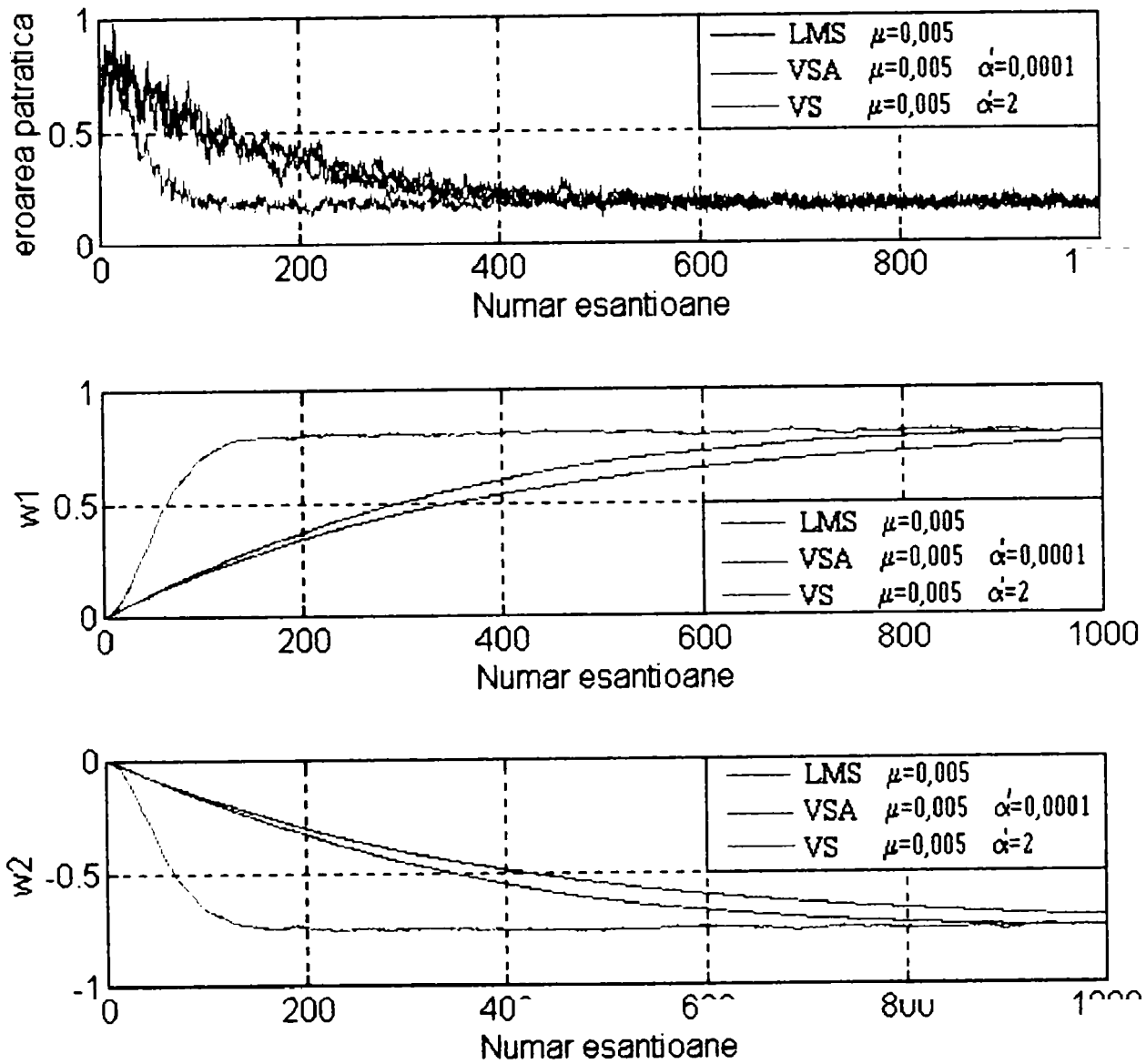


Fig.2.13. Variația erorii pătratice medii și a coeficienților filtrului la utilizarea algoritmilor LMS,VS,VSA.

3. IMPLEMENTAREA FILTRELOR ADAPTIVE IIR UTILIZÂND STRUCTURI LATTICE

Capitolul de față prezintă în prima parte o clasificare a principalelor structuri de filtre lattice (zăbrea) corelată cu un studiu comparativ asupra performanțelor filtrelor adaptive ce utilizează aceste structuri

Deoarece forma directă de implementare a filtrelor IIR prezintă unele dezavantaje este arătat în continuare cum poate fi utilizată structura lattice pentru eliminarea acestora.

În incheiere autorul prezintă posibilitățile de implementare a filtrelor adaptive IIR utilizând diferite metode.

3.1. Tipuri de structuri de filtre lattice fără reacție

Așa cum s-a prezentat în [1] și [2], multe metode și algoritmi adaptivi au dus la structuri ale filtrelor adaptive care utilizează celula elementară de tip lattice.

3.1.1. Structura bazată pe algoritmul lui Durbin

În [1] s-a prezentat un alt algoritm de rezolvare a ecuației matriciale (2.4), care permite determinarea recursivă a coeficienților de predicție, algoritm denumit algoritmul lui Durbin. Utilizând acest algoritm, relațiile de recurență pentru erorile de predicție *înainte* și *înapoi* sunt date de relațiile

$$\begin{aligned} e_p^f(n) &= e_{p-1}^f(n) - k_p \cdot e_{p-1}^b(n-1) \\ e_p^b(n) &= e_{p-1}^b(n-1) - k_p \cdot e_{p-1}^f(n), \end{aligned} \quad (3.1)$$

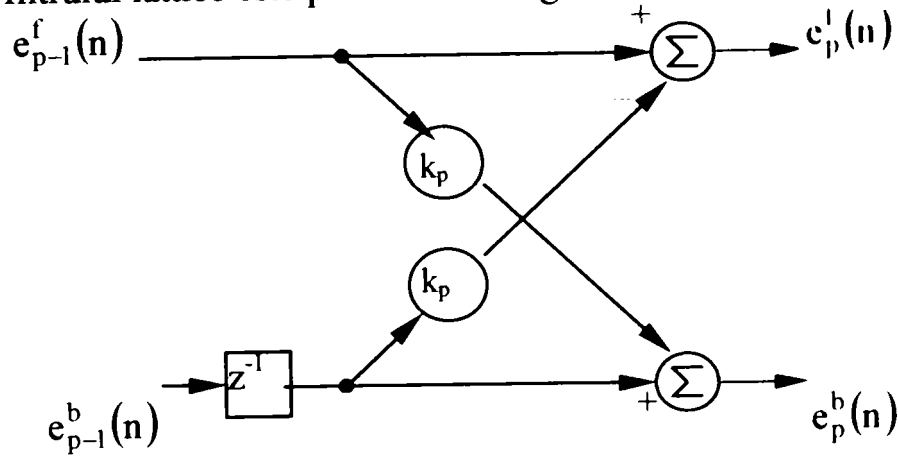
unde coeficienții k_p , denumiți coeficienți de reflexie, se deduc recursiv potrivit relațiilor

$$k_p = \frac{1}{\varepsilon_{p-1}} \left[\varphi_x(p) - \sum_{j=1}^{p-1} w_j^{(p-1)} \cdot \varphi_x(p-j) \right] \quad (3.2)$$

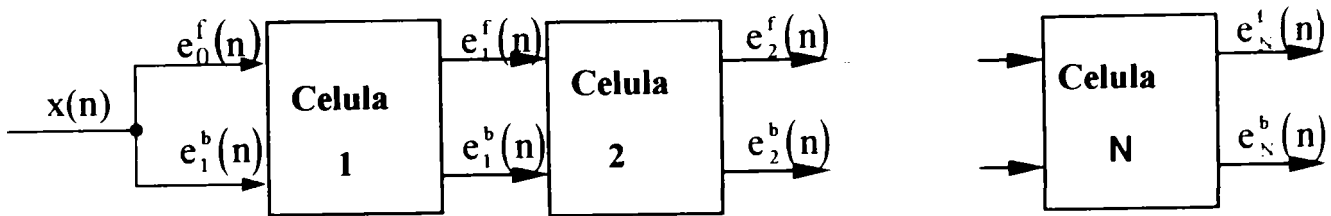
$$\varepsilon_p = (1 - k_p^2) \cdot \varepsilon_{p-1},$$

în care ε_{p-1} reprezintă eroarea minimă de predicție obținută la pasul $p-1$, $\varphi_x(i)$ reprezintă valorile funcției de autocorelație, iar $w_j^{(p-1)}$ reprezintă coeficienții optimi la pasul $p-1$, iar k_p reprezintă coeficienții de reflexie optimi la pasul p .

Structura filtrului lattice este prezentată în figura 3.2.



a)



b)

Fig. 3.1. Implementarea filtrului lattice: a) Structura unei celule a filtrului lattice; b) Interconectarea în cascadă a N celule pentru realizarea filtrului de predicție .

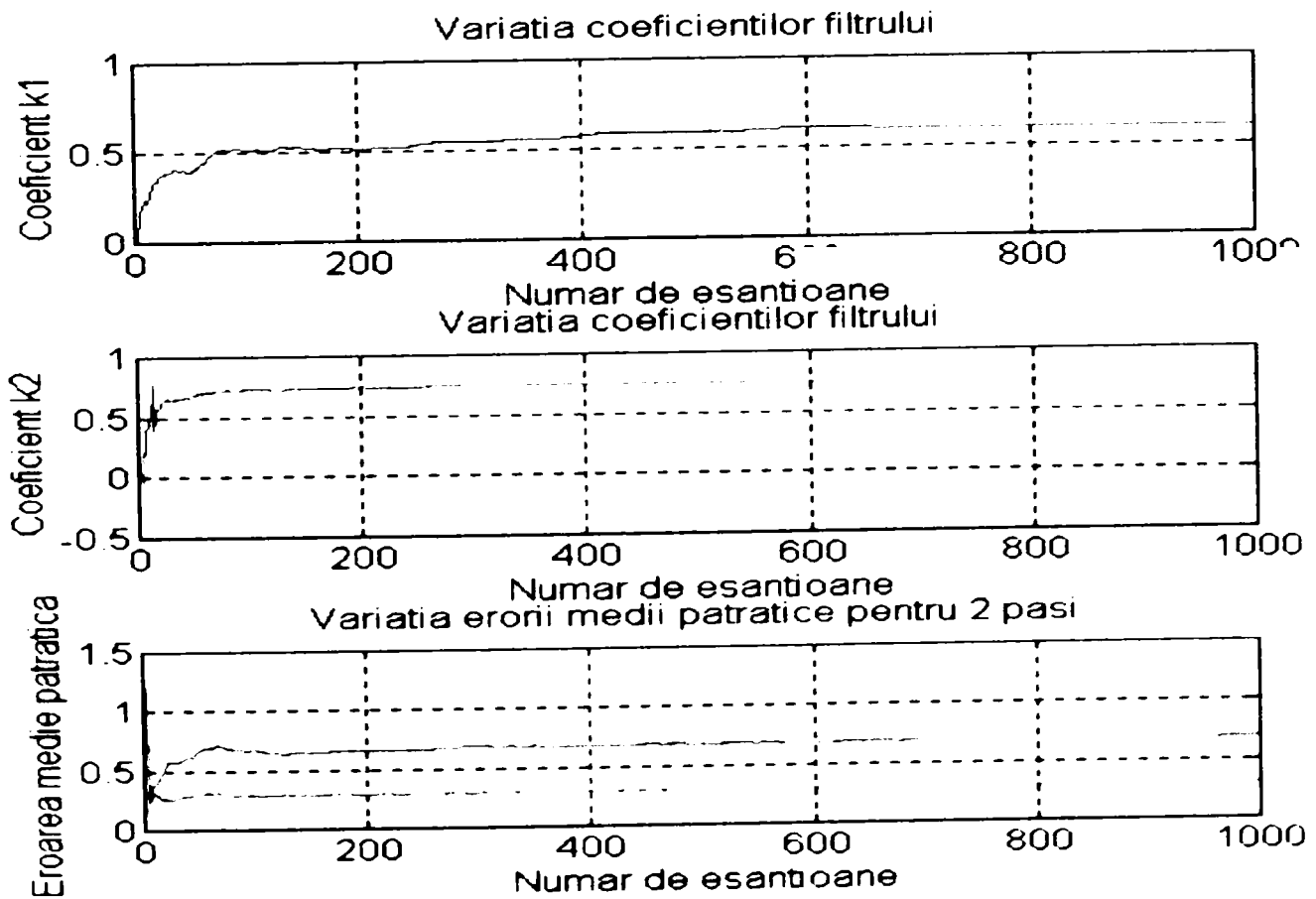


Fig. 3.2. Variația coeficienților de reflexie și a erorilor medii pătratice în cazul utilizării a două celule în cascadă.

Pe baza relațiilor prezentate algoritmul de determinare a coeficienților de predicție pentru un predictor cu N coeficienți este următorul:

1. Se calculează valorile funcțiilor de autocorelație $\varphi_x(0), \dots, \varphi_x(N-1)$, utilizând estimatorul prezentat în capitolul 2.

2. Se stabilește valoarea inițială a erorii medii pătratice $\varepsilon_0 = \varphi_x(0)$.

3. Pentru $1 \leq p \leq N$ se repetă

$$k_p = \frac{1}{\varepsilon_{p-1}} \left[\varphi_x(p) - \sum_{j=1}^{p-1} w_j^{(p-1)} \cdot \varphi_x(p-j) \right]$$

$$w_p^{(p)} = k_p$$

Pentru $1 \leq j \leq p-1$ se repetă

$$w_j^{(p)} = w_j^{(p-1)} - k_p \cdot w_{p-j}^{(p-1)}$$

$$\varepsilon_p = (1 - k_p^2) \cdot \varepsilon_{p-1}$$

la sfârșitul celei de-a N-a iterații obținându-se coeficienții $w_j^{(N)}$ ai filtrului linear predictiv optimal.

Odată cu creșterea numărului de celule din structura filtrului lattice, eroarea medie pătratică scade tinzând spre o valoare minimă, în timp ce volumul de calcule necesar implementării filtrului crește. Din acest motiv alegerea numărului N de coeficienți ai predictorului, deci și a numărului de celule conectate în cascadă, este foarte importantă. Dacă numărul de coeficienți utilizați este mai mic decât cel optim volumul de calcule necesar este mai mic dar eroarea obținută este mai mare. O creștere prea mare a numărului de coeficienți nu aduce o îmbunătățire în performanțele filtrului în ceea ce privește eroarea medie pătratică, în timp ce volumul de calcule crește.

Autorul a verificat experimental acest lucru utilizând modelul canalului influențat de zgomot prezentat în capitolul precedent în care, așa cum s-a arătat, numărul de coeficienți de predicție necesar era 2. În cele ce urmează se prezintă acest experiment.

Semnalul de la ieșirea sistemului, $x(n)$, reprezenta un proces autoregresiv de ordin 2 descris de ecuația diferențială

$$x(n) + c_1 \cdot x(n-1) + c_2 \cdot x(n-2) = v_1(n), \quad (3.3)$$

unde $c_1 = -0,15$; $c_2 = -0,75$.

Valorile optime ale coeficienților de reflexie se obțin prin rezolvarea sistemului de ecuații

$$\begin{aligned} k_1 \cdot [1 - k_2] &= -c_1 \\ k_2 &= -c_2 \end{aligned} \quad (3.4)$$

Rezolvând sistemul de ecuații (3.4) se obține $k_2=0,75$ și $k_1=0,6$.

Utilizând programul prezentat în Anexa III.1 **autorul a obținut graficul prezentat în figura 3.2 în care se observă** forma de variație a coeficienților de reflexie k_1 și k_2 , precum și **faptul că eroarea medie pătratică după a doua celulă este mai mică decât cea după prima celulă.**

În figura 3.3 este reprezentată forma de variație a coeficienților de reflexie în cazul utilizării unui filtru lattice cu trei celule precum și variația erorii medii pătratice.

Programul realizat de autor este prezentat în Anexa III.2. Se observă că valoarea coeficientului de reflexie k_3 este foarte mică în comparație cu a celorlalți doi fapt ce verifică faptul că rolul celei de a treia celule este practic foarte mic. Totodată, eroarea medie pătratică corespunzătoare celei de a treia celule este practic suprapusă peste a celei de a doua, fapt ce arată că performanțele filtrului în ceea ce privește reducerea erorii medii pătratice nu s-au îmbunătățit prin introducerea unei celule suplimentare. Pe de altă parte, timpul de calcul necesar a crescut, fapt ilustrat și din rezultatele prezentate centralizat în Tabelul 3.1.

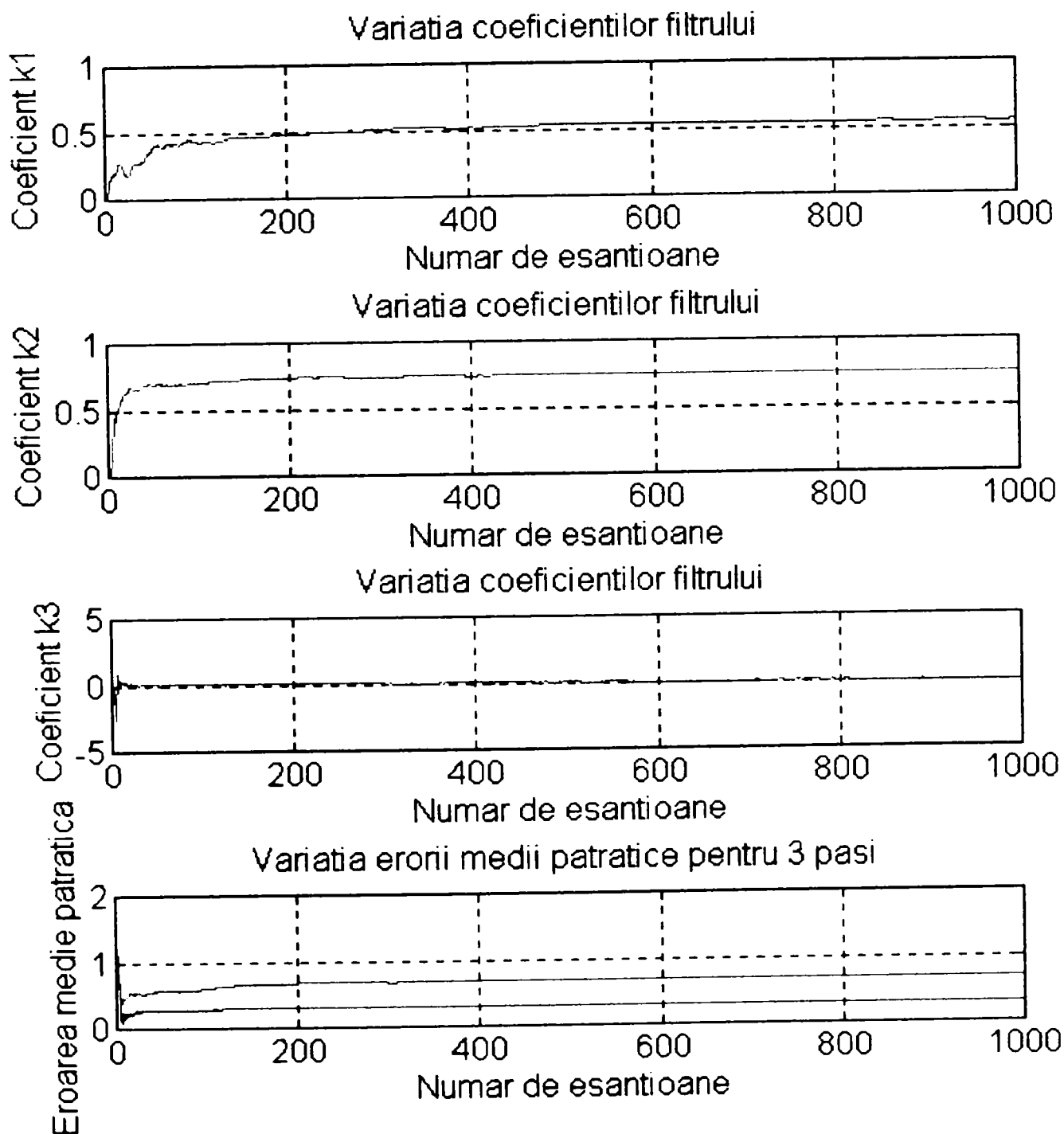


Fig. 3.3. Variația coeficienților de reflexie și a erorilor medii pătratice în cazul utilizării a trei celule în cascadă.

Tabelul 3.1.

Valorile medii ale coeficienților de reflexie, ale erorilor medii pătratice și timpul de calcul necesar implementării filtrului lattice.

	K_1	K_2	K_3	ϵ_1	ϵ_2	ϵ_3	Timp [s]
Filtru cu 2 celule	0,5041	0,6488	-	0,8087	0,4561	-	323,08
Filtru cu 3 celule	0,5056	0,6432	0,0440	0,8159	0,4254	0,4489	492,29

3.1.2. Structura bazată pe metoda gradientului adaptiv

A doua variantă de implementare a filtrului lattice s-a obținut utilizând metoda bazată pe gradientul adaptiv, prezentată în [1], metodă care a condus la aceleași relații de recurență date de ecuațiile (3.1) pentru erorile de predicție, în timp ce coeficienții de reflexie se calculează recursiv pe baza relației

$$k_p(n+1) = k_p(n) + \beta \cdot [e_p^f(n) \cdot e_{p-1}^b(n-1) + e_p^b(n) \cdot e_{p-1}^f(n)], \quad (3.5)$$

unde β reprezintă un parametru de adaptare al coeficienților de reflexie. Structura unei celule a filtrului lattice este cea prezentată în figura 3.1.

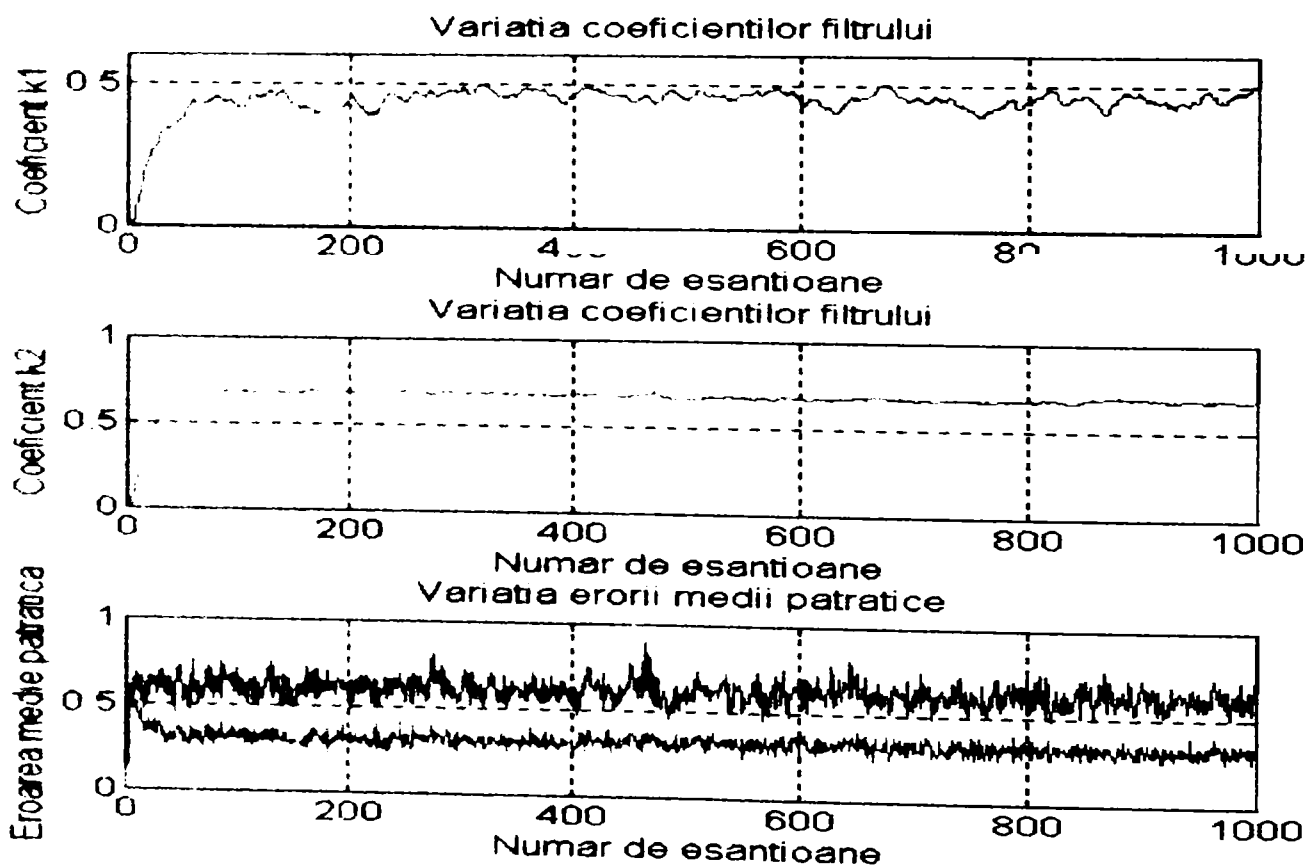


Fig. 3.4. Variația coeficienților de reflexie și a erorii medii pătratice în cazul utilizării metodei gradientului adaptiv.

Programul realizat de autor și prezentat în Anexa III.3 a permis obținerea formei de variație a coeficienților de reflexie precum și a erorii medii pătratice, prezentată în figura 3.4, pentru o valoare a parametrului de adaptare $\beta=0,05$.

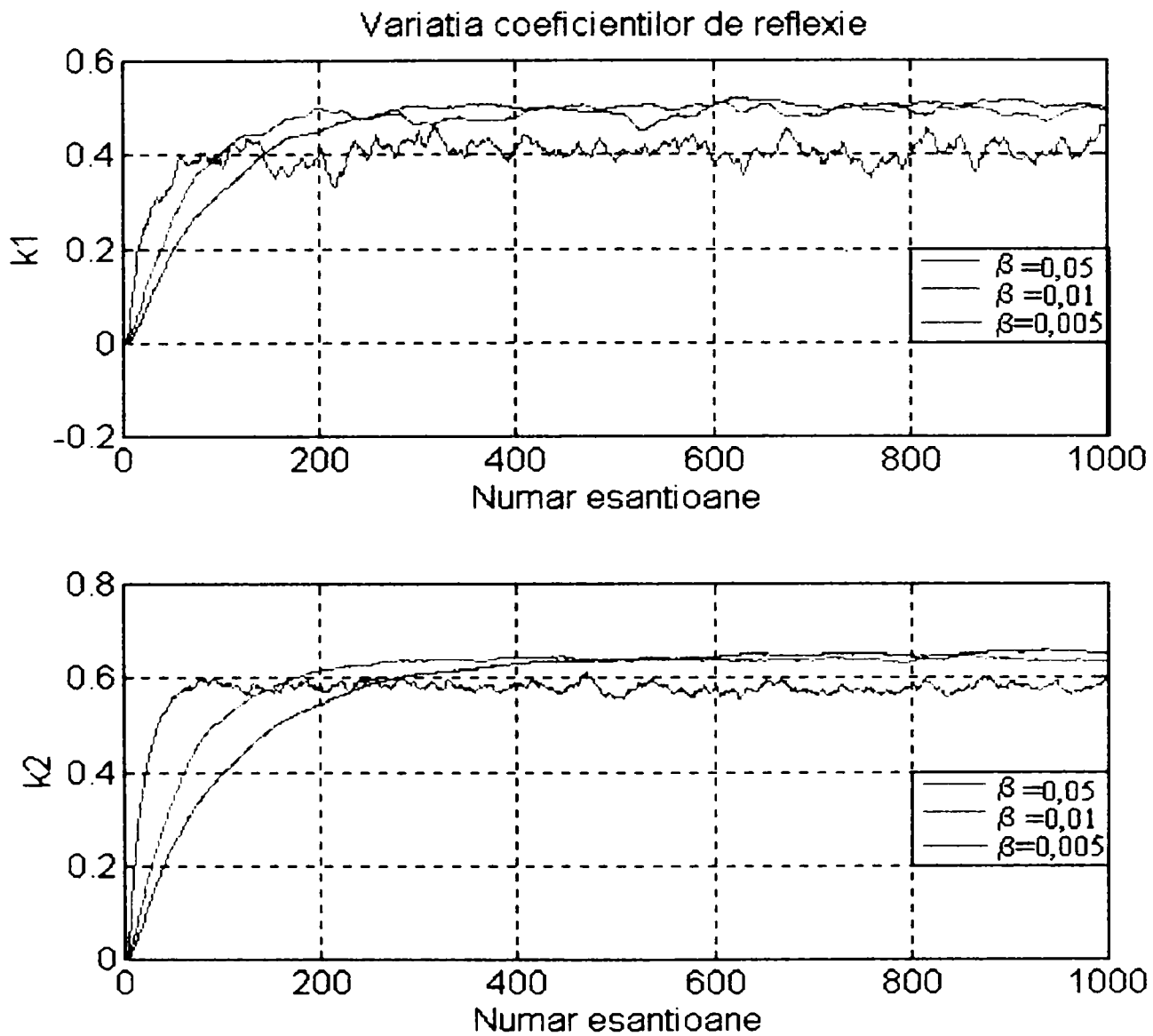


Fig. 3.5. Variația coeficienților de reflexie în funcție de parametrul de adaptare.

În funcție de valoarea parametrului de adaptare β coeficienții de reflexie își ating mai rapid (pentru valori mai mari ale lui β) sau mai lent (pentru valori mai mici ale lui β) valorile optime fapt ilustrat în figura 3.5.

Se observă, de asemenea, că la valori mari ale parametrului de adaptare coeficienții de reflexie au oscilații mai mari în jurul valorilor optime în timp ce la valori mici oscilațiile sunt mult reduse.

Din punct de vedere al timpului de calcul, în timp ce implementarea algoritmului precedent necesita un timp de calcul de 323,08 secunde pentru un filtru cu două celule și medierea statistică a 200 de evenimente, acest algoritm, în condiții similare, necesită un timp de calcul de 220 secunde.

Pentru a analiza influența zgomotelor prezente la ieșirea canalului asupra variației valorilor coeficienților filtrului, în figura 3.6 este reprezentată variația coeficienților de reflexie în funcție de dispersia zgomotului σ_z^2 , pentru $\beta = 0,05$. Se observă că **valorile optime ale coeficienților de reflexie variază odată cu modificarea dispersiei zgomotului** aceasta explicându-se prin faptul că, așa cum s-a arătat în capitolul precedent, valorile optime ale coeficienților depind de matricea de autocorelație care depinde la rândul ei de dispersia zgomotului

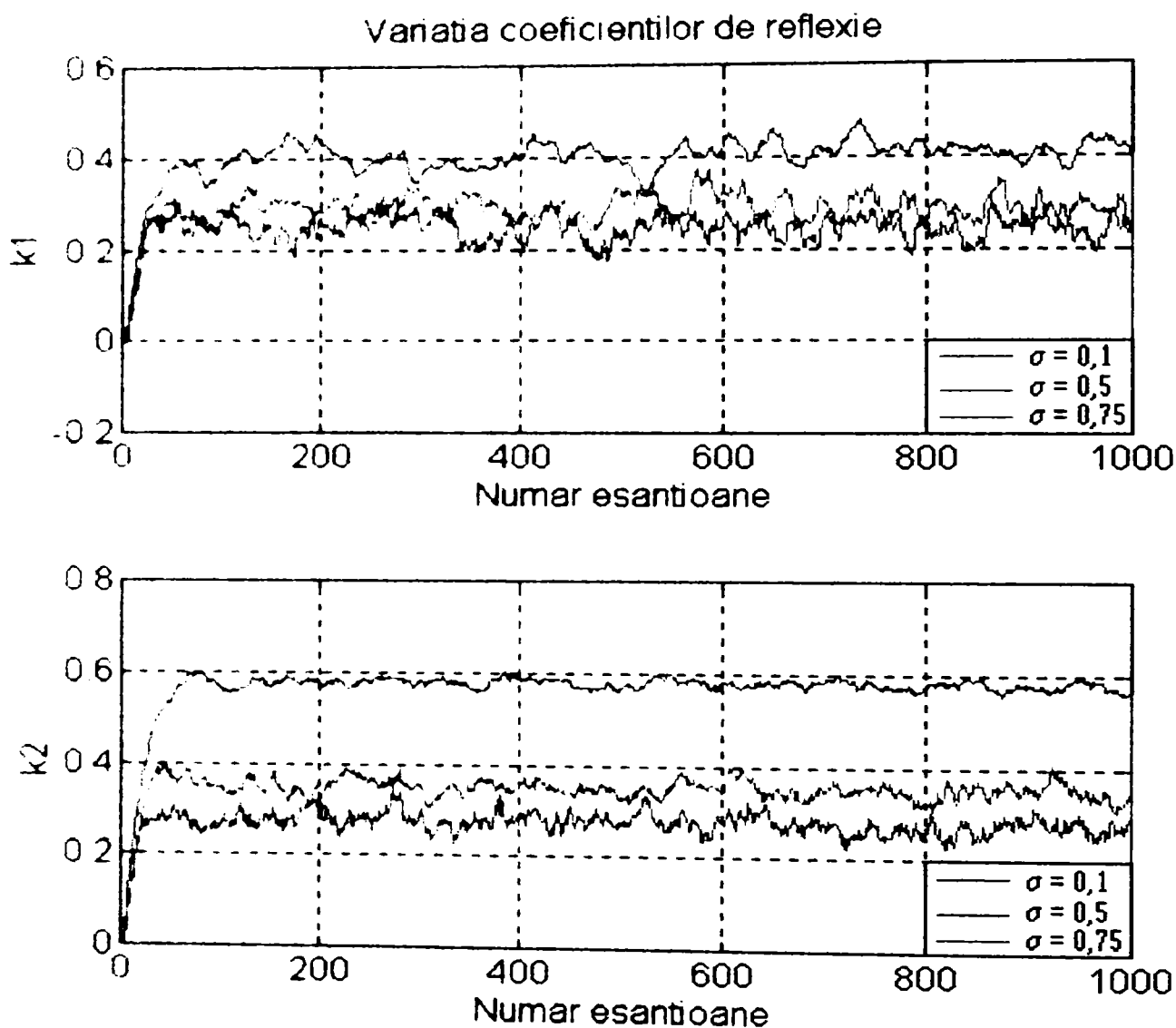


Fig. 3.6. Variația coeficienților de reflexie în funcție de dispersia zgomotului.

Se observă că odată cu creșterea dispersiei zgomotului cresc și oscilațiile coeficienților de reflexie, în special ale lui k_2 . Aceasta se explică prin faptul că valorile coeficientului k_2 se determină recursiv, depinzând de valorile erorilor de la ieșirea primei celule. **Autorul a constatat experimental că la valori mai mari ale dispersiei zgomotului coeficienții filtrului nu mai converg la valorile optime, rezultând că nu orice valoare a parametrului de adaptare β asigură convergența coeficienților filtrului indiferent de zgomotul prezent, ci că valoarea parametrului de adaptare trebuie corelată cu zgomotul.**

3.1.3. Structura bazată pe metoda gradientului adaptiv normalizat

S-a arătat în [1] și [23] că prin normalizarea coeficientului de adaptare β se evită problema instabilității filtrelor lattice. Metoda presupune înlocuirea lui β cu $\frac{1}{d_p(n)}$, unde $d_p(n)$ se deduce recursiv pe baza relației

$$d_p(n) = (1 - \alpha) \cdot d_p(n-1) + [e_p^f(n)]^2 + [e_p^b(n)]^2, \quad (3.6)$$

unde $\alpha \ll 1$ este o constantă.

Coeficienții de reflexie se determină recursiv pe baza relației

$$k_p(n+1) = k_p(n) + \frac{1}{d_p(n)} \cdot [e_p^f(n) \cdot e_{p-1}^b(n-1) + e_p^b(n) \cdot e_{p-1}^f(n)]. \quad (3.7)$$

Programul realizat de autor și prezentat în Anexa III.4. permite obținerea reprezentării formei de variație a coeficienților de reflexie pentru două valori ale constantei α , precum și în situația neutilizării normalizării coeficientului de adaptare β , din figura 3.7. Se observă că odată cu scăderea valorii constantei α , valorile coeficienților filtrului prezintă oscilații mai mici, filtrul devenind mai stabil.

Din punct de vedere al timpului de lucru, în aceleași condiții de mediere a 200 de evenimente, acesta a crescut la 283 secunde, fapt explicabil datorită necesității reactualizării parametrului $d_p(n)$.

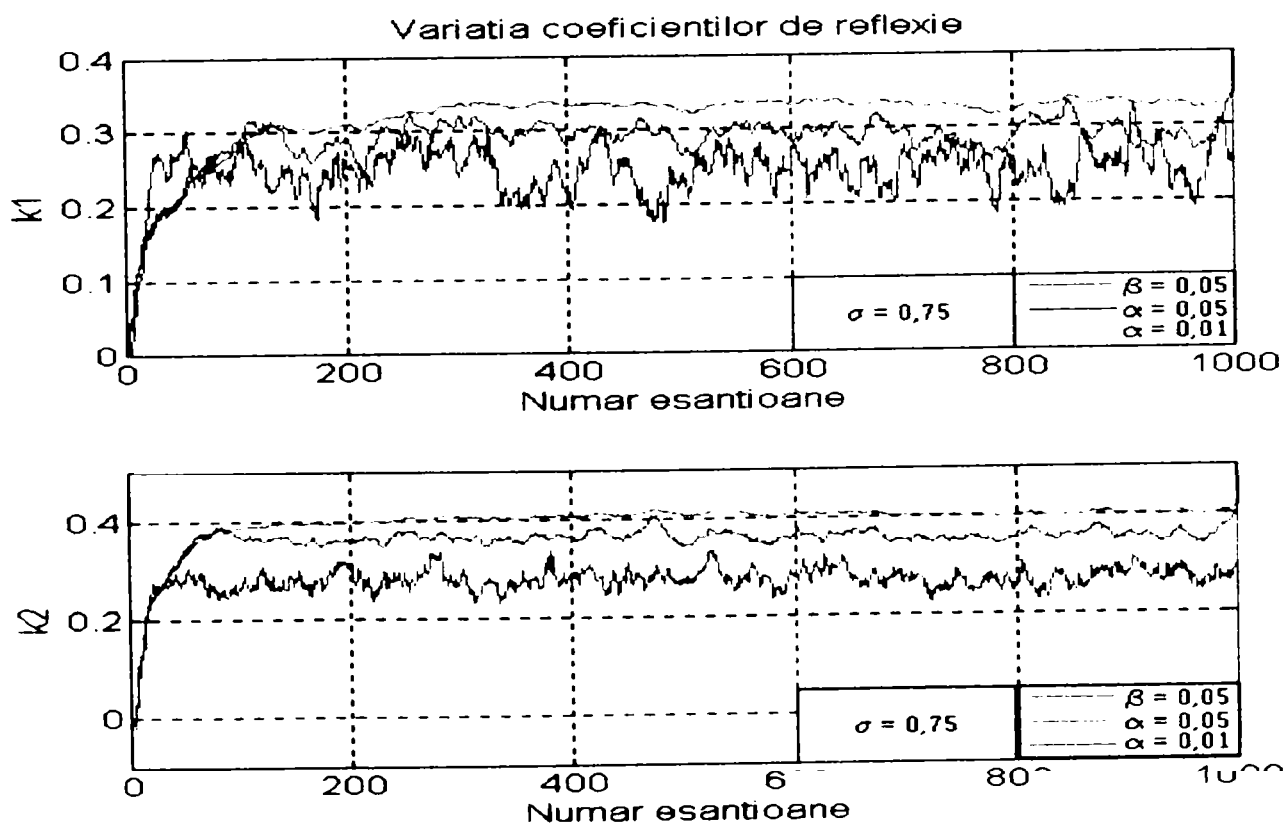


Fig. 3.7. Variația coeficienților de reflexie utilizând metoda gradientului adaptiv normalizat.

3.1.4. Structura bazată pe utilizarea conceptului de spații vectoriale

Această metodă de implementare a unui filtru lattice prezentată în [1] permite obținerea structurii din figura 3.8 în care se observă că valorile coeficienților de reflexie “înainte” și “înapoi” sunt diferite.

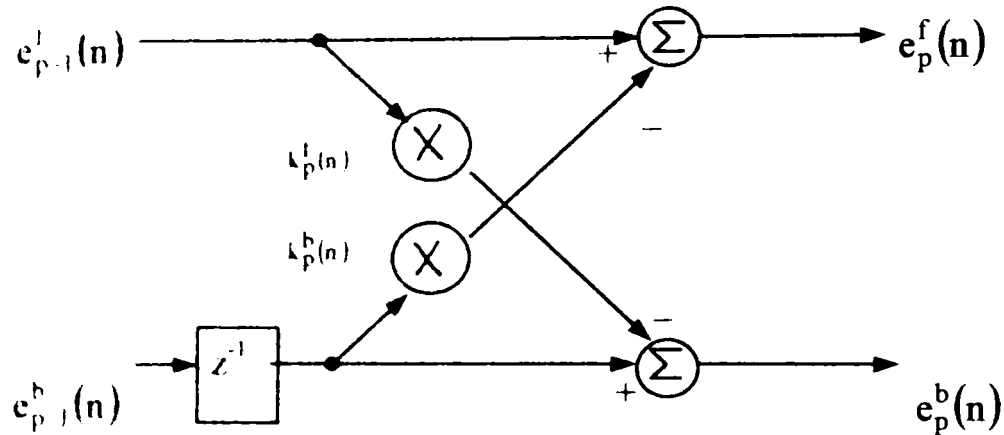


Fig. 3.8. Structura unei celule a filtrului lattice bazată pe utilizarea spațiilor vectoriale.

Algoritmul de implementare a structurii filtrului lattice prezentat în [1] este

1. Inițializare

$$e_{p-1}^b(0) = \Delta_{p-1}(0) = 0,$$

$$\gamma_{p-1}(0) = 1,$$

$$\varepsilon_{p-1}^f(0) = \varepsilon_{p-1}^b(0) = \delta.$$

2. For $n = 1$ to n final do:

$$e_0^b(n) = e_0^f(n) = x(n),$$

$$\varepsilon_0^b(n) = \varepsilon_0^f(n) = \varepsilon_0^f(n-1) + x^2(n),$$

$$\gamma_0(n) = 1.$$

For $1 \leq p \leq N$ do:

$$\Delta_p(n) = \Delta_p(n-1) + \frac{e_{p-1}^f(n) \cdot e_{p-1}^b(n-1)}{\gamma_{p-1}(n-1)},$$

$$e_p^f(n) = e_{p-1}^f(n) - \frac{\Delta_p(n) \cdot e_{p-1}^b(n-1)}{\varepsilon_{p-1}^b(n-1)},$$

$$e_p^b(n) = e_{p-1}^b(n-1) - \frac{\Delta_p(n) \cdot e_{p-1}^f(n)}{\varepsilon_{p-1}^f(n)}, \quad (3.8)$$

$$\varepsilon_p^f(n) = \varepsilon_{p-1}^f(n) - \frac{\Delta_p^2(n)}{\varepsilon_{p-1}^b(n-1)},$$

$$\varepsilon_p^b(n) = \varepsilon_{p-1}^b(n-1) - \frac{\Delta_p^2(n)}{\varepsilon_{p-1}^f(n)},$$

$$\gamma_p(n-1) = \gamma_{p-1}(n-1) - \frac{[\varepsilon_{p-1}^b(n-1)]^2}{\varepsilon_{p-1}^b(n-1)},$$

în care s-a notat:

- δ valoarea inițială a erorii reziduale de predicție,
- ε_p^f și ε_p^b erorile reziduale de predicție “înainte” și “înapoi”;
- Δ_p coeficientul de corelație parțial (PARCOR);
- γ_p parametrul unghiular.

Pe baza acestor mărimi se pot defini expresiile coeficienților de reflexie “înainte” și “înapoi” conform relațiilor

$$k_p^f(n) = \frac{\Delta_p(n)}{\varepsilon_{p-1}^f(n)},$$

$$k_p^b(n) = \frac{\Delta_p(n)}{\varepsilon_{p-1}^b(n-1)}.$$
(3.9)

Pe baza relațiilor (3.8) și (3.9) autorul a realizat programul prezentat în Anexa III.5 care permite determinarea formei de variație a coeficienților de reflexie “înainte” și “înapoi”, precum și a erorii de predicție obținută după fiecare celulă, reprezentată în figura 3.9.

Se observă că ambii coeficienți de reflexie “înainte” își ating valorile optime mai repede decât coeficienții de reflexie “înapoi”.

Viteza de convergență a algoritmului depinde de valoarea inițială aleasă pentru eroarea reziduală de predicție, δ , graficul fiind obținut pentru o valoarea inițială $\delta=10$.

Alegerea valorii inițiale a erorii reziduale de predicție nu influențează valoarea finală a coeficienților de reflexie “înainte” și “înapoi”, ci doar numărul de iterații după care această valoare este atinsă. Acest fapt este ilustrat în figura 3.10. în care este reprezentată forma de variație a coeficienților de reflexie pentru diferite valori inițiale ale erorii reziduale de predicție.

Se observă că o valoare mare a erorii reziduale de predicție implică o valoare redusă a vitezei de convergență a valorilor coeficienților de reflexie ai filtrului și reciproc.

Graficele prezentate în figurile 3.9 și 3.10 au fost obținute în urma medierii statistice a 200 de evenimente, fiecare eveniment conținând un număr de 1000 de eșantioane. Timpul necesar rulării pe un calculator Pentium la 100 MHz a fost 498 secunde, explicabil prin complexitatea relațiilor pe baza cărora a fost implementat filtrul.

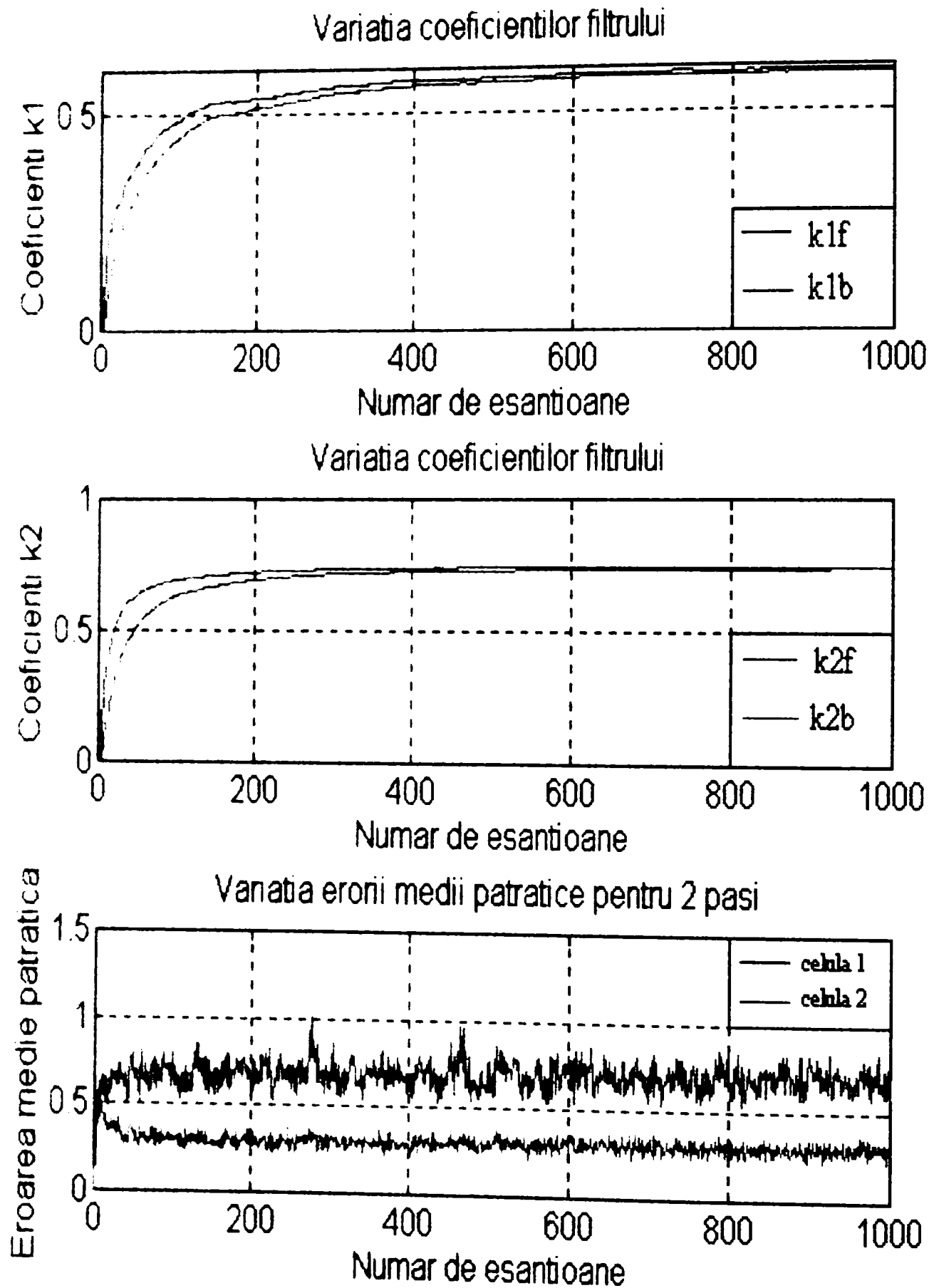


Fig. 3.9. Variația coeficienților de reflexie și a erorii medii patratice.

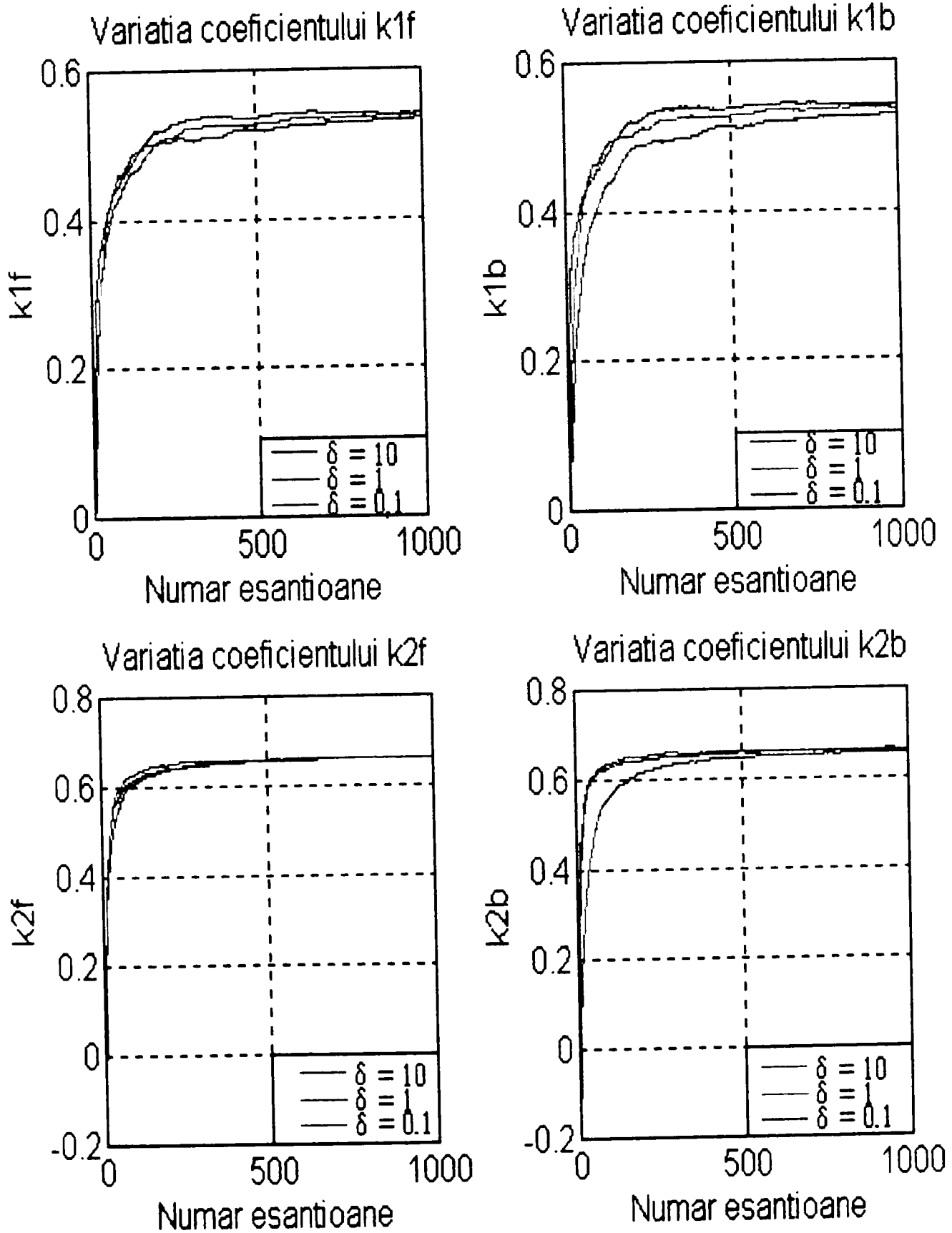


Fig. 3.10. Variația coeficienților de reflexie în funcție de valoarea inițială a erorii reziduale de predicție.

3.1.5. Structura bazată pe reactualizarea coeficientului unghiular

În [22] a fost prezentată o posibilitate de obținere a unei structuri lattice, bazată pe reactualizarea coeficientului unghiular. Obținerea acestei structuri s-a făcut pornind de la relațiile de recurență pe baza cărora s-a implementat structura prezentată în figura 3.1, relații care pot fi scrise sub forma matriceală

$$\begin{bmatrix} e_p^f(n) \\ e_p^b(n) \end{bmatrix} = \begin{bmatrix} 1 & -k_p \\ -k_p & 1 \end{bmatrix} \cdot \begin{bmatrix} e_{p-1}^f(n) \\ e_{p-1}^b(n-1) \end{bmatrix} \quad (3.10)$$

S-a definit în [22] coeficientul de reflexie normalizat

$$k_{pn} = \frac{1}{\sqrt{1+k_p^2}} < 1,$$

pe baza căruia relația (3.10) devine

$$\begin{bmatrix} e_p^f(n) \\ e_p^b(n) \end{bmatrix} = \frac{1}{\sqrt{1-k_{pn}^2}} \begin{bmatrix} 1 & -k_{pn} \\ -k_{pn} & 1 \end{bmatrix} \cdot \begin{bmatrix} e_{p-1}^f(n) \\ e_{p-1}^b(n-1) \end{bmatrix}, \quad (3.11)$$

Forma generală de determinare recursivă a erorilor de predicție printr-o rotație hiperbolică se obține notând $k_{pn} = \text{th}\theta_p$ și este dată de relația

$$\begin{bmatrix} e_p^f(n) \\ e_p^b(n) \end{bmatrix} = \begin{bmatrix} \text{ch}\theta_p & -\text{sh}\theta_p \\ -\text{sh}\theta_p & \text{ch}\theta_p \end{bmatrix} \cdot \begin{bmatrix} e_{p-1}^f(n) \\ e_{p-1}^b(n-1) \end{bmatrix} \quad (3.12)$$

Operația de adaptare atât a coeficientului unghiular cât și a coeficientului de reflexie normalizat se face pornind de la minimizarea puterii instantanee a erorii, definită ca

$$e_p(n) = [e_p^f(n)]^2 + [e_p^b(n)]^2.$$

S-a demonstrat în [22] că soluția optimă care minimizează puterea instantanee a erorii este dată de relația

$$\theta_p = \frac{1}{2} \arg \text{th} \frac{2 \cdot E \{ e_{p-1}^f(n) \cdot e_{p-1}^b(n-1) \}}{E \{ [e_{p-1}^f(n)]^2 \} + E \{ [e_{p-1}^b(n-1)]^2 \}},$$

relație care permite obținerea gradientului puterii instantanee a erorii $e_p(n)$ raportat la variația unghiulară

$$\nabla e_p(n) = -\alpha \cdot e_p^f(n) \cdot e_p^b(n), \quad (3.13)$$

unde α reprezintă o constantă de proporționalitate.

Din relația (3.7) rezultă relațiile de recurență care pot fi utilizate independent la implementarea filtrului

$$\begin{aligned} \theta_p(n+1) &= \theta_p(n) + \mu \cdot e_p^f(n) \cdot e_p^b(n) \\ k_{pn}(n+1) &= k_{pn}(n) + \mu \cdot e_p^f(n) \cdot e_p^b(n) \cdot \frac{1}{1-k_{pn}^2(n)}, \end{aligned} \quad (3.14)$$

unde μ reprezintă parametrul de adaptare.

În figura 3.11 este prezentată structura unei celule a filtrului de predicție bazată pe utilizarea coeficientului de reflexie normalizat.

Programul realizat de autor și prezentat în Anexa III.6. a permis reprezentarea formei de variație a coeficienților de reflexie normalizați precum și a erorii medii pătratice reprezentate în figura 3.12 pentru o valoare a parametrului de adaptare $\mu = 0,01$.

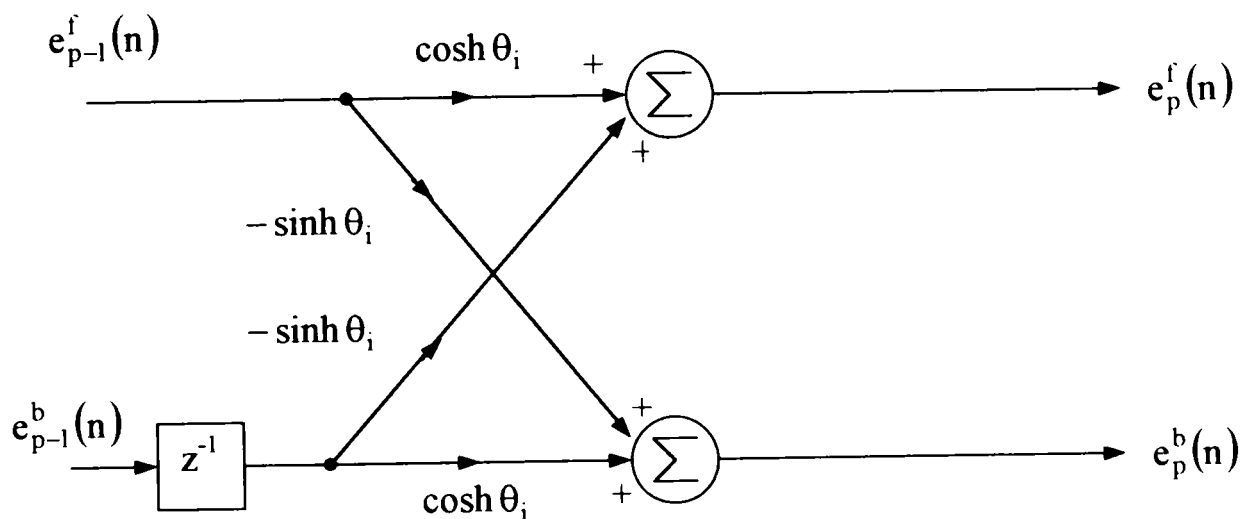


Fig. 3.11. Structura celulei lattice bazată pe reactualizarea coeficientului unghiular.

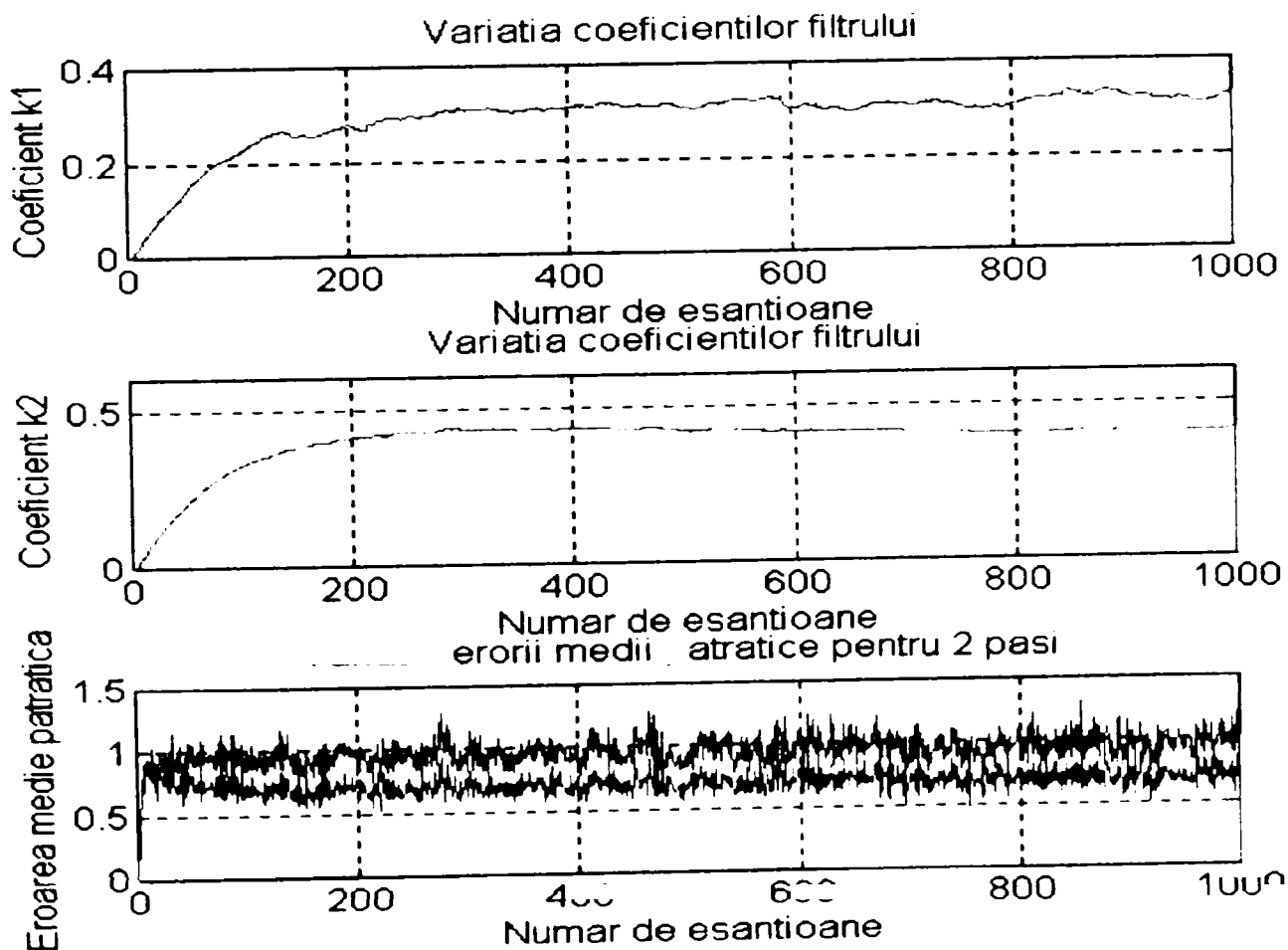


Fig. 3.12. Forma de variație a coeficienților de reflexie și a erorii medii pătratice.

Se observă că, față de alte structuri prezentate anterior, eroarea medie pătratică este mai mare.

Utilizarea acestor funcții influențează și timpul de lucru, acest algoritm având un timp de lucru de 3305 secunde, în aceleași condiții de mediere statistică a 200 de evenimente, fiecare eveniment conținând 1000 de eșantioane.

Convergența valorilor coeficienților filtrului spre valorile optimale este influențată de valoarea parametrului de adaptare, fapt ilustrat în figura 3.13. Se observă că odată cu creșterea valorii parametrului de adaptare coeficienții filtrului prezintă oscilații mai mari. De exemplu, pentru valoarea $\mu = 0,01$ oscilațiile sunt mai mari decât pentru valoarea $\mu = 0,005$, iar pentru valoarea $\mu = 0,05$ coeficienții nu mai converg spre valorile optimale.

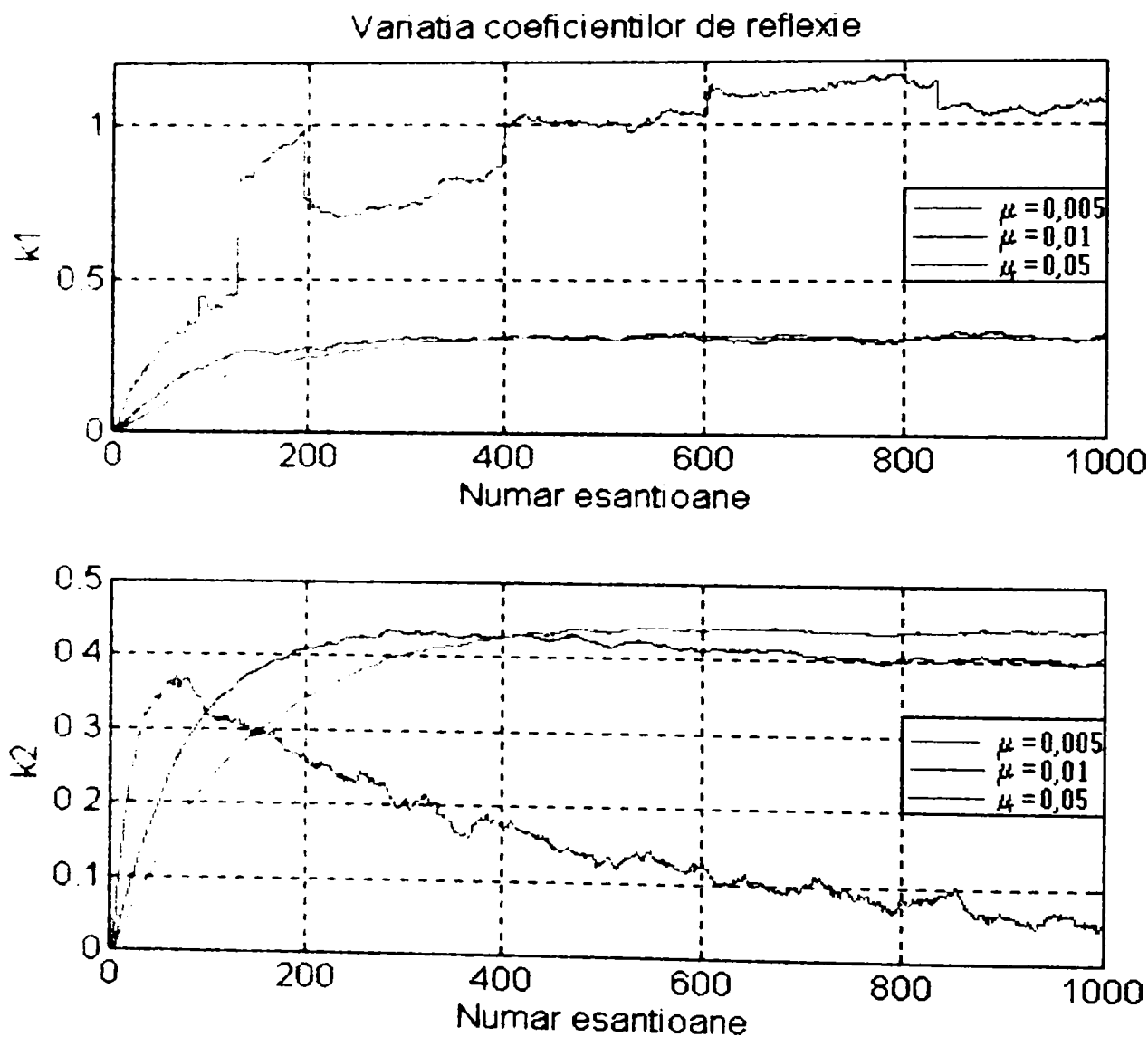


Fig. 3.13. Variația coeficienților de reflexie în funcție de parametrul de adaptare.

Deși din cele prezentate rezultă o serie de dezavantaje în utilizarea unei astfel de structuri, totuși, unul dintre avantajele reactualizării parametrului θ_p , față de reactualizarea lui k_p , constă în posibilitatea utilizării unui procesor CORDIC care operează în modul de rotație hiperbolic.

Unghiul de rotație θ_p se descompune într-o sumă de B unghiuri elementare potrivit relației

$$\theta_p = \sum_{j=1}^B \beta_{pj} \cdot a_j ,$$

unde $\beta_{pj} = \pm 1$, iar $a_j = \text{argth}2^{-j}$ reprezintă unghiurile de rotație elementară . Relația de recurență pentru erorile de predicție, dedusă în [18], poate fi rescrisă sub forma

$$\begin{bmatrix} e_p^f(n) \\ e_p^b(n) \end{bmatrix} = G_p \cdot \prod_{j=1}^B \begin{bmatrix} 1 & \beta_{pj} \cdot 2^{-j} \\ \beta_{pj} \cdot 2^{-j} & 1 \end{bmatrix} \cdot \begin{bmatrix} e_{p-1}^f(n) \\ e_{p-1}^b(n-1) \end{bmatrix} , \quad (3.15)$$

unde

$$G_p = \prod_{j=1}^B (1 - 2^{-2j})^{-1/2}$$

Pentru determinarea erorilor de predicție corespunzătoare unei celule sunt necesare un număr de $2B$ adunări și deplasări urmate de două înmulțiri . Utilizarea acestei implementări reprezintă introducerea unei restricții în ceea ce privește numărul de valori distincte pe care le poate lua θ_p , mai precis pentru o valoare a parametrului B rezultă că θ_p poate lua 2^B valori distincte.

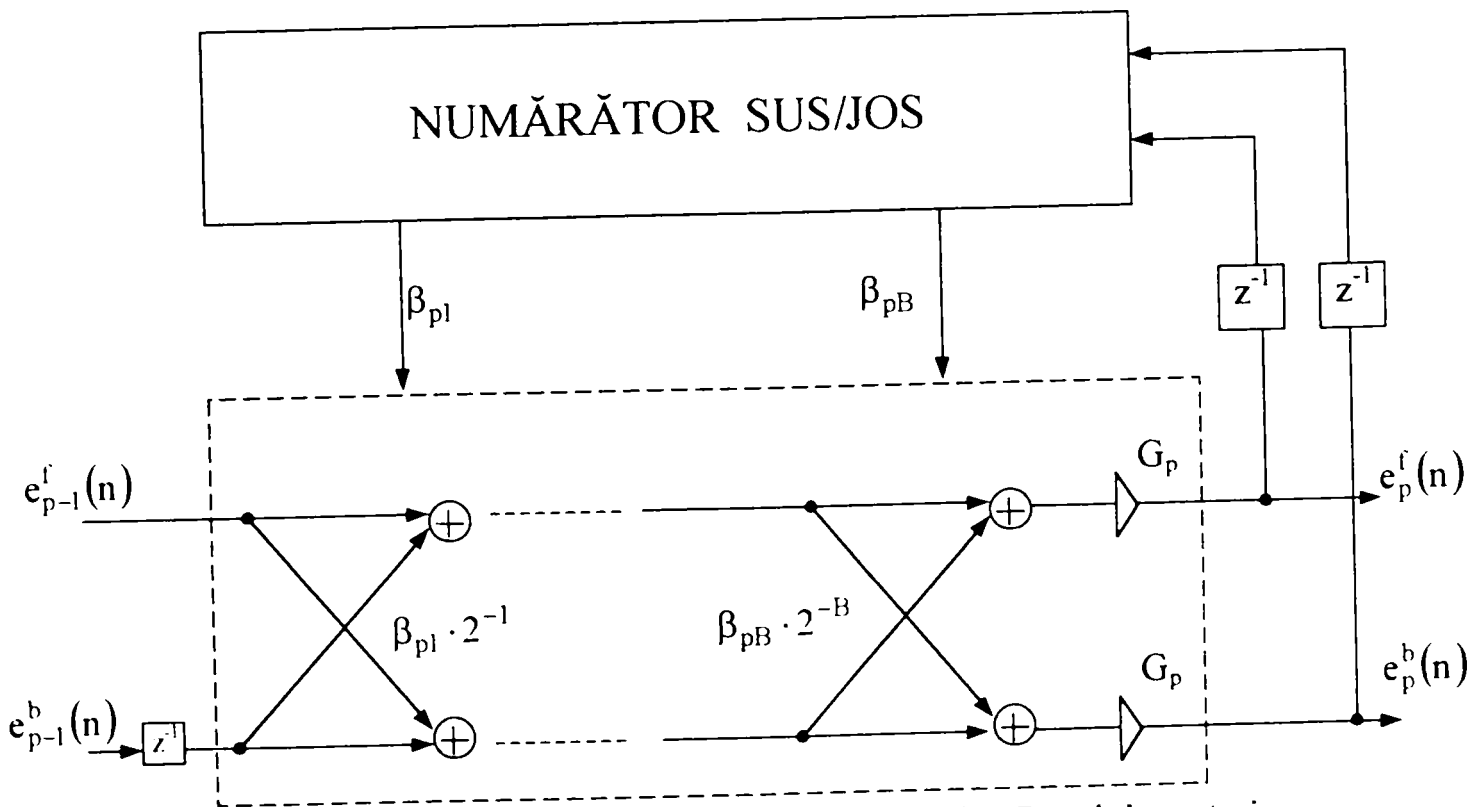


Fig. 3.14. Structura unui pas a filtrului lattice conținând B pași elementari.

Pentru creșterea rezoluției în alegerea lui θ_p trebuie crescută valoarea parametrului B, fapt care duce la creșterea volumului de calcule . În [22] s-a demonstrat că este mai ușoară reactualizarea coeficienților β_{pj} decât a lui θ_p . Aceasta se poate realiza utilizând o structură hardware foarte simplă. Modulul care generează noul set de coeficienți $\beta_{pj}, 1 \leq j \leq B$ poate fi

implementat cu ajutorul unui numărator binar bidirecțional cu o logică de comandă simplă. Cele B ieșiri ale număratorului vor constitui vectorul β_{pj} care controlează procesorul CORDIC care are structura prezentată în figura 3.14.

Pentru îmbunătățirea vitezei de lucru s-au dezvoltat două strategii, prima strategie utilizabilă constând în utilizarea tehnicii pipeline între etapele succesive de calcul ale erorilor de predicție, iar a doua variantă constând în introducerea tehnicii pipeline chiar între pașii elementari.

3.2. Tipuri de structuri de filtre lattice cu reacție

3.2.1. Structura generală a celulei lattice cu reacție

Forma prin care se descrie un filtru IIR, prezentată în [3], este dată de relația

$$\sum_{k=0}^N a_k(n)y(n-k) = \sum_{k=0}^N b_k(n) \cdot x(n-k), \quad a_0=1 \quad (3.16)$$

unde $y(n)$ reprezintă ieșirea obținută prin filtrarea semnalului de intrare $x(n)$.

Se cunoaște din [3] și [20] că forma directă de implementare a filtrelor IIR prezintă unele dezavantaje cum ar fi necesitatea rezolvării unor probleme relativ dificile în ceea ce privește stabilitatea, viteza de convergență redusă, precum și proprietăți slabe în ceea ce privește precizia cu care se calculează valoarea semnalului de la ieșire datorată reprezentării datelor în calculator. Pe de altă parte structura lattice are proprietăți superioare în ceea ce privește precizia față de forma directă. De asemenea, în timp ce problema realizării stabilității formei directe a filtrelor IIR este în general dificil de rezolvat necesitând un volum de calcule mare, în cazul filtrelor lattice problema stabilității este practic rezolvată fără a mai necesita calcule suplimentare.

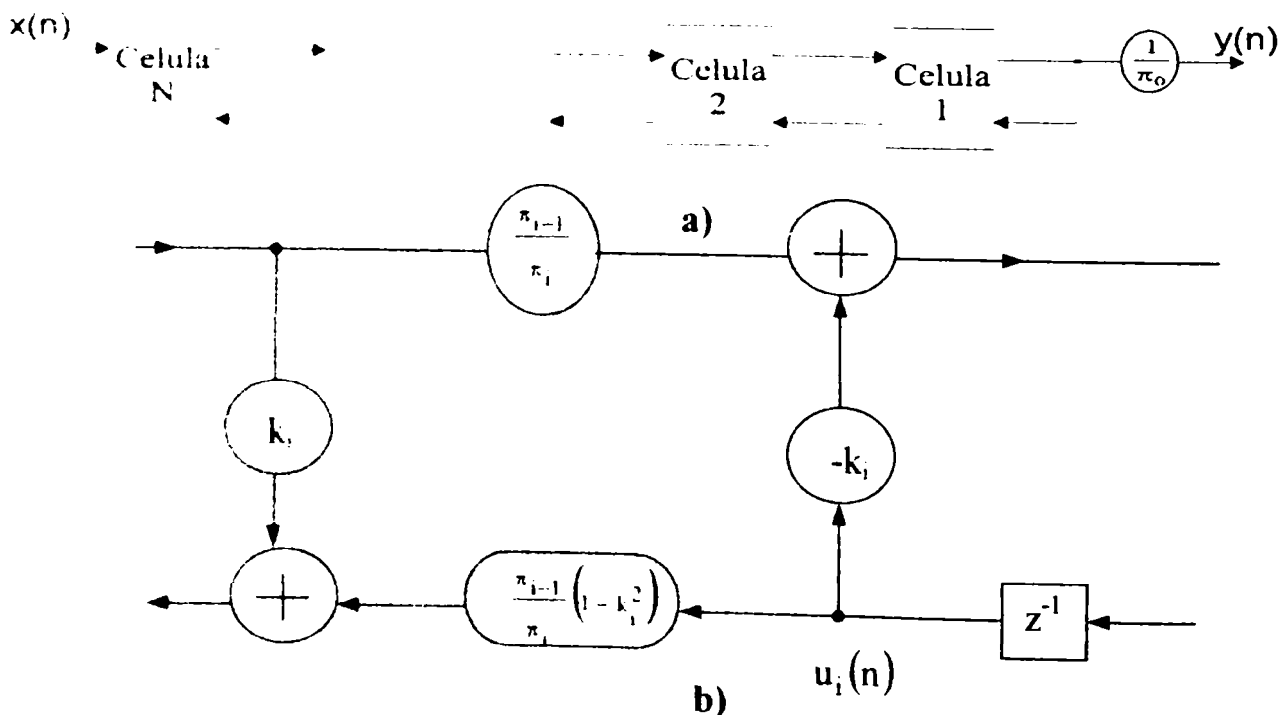


Fig. 3.15. Structura lattice feedback: a) Structura generală; b) Structura unei celule.

Deoarece funcția de transfer $H(z)$ a filtrului IIR este o funcție rațională, rezultă că pentru implementarea structurii care caracterizează numitorul este necesară realizarea unei reacții de la ieșire la intrare (feedback). În cazul utilizării structurilor lattice devine necesară implementarea formei lattice feedback cu structura globală prezentată în fig. 3.15.

S-a arătat în [20] că dacă coeficienții k_i , $i = 1, K, N$ reprezintă parametrii filtrului, π_i reprezentând parametri π ai filtrului exprimați în funcție de coeficienții k_i ($\pi_N = 1$), structura generală prezentată în fig. 3.15 a) este descrisă de ecuațiile (3.17)

$$\begin{cases} \mathbf{u}(n+1) = \mathbf{A} \cdot \mathbf{u}(n) + \mathbf{B} \cdot x(n) \\ y(n) = \mathbf{C} \cdot \mathbf{u}(n) + x(n) \end{cases}, \quad (3.17)$$

unde

$$\mathbf{A} = \begin{pmatrix} -k_1 & -\frac{\pi_0}{\pi_1} k_2 & -\frac{\pi_0}{\pi_2} k_3 & \dots & -\frac{\pi_0}{\pi_{N-1}} k_N \\ \frac{\pi_1}{\pi_0} (1 - k_1^2) & -k_1 k_2 & -\frac{\pi_1}{\pi_2} k_1 k_3 & \dots & \frac{\pi_1}{\pi_{N-1}} k_1 k_N \\ 0 & \frac{\pi_2}{\pi_1} (1 - k_2^2) & -k_2 k_3 & \dots & \frac{\pi_2}{\pi_{N-1}} k_2 k_N \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & -k_{N-1} k_N \end{pmatrix}, \quad (3.18)$$

$$\mathbf{B}^T = \begin{pmatrix} \frac{\pi_0}{\pi_N} & \frac{\pi_0}{\pi_N} k_1 & \dots & \frac{\pi_0}{\pi_N} k_{N-2} & \frac{\pi_{N-1}}{\pi_N} k_{N-1} \end{pmatrix},$$

$$\mathbf{C} = \begin{pmatrix} -\frac{k_1}{\pi_0} & \dots & -\frac{k_N}{\pi_{N-1}} \end{pmatrix}$$

$$\mathbf{u}^T(n) = (u_1(n) \ u_2(n) \ \dots \ u_N(n)),$$

ecuații care conduc la forma generală a unei singure celule prezentată în figura 3.15. b).

Din figura 3.15 se observă că structura generală a unei celule este relativ greu de implementat datorită volumului mare de calcule necesar. Prin particularizarea valorilor coeficienților π_i și k_i se obțin forme mai ușor de implementat ale structurii celulelor.

3.2.2. Structura lattice feedback cu două multiplicări

Această structură rezultă considerând valorile coeficienților π_i egale cu 1, structura fiind descrisă de relațiile (3.17), în care matricele sunt

$$\mathbf{A} = \begin{bmatrix} -k_1 & -k_2 & -k_3 & \dots & -k_N \\ (1 - k_1^2) & -k_1 k_2 & -k_1 k_3 & \dots & -k_1 k_N \\ 0 & (1 - k_2^2) & -k_2 k_3 & \dots & -k_2 k_N \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & -k_{N-1} k_N \end{bmatrix},$$

$$\mathbf{B}^T = [1 \quad k_1 \quad k_{N-2} \quad k_{N-1}], \quad (3.19)$$

$$\mathbf{C} = [-k_1 \quad k_2 \quad \dots \quad -k_N].$$

Se obține filtrul lattice a cărui structură este prezentată în figura 3.16.

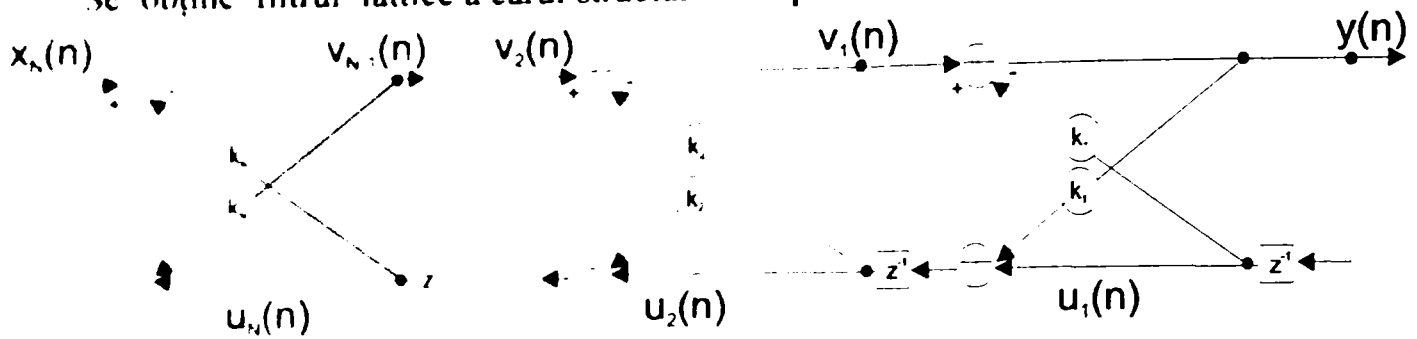


Fig. 3.16. Structura lattice feedback cu două multiplicări.

3.2.3. Structura lattice feedback normalizată

Obținerea structurilor normalizate se bazează pe alegerea corespunzătoare a coeficienților matricelor din ecuațiile (3.17) și (3.18). Se cunosc două forme normalizate de structură.

a) Dacă se aleg coeficienții conform relațiilor

$$k_i = \sin \theta_i$$

$$\frac{\pi_{i+1}}{\pi_i} = \sqrt{1 - k_i^2} = \cos \theta_i, \quad \forall i = 1, 2, \dots, N \quad (3.20)$$

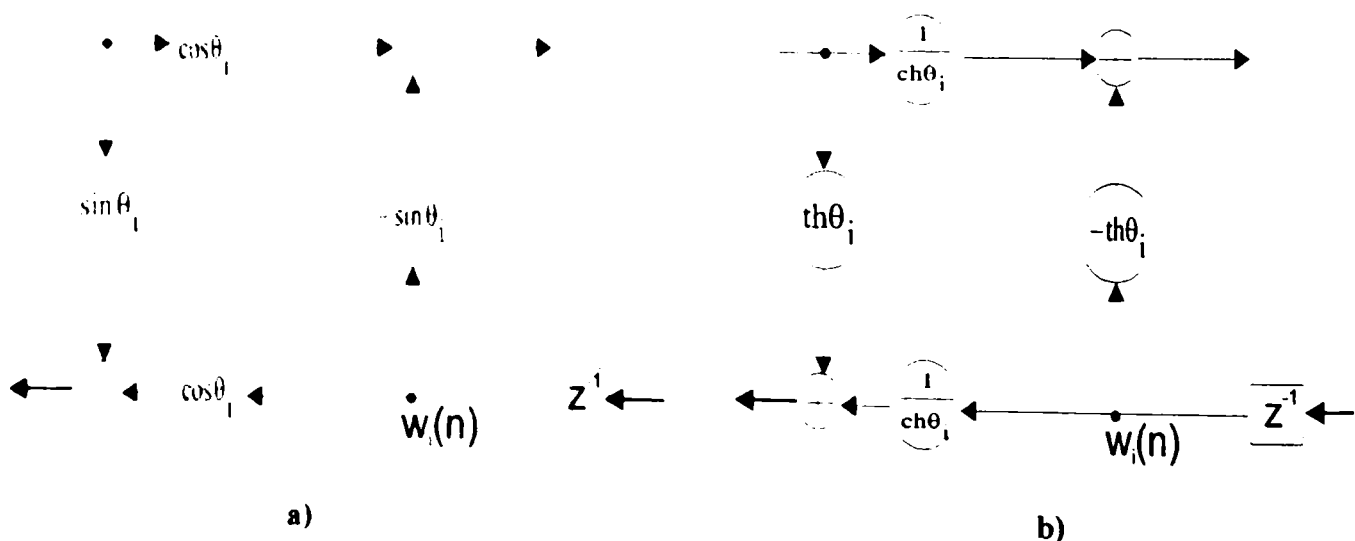
se obține structura lattice feedback normalizată în forma 1 prezentată în figura 3.17 a) care permite utilizarea tehnicilor CORDIC pentru implementare.

b) Dacă se aleg coeficienții conform relațiilor

$$k_i = \text{th} \theta_i$$

$$\frac{\pi_{i+1}}{\pi_i} = \frac{1}{\text{ch} \theta_i}, \quad \forall i = 1, \dots, N \quad (3.21)$$

se obține structura lattice feedback normalizată în forma 2, prezentată în figura 3.17 b)



a) b)
Fig. 3.17. Structura celei lattice feedback normalizate :
a) forma normalizată 1; b) forma normalizată 2.

3.3. Filtre adaptive IIR

3.3.1. Arhitectura filtrelor adaptive IIR utilizând structuri lattice

Pentru formele normalizate de implementare matricea \mathbf{A} și vectorii \mathbf{B} și \mathbf{C} verifică ecuațiile date de relațiile

$$\begin{aligned} \mathbf{A}' &= \boldsymbol{\pi} \cdot \mathbf{A} \cdot \boldsymbol{\pi}^{-1} \\ \mathbf{B}' &= \boldsymbol{\pi} \cdot \mathbf{B} \\ \mathbf{C}' &= \mathbf{C} \cdot \boldsymbol{\pi}^{-1} \end{aligned} \quad (3.22)$$

unde matricea $\boldsymbol{\pi}$ este o matrice diagonală conținând elementele $\pi_0, \pi_1, \dots, \pi_{N-1}$.

Implementarea filtrelor IIR utilizând structura lattice cascade presupune determinarea unei relații între coeficienții $a_i, i=1, \dots, N$ și coeficienții $k_i, i=1, \dots, N$. În [20] și [23] s-a prezentat relația de legătură

$$\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix} = \mathbf{T}^T \cdot \begin{bmatrix} k_1 \\ k_2 \\ \vdots \\ k_N \end{bmatrix}, \quad (3.23)$$

unde matricea \mathbf{T} se poate determina conform algoritmului

$$\left\{ \begin{array}{l} \mathbf{T}_1 = 1 \\ \mathbf{T}_2 = \begin{bmatrix} 1 & 0 \\ k_1 & 1 \end{bmatrix} \\ \dots \\ \mathbf{T}_m = \begin{bmatrix} & & & 0 \\ & & & \vdots \\ & & & 0 \\ k_{m-1} & (k_{m-1}k_1 & k_{m-1}k_2 & \dots & k_{m-1}k_{m-2} & 1) \cdot \mathbf{T}_{m-1} \end{bmatrix} \end{array} \right. \quad (3.24)$$

Din relațiile (3.24) se observă că matricea \mathbf{T} este exprimată într-o formă recursivă.

Dacă filtrul lattice este utilizat într-o structură adaptivă, coeficienții vor fi variabili în timp, $\hat{k}_i(n)$, deci și matricea \mathbf{T} va deveni o matrice variabilă în timp, $\hat{\mathbf{T}}_N(n)$. Structura unui filtru IIR cu numitorul realizat cu o structură lattice cu două multiplicări este prezentată în figura 3.18. Deoarece relația (3.23) rămâne valabilă și dacă parametrii filtrului corespund structurii normalizate, o altă posibilitate de a implementa un filtru IIR este cea prezentată în figura 3.19., în care numitorul este implementat cu o structură lattice normalizată. De asemenea, dacă vectorul de stare care caracterizează structura normalizată este

$$\mathbf{w}(n) = [w_1(n) \ w_2(n) \ \dots \ w_N(n)]^T$$

iar cel care caracterizează structura cu două multiplicări este

$$\mathbf{u}(n) = [u_1(n) \ u_2(n) \ \dots \ u_N(n)]^T,$$

s-a arătat în [20] că este satisfăcută relația

$$\mathbf{w}(n) = \pi \mathbf{u}(n) = \pi \cdot \mathbf{T} \cdot \mathbf{y}(n) \quad (3.25)$$

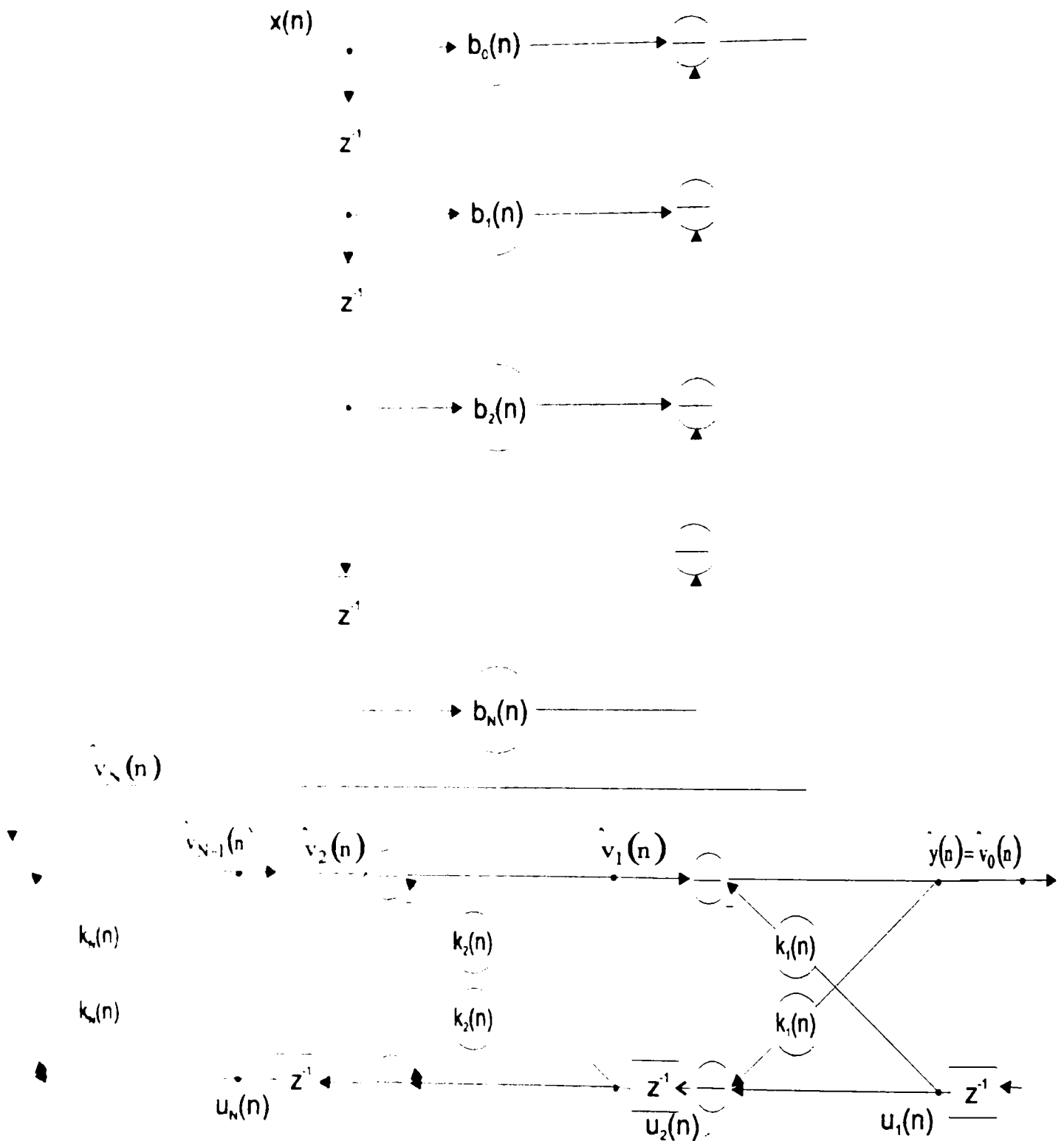


Fig. 3.18. Filtru IIR realizat prin implementarea numitorului cu structură lattice cu două multipicări.

În [23] s-a prezentat o altă formă posibilă de implementare a filtrului IIR, numită filtru IIR normalizat și prezentată în figura 3.20.

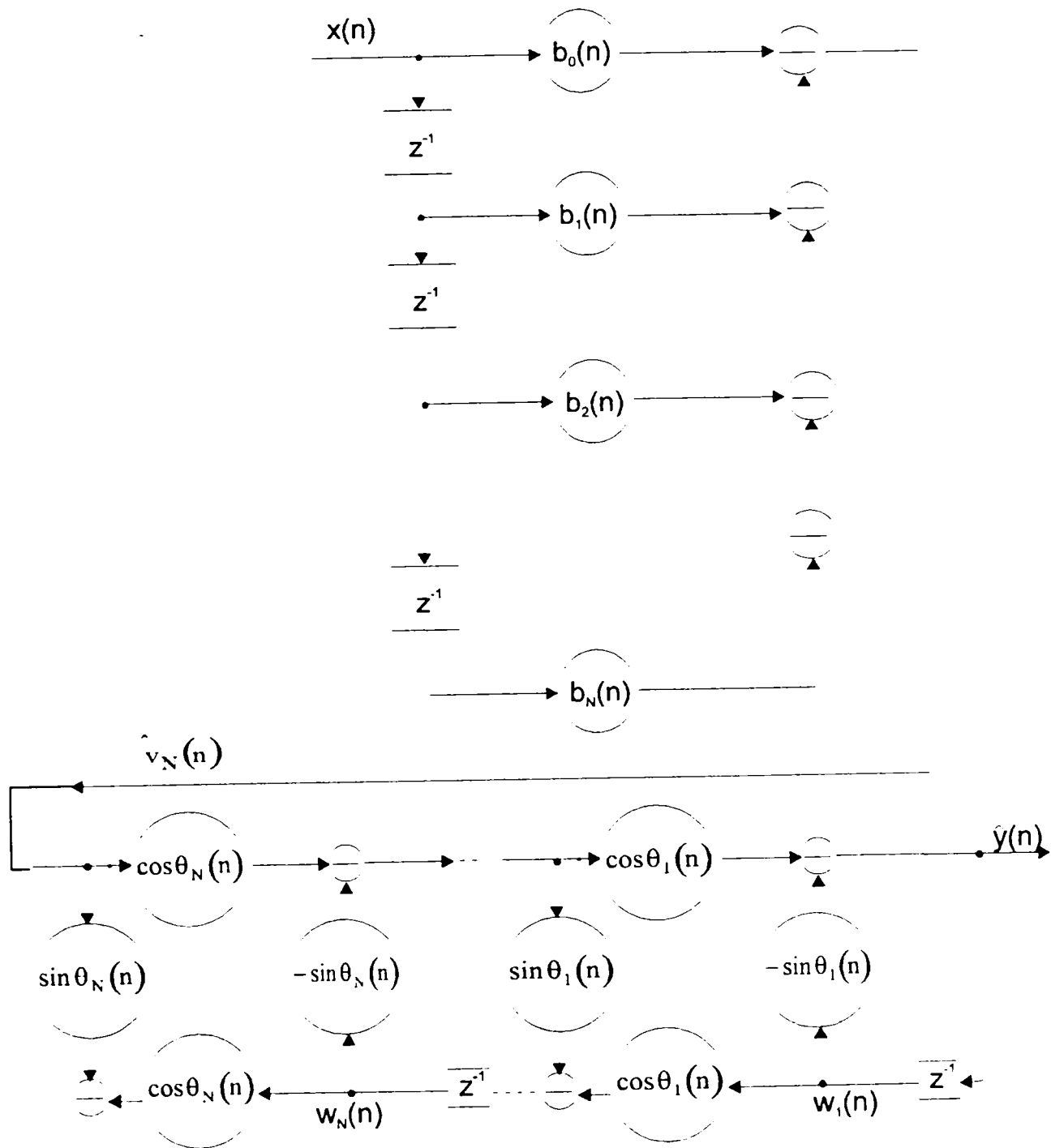


Fig. 3.19. Filtru IIR realizat prin implementarea numitorului cu structura lattice normalizată.

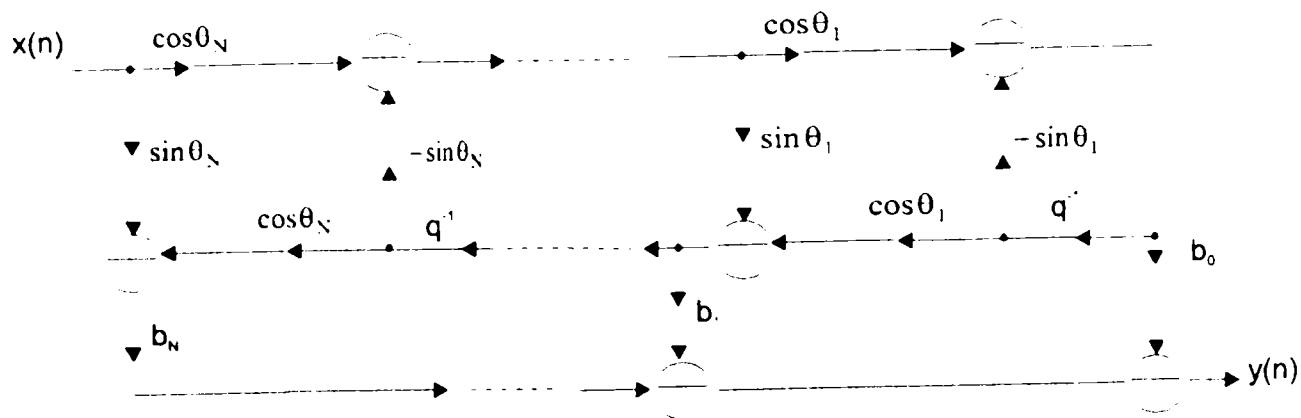


Fig. 3.20. Filtru lattice cu structura normalizată.

3.3.2. Algoritmi de implementare a filtrelor adaptive IIR

Pentru realizarea unui filtru adaptiv este necesară reactualizarea coeficienților filtrului la fiecare moment de timp. Implementarea unui filtru adaptiv utilizând un filtru IIR presupune, în varianta implementării filtrului IIR cu numitorul având o structură lattice cascade cu celule realizate cu două multiplicări, determinarea relațiilor de recurență pentru coeficienții b_i și k_i ai filtrului

3.3.2.1. Algoritm bazat pe utilizarea gradientului

Acest algoritm determină setul de parametri $[\hat{b}_0 \ \hat{b}_1 \ \dots \ \hat{b}_N \ \hat{k}_1 \ \dots \ \hat{k}_N]$ ai filtrului utilizând o relație similară cu cea de la metoda gradientului

$$\begin{bmatrix} \hat{\mathbf{k}}(n+1) \\ \hat{\mathbf{b}}(n+1) \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{k}}(n) \\ \hat{\mathbf{b}}(n) \end{bmatrix} - \frac{1}{2} \mu \cdot \frac{d}{d\hat{\theta}_1} \cdot E\{e^2(n)\}, \quad (3.26)$$

unde $\hat{\theta}_1 = [\hat{\mathbf{k}}^T \ \hat{\mathbf{b}}^T]$ reprezintă vectorul parametrilor filtrului IIR cu structura lattice, μ reprezintă o constantă pozitivă iar $e(n)$ reprezintă eroarea de la ieșire de forma

$$e(n) = y(n) - \hat{y}(n).$$

S-a notat cu

$$\bar{A}(q^{-1}, n) = \begin{bmatrix} 1 & \bar{k}_1(n) & \bar{k}_2(n) & \dots & \bar{k}_N(n) \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & \bar{T}_N(N-1) \end{bmatrix} \cdot \begin{bmatrix} 1 \\ q^{-1} \\ \vdots \\ q^{-N} \end{bmatrix}, \quad (3.27)$$

unde q^{-1} reprezintă operatorul de întârziere în domeniul timp.

S-a demonstrat în [20] că relația (3.27) este echivalentă într-o versiune simplificată cu relația de reactualizare a parametrilor filtrului IIR.

$$\begin{bmatrix} \bar{\mathbf{k}}(n+1) \\ \bar{\mathbf{b}}(n+1) \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{k}}(n) \\ \bar{\mathbf{b}}(n) \end{bmatrix} + \mu \cdot \begin{bmatrix} -\bar{\mathbf{u}}_A(n) \\ \bar{\mathbf{x}}_A(n) \end{bmatrix} \cdot e(n), \quad (3.28)$$

unde

$$\bar{\mathbf{x}}_A(n) = \begin{bmatrix} \bar{x}(n) \\ \bar{A}(q^{-1}, n) \\ \vdots \\ \bar{x}(n-N) \\ \bar{A}(q^{-1}, n) \end{bmatrix} \quad \text{și} \quad \bar{\mathbf{u}}_A(n) = \begin{bmatrix} \bar{u}(n) \\ \bar{A}(q^{-1}, n) \\ \vdots \\ \bar{u}(n-N) \\ \bar{A}(q^{-1}, n) \end{bmatrix} \quad (3.29)$$

Această versiune a algoritmului este ușor de implementat deoarece nu necesită operații cu matrice, relația (3.28) fiind echivalentă cu un sistem de $2N+1$ ecuații scalare.

Structura unui filtru adaptiv bazată pe acest algoritm este prezentată în figura 3.21.

Rolul filtrelor lattice este de a determina vectorii $\bar{\mathbf{x}}_A(n)$ respectiv $\bar{\mathbf{u}}_A(n)$ necesari pentru implementarea algoritmului.

3.3.2.2. Algoritmul Steiglitz - McBride

Așa cum s-a prezentat în [20] și [23], setul de termeni ai matricilor $\hat{\mathbf{x}}_A(n)$ și $\hat{\mathbf{u}}_A(n)$ se poate determina utilizând direct ieșirea sistemului $y(n)$ și nu din ieșirea filtrului IIR lattice. Pe baza celor prezentate, structura unui filtru adaptiv bazată pe utilizarea unui filtru lattice IIR este prezentată în figura 3.22.

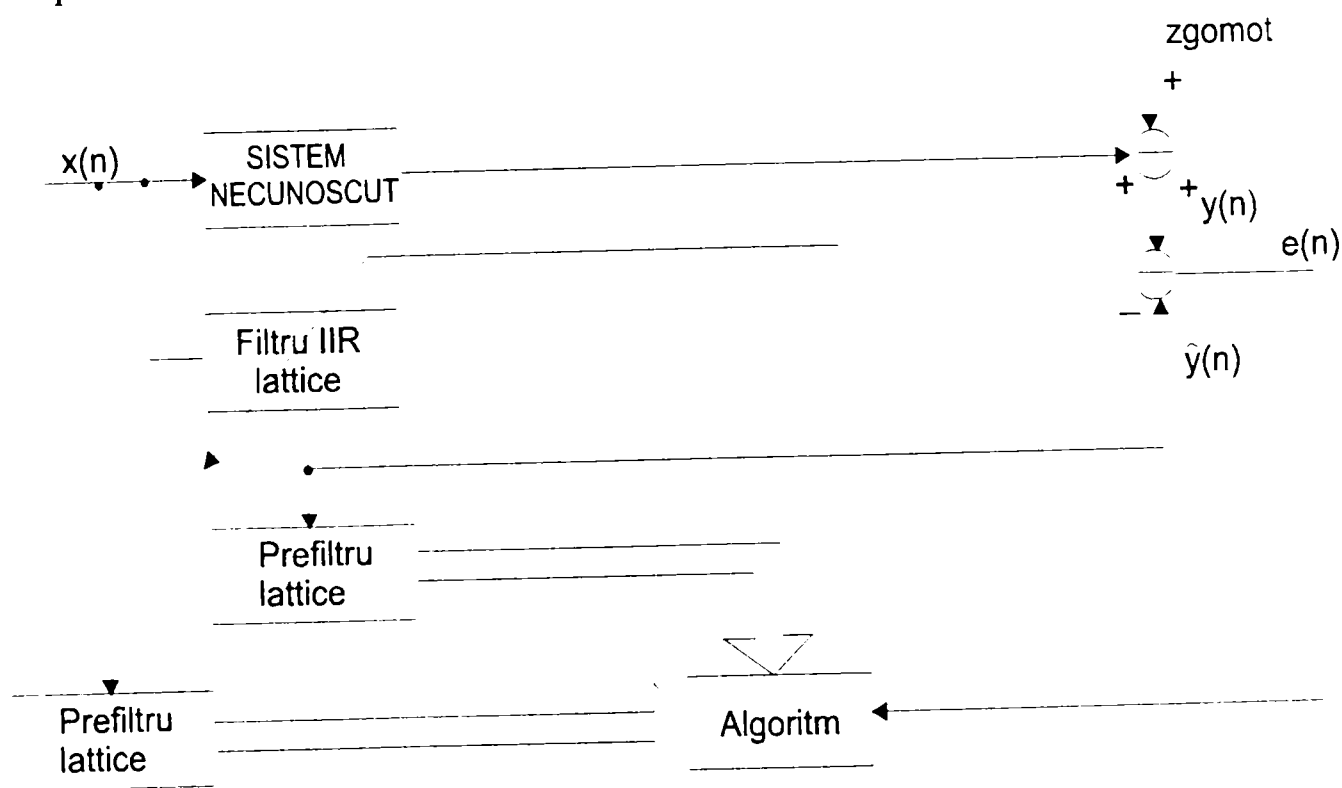


Fig. 3.21. Implementarea filtrului adaptiv bazată pe algoritmul gradientului.

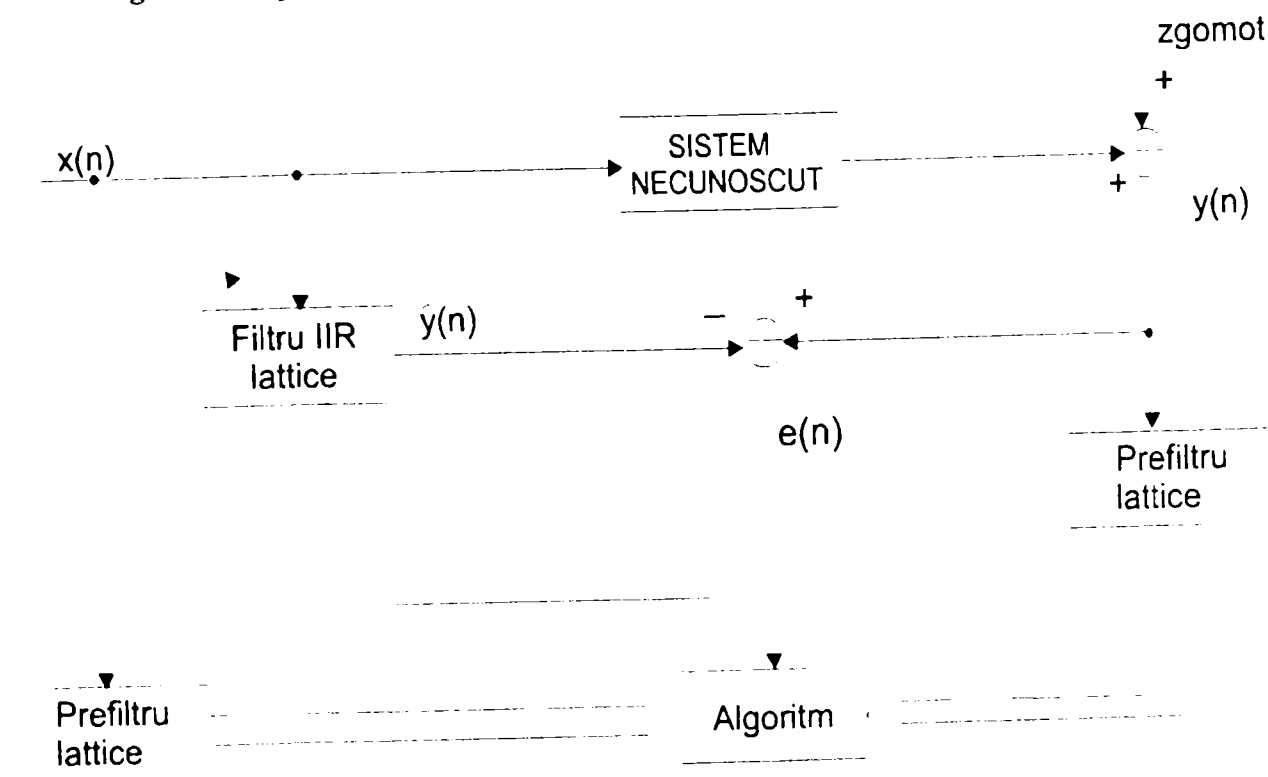


Fig. 3.22. Implementarea filtrului adaptiv utilizând algoritmul Steiglitz – McBride.

3.3.2.3. Algoritmul SHARF

În [20] s-a prezentat un algoritm de reactualizare a coeficienților filtrului IIR lattice atât în funcție de valoarea curentă a erorii dar și de valorile precedente ale ei, algoritm denumit SHARF. Reactualizarea coeficienților filtrului IIR lattice se face pe baza relației

$$\begin{bmatrix} \mathbf{k}_N(n+1) \\ \mathbf{b}_N(n+1) \end{bmatrix} = \begin{bmatrix} \mathbf{k}_N(n) \\ \mathbf{b}_N(n) \end{bmatrix} + \mu \cdot \begin{bmatrix} -\hat{\mathbf{u}}_N(n) \\ \mathbf{x}_N(n) \end{bmatrix} \cdot v(n),$$

unde

$$v(n) = e(n) + \sum_{i=1}^N c_i e(n-i)$$

Pentru asigurarea stabilității filtrului este necesară îndeplinirea condiției

$$\operatorname{Re} \left(\frac{1 + c_1 z^{-1} + \dots + c_N z^{-N}}{1 + a_1 z^{-1} + \dots + a_N z^{-N}} \right)_{|z|=1} > 0$$

Structura filtrului adaptiv lattice care utilizează algoritmul SHARF este prezentată în figura 3.23.

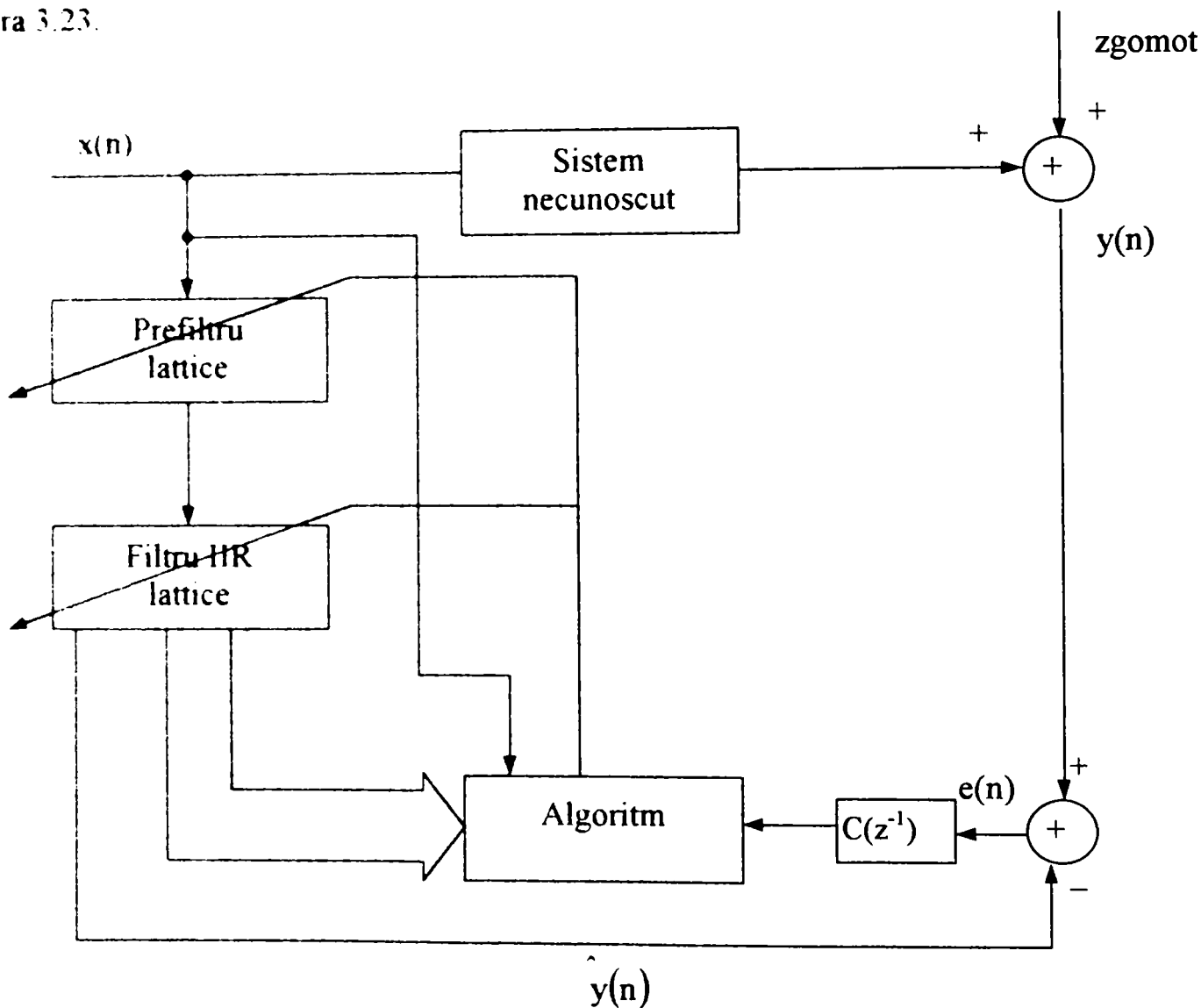


Fig. 3.23. Implementarea filtrului adaptiv utilizând algoritmul SHARF.

3.4. CONCLUZII

În acest capitol au fost prezentate 5 tipuri de structuri de filtre adaptive lattice, autorul studiind performanțele fiecărei structuri și obținând următoarele concluzii:

1. Numărul de celule al filtrului lattice este determinat de structura sistemului, deci eroarea medie pătratică minimă care se poate obține la ieșirea filtrului adaptiv este bine determinată. Creșterea numărului de celule a filtrului nu duce la reducerea în continuare ale erorii medii pătratice ci doar la creșterea timpului de calcul.

2. În cazul structurii bazate pe metoda gradientului adaptiv, autorul a demonstrat că stabilitatea filtrului depinde de valoarea parametrului de adaptare (figura 3.5) precum și de valoarea zgomotului care influențează canalul (figura 3.6.). Autorul a explicat de ce valorile optime ale coeficienților de reflexie depind de valoarea dispersiei zgomotului.

3. În cazul structurii bazate pe metoda gradientului adaptiv normalizat, autorul a arătat că o alegere corespunzătoare a constantei α conduce la o stabilitate mai bună a filtrului (figura 3.7).

4. În cazul structurii bazate pe utilizarea conceptului de spații vectoriale autorul a putut determina forma de variație a coeficienților de reflexie “înainte” și “înapoi” (figura 3.9) precum și influența erorii reziduale de predicție asupra formei de variație a acestora (figura 3.10).

5. În cazul structurii bazate pe reactualizarea coeficientului unghiular autorul a determinat forma de variație a coeficienților de reflexie (figura 3.12) precum și influența parametrului de adaptare asupra convergenței valorilor coeficienților filtrului (figura 3.13).

6. Autorul a realizat o comparație privitoare la timpul de calcul necesar, testând cele 5 structuri în aceleași condiții de mediere statistică a 200 de evenimente, fiecare conținând un număr de 1000 de eșantioane. Rezultatele sunt sintetizate în Tabelul 3.2..

Tabelul 3.2.
Timpul de calcul necesar implementării diferitelor structuri lattice.

Tipul de structură de filtru lattice	Timp(secunde)
Structura bazată pe algoritmul lui Durbin	323
Structura bazată pe metoda gradientului adaptiv	220
Structura bazată pe metoda gradientului adaptiv normalizat	283
Structura bazată pe utilizarea conceptului de spații vectoriale	498
Structura bazată pe reactualizarea coeficientului unghiular	3305

Deși timpul de calcul necesar implementării soft a structurii bazată pe reactualizarea coeficientului unghiular este cel mai mare, această structură este utilizată datorită existenței procesoarelor de semnal CORDIC care permit reducerea semnificativă a timpului de calcul.

7. În paragraful 3.2 au fost prezentate principalele tipuri de structuri lattice cu reacție, iar în paragraful 3.3 au fost prezentate posibilitățile de implementare a structurilor de filtre IIR utilizând structura lattice cu reacție.

4. FILTRE ADAPTIVE NELINIARE

Capitolul de față prezintă un studiu realizat de autor referitor la posibilitățile de utilizare a filtrelor adaptive neliniare atât în scopul filtrării perturbațiilor cu caracter negaussian cât și în scopul prelucrării semnalelor cu nivelul mediei nestaționar (de exemplu prelucrarea imaginilor). **Deși filtrarea, netezirea sau predicția mărimii de interes este un operator liniar de datele prezente la intrarea filtrului, deoarece, așa cum se va prezenta în continuare, relațiile de reactualizare a coeficienților filtrului adaptiv sunt neliniare, filtrele adaptive prezentate sunt considerate în literatură ca filtre adaptive neliniare.** Pe baza concluziilor desprinse, autorul a propus un nou tip de filtru adaptiv, care prezintă atât avantajele filtrelor adaptive neliniare cât și a filtrelor adaptive IIR. De asemenea, autorul a realizat o prezentare a principalelor tipuri de filtre neliniare în ceea ce privește structura, mod de implementare și performanțe, arătând la fiecare tip de filtru neliniar cum poate fi folosit ca filtru neliniar adaptiv.

4.1. Algoritmi OSLMS

4.1.1. Influența perturbațiilor cu caracter negaussian asupra filtrării LMS

S-a arătat că filtrele adaptive liniare bazate pe metoda gradientului și obținute prin implementarea algoritmului LMS utilizează o estimare instantanee asupra gradientului la suprafața erorii medii pătratice pentru a reactualiza coeficienții filtrului. O astfel de strategie implică faptul că algoritmul LMS este extrem de vulnerabil la influența impulsurilor, în special în cazul impulsurilor de amplitudine mare și durată scurtă.

Acest fapt rezultă și din figura 4.1. în care s-a reprezentat forma de variație a coeficienților filtrului Wiener utilizat pentru eliminarea perturbațiilor introduse de canalul de comunicații prezentat în capitolul 2, precum și a erorii medii pătratice, dacă peste semnalul de la intrarea filtrului adaptiv se suprapun impulsuri de amplitudini diferite, respectiv 1,2,3 unități, valoarea parametrului de adaptare fiind $\alpha = 0,05$. Se observă că odată cu creșterea amplitudinii impulsurilor variația momentană a coeficienților filtrului este tot mai mare, simultan cu creșterea valorii erorii medii pătratice. De asemenea, se observă că între semnalul de la ieșirea filtrului și semnalul dorit a se obține există diferență în perioada de atingere a valorilor optime a coeficienților optimi (aproximativ primele 150 de eșantioane) precum și după fiecare apariție a unui impuls, amplitudinea impulsului influențând și perioada în care cele două semnale nu coincid. Pierderea stabilității algoritmului poate fi controlată în practică prin reducerea parametrului de adaptare α , dar cu repercursiuni asupra capacității algoritmului de a răspunde la semnale nestaționare de intrare.

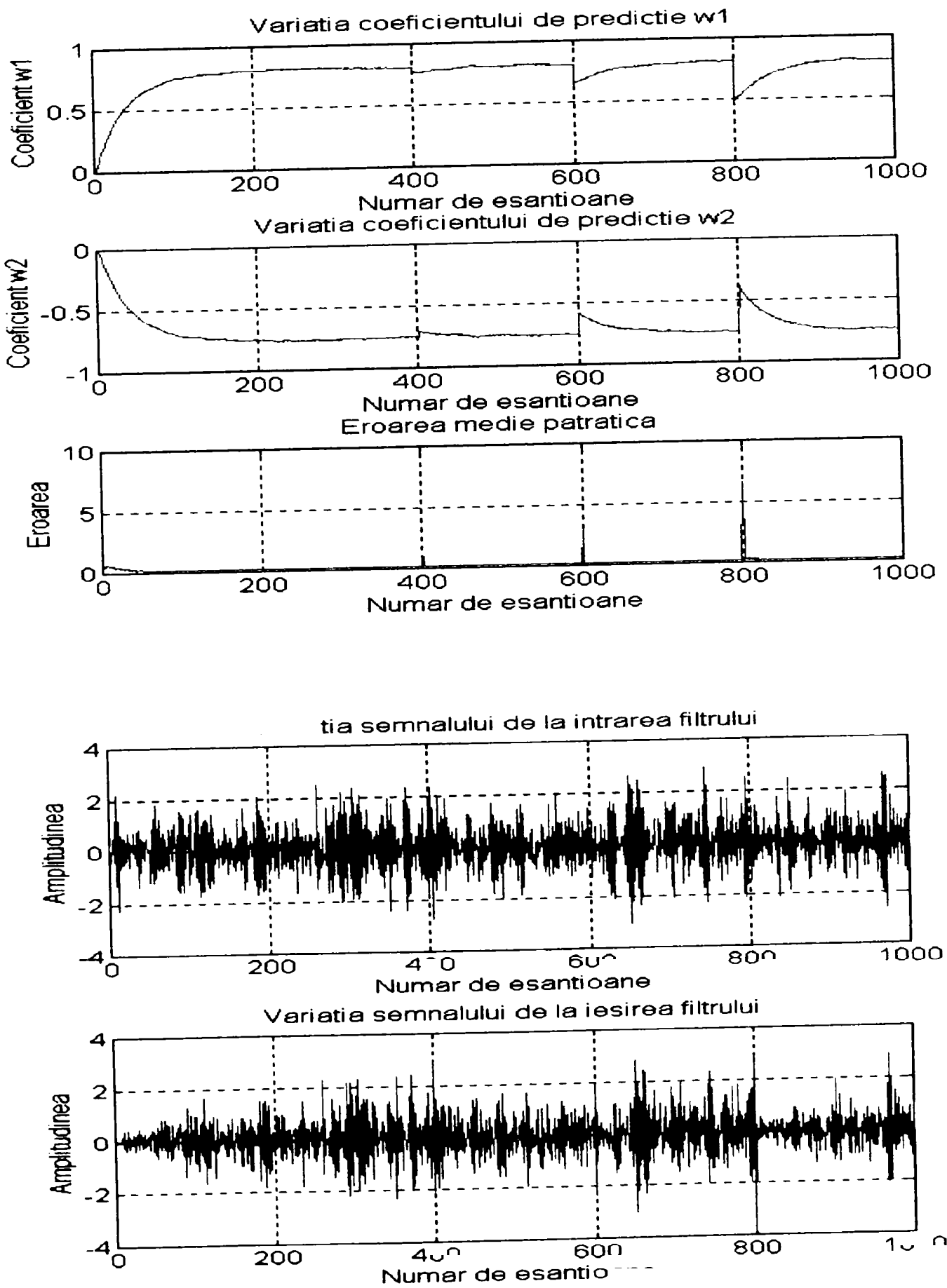


Fig. 4.1. Variația coeficienților de predicție, a erorii medii pătratice și a semnalului de la intrarea și ieșirea filtrului adaptiv.

Algoritmii OSLMS (Order Statistic Least Mean Square) permit obținerea unor performanțe ale filtrelor adaptive mai apropiate de cerințele optime și într-o gamă mult mai largă în ceea ce privește domeniul evenimentelor de la intrarea filtrului începând cu zgomotul gaussian, până la impulsurile de amplitudine mare și durată scurtă.

4.1.2. Principiul filtrării OSLMS

O variantă a algoritmului LMS, (filtrarea OSLMS), se obține reactualizând coeficienții filtrului pe baza relației (4.1), prezentată în [41]

$$\mathbf{w}_N(n+1) = \mathbf{w}_N(n) + \alpha \cdot \mathbf{M}_N(n) \mathbf{a}, \quad (4.1)$$

unde α este parametrul de adaptare, iar $\mathbf{M}_N(n)$ este o matrice de dimensiunile $(N \times L)$ dată de relația

$$\mathbf{M}_N(n) = \Gamma \{ \mathbf{e}(n) \mathbf{x}_N(n) \}. \quad (4.2)$$

Prin $\Gamma \{ \cdot \}$ se înțelege o transformare algebrică de ordonare în sens crescător aplicată fiecărei linii a secvenței

$$\{ \mathbf{e}(n) \mathbf{x}_N(n) \quad \mathbf{e}(n-1) \mathbf{x}_N(n-1) \quad \dots \quad \mathbf{e}(n-L+1) \mathbf{x}_N(n-L+1) \},$$

iar vectorul \mathbf{a} este dat de relația

$$\mathbf{a}' = [a(1) \quad a(2) \quad \dots \quad a(L)], \quad (4.3)$$

\mathbf{a} reprezentând vectorul cu coeficienți de ponderare a elementelor matricei $\mathbf{M}_N(n)$.

De obicei se utilizează valori impare pentru parametrul L , acesta purtând denumirea de **lungime a ferestrei de ordonare**. Linia i a matricei $\mathbf{M}_N(n)$ este o linie care poate fi scrisă sub forma

$$\Gamma \{ \mathbf{e}(n) \cdot \mathbf{x}(n-i+1) \quad \mathbf{e}(n-1) \cdot \mathbf{x}(n-i) \quad \dots \quad \mathbf{e}(n-L+1) \cdot \mathbf{x}(n-i-L+2) \}, \quad (4.4)$$

unde $i = 1, \dots, N$. Se observă că fiecare coeficient de predicție este calculat prin aplicarea relației (4.1). Dacă toți coeficienții vectorului \mathbf{a} sunt egali cu 1, filtrele OS astfel obținute poartă numele de **filtre OS netede**.

Ieșirea filtrelor OSLMS este obținută pe baza acelorași relații ca și în cazul general

$$y(n) = \mathbf{w}_N^T(n) \cdot \mathbf{x}_N(n), \quad (4.5)$$

iar eroarea de predicție este dată de relația

$$e(n) = d(n) - y(n). \quad (4.6)$$

4.1.3. Tipuri de filtre OSLMS

4.1.3.1. Filtrul median LMS (MLMS)

Acest tip de filtru este definit pe baza relației (4.7) prezentată în [30],[31],[41]

$$\begin{cases} a\left(\frac{L+1}{2}\right) = 1 \\ a(i) = 0, \text{ pentru } i \neq \frac{L+1}{2} \end{cases} \quad (4.7)$$

O notație echivalentă pentru filtrul MLMS este

$$\mathbf{w}_N(n+1) = \mathbf{w}_N(n) + \alpha \cdot \text{med}\{\mathbf{e}(n)\mathbf{x}_N(n)\}_L, \quad (4.8)$$

unde $\text{med}\{\mathbf{e}(n)\mathbf{x}_N(n)\}_L$ este un vector coloană de N linii care conține elementele din mijloc ale lui

$$\{\mathbf{e}(n)\mathbf{x}_N(n) \quad \mathbf{e}(n-1)\mathbf{x}_N(n-1) \quad \dots \quad \mathbf{e}(n-L+1)\mathbf{x}_N(n-L+1)\}.$$

Ca și în cazul utilizării algoritmului LMS, în situația utilizării unui filtru de tip MLMS convergența coeficienților de predicție este influențată de alegerea parametrului de adaptare α , fapt ilustrat în figura 4.2. Se observă că odată cu creșterea valorii parametrului de adaptare convergența coeficienților filtrului este mai bună, însă stabilitatea valorilor coeficienților de predicție este mai mică rezultând că în alegerea valorii parametrului de adaptare trebuie ținut seama de ambele aspecte.

Programul realizat de autor care a permis implementarea filtrului MLMS este prezentat în Anexa IV.1, rezultatele fiind obținute în mod similar prin medierea statistică a 200 de evenimente și alegerea unei lungimi a ferestrei de 3 eșantioane.

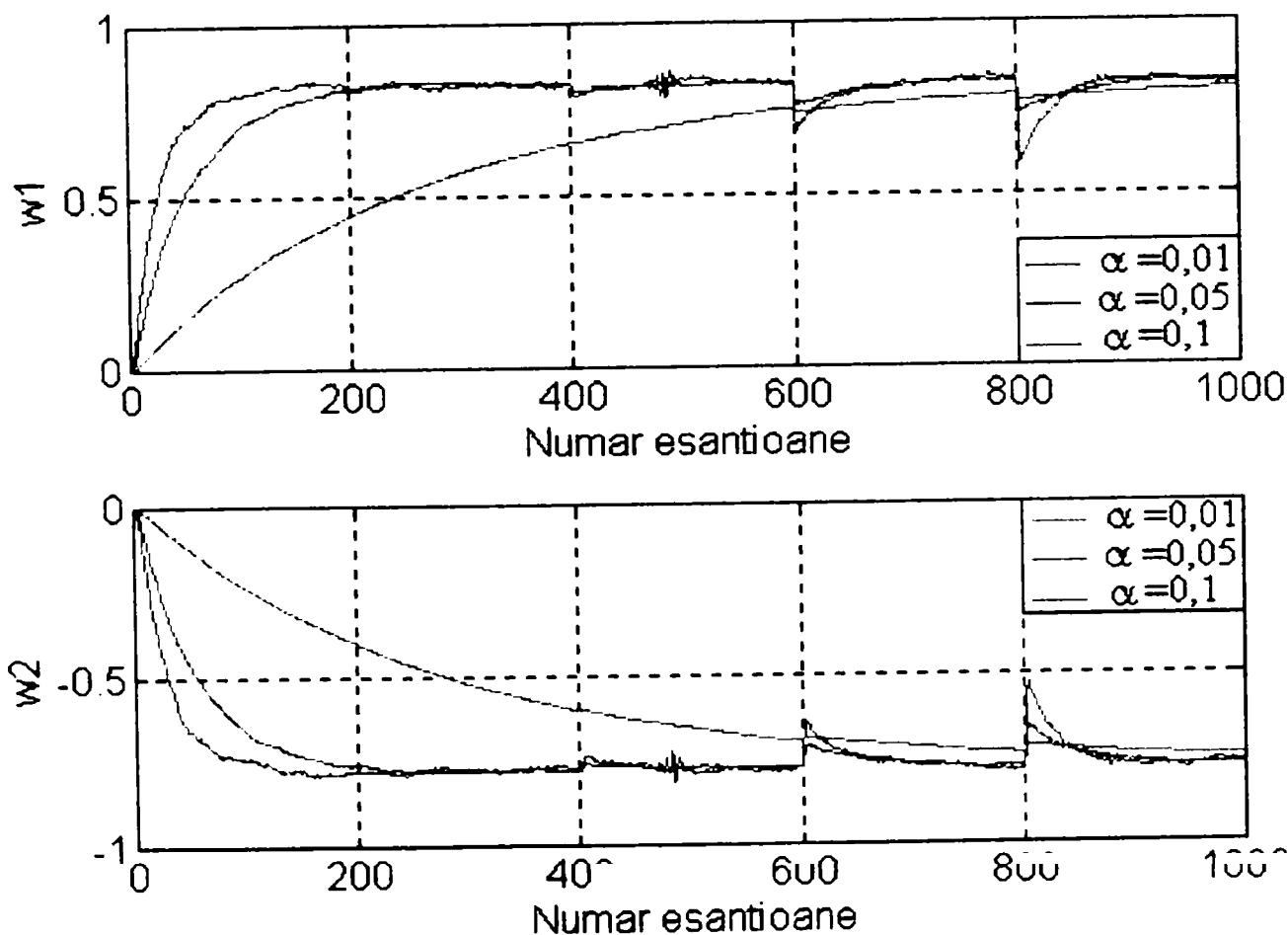


Fig. 4.2. Variația coeficienților de predicție în cazul utilizării filtrului MLMS.

4.1.3.2. Filtrul medie LMS (ALMS)

Acest tip de filtru din categoria OSLMS se obține dacă se aleg coeficienții de ponderare

$$\mathbf{a}(i) = \frac{1}{L}, i = 1, \dots, L. \quad (4.9)$$

Acesta este singurul caz în care algoritmi liniari și algoritmi OSLMS coincid.

În figura 4.3. este prezentată forma de variație a coeficienților de predicție în situația utilizării unui filtru de tip ALMS pentru aceleași valori ale parametrului de adaptare α și aceeași lungime a ferestrei de 3 eșantioane.

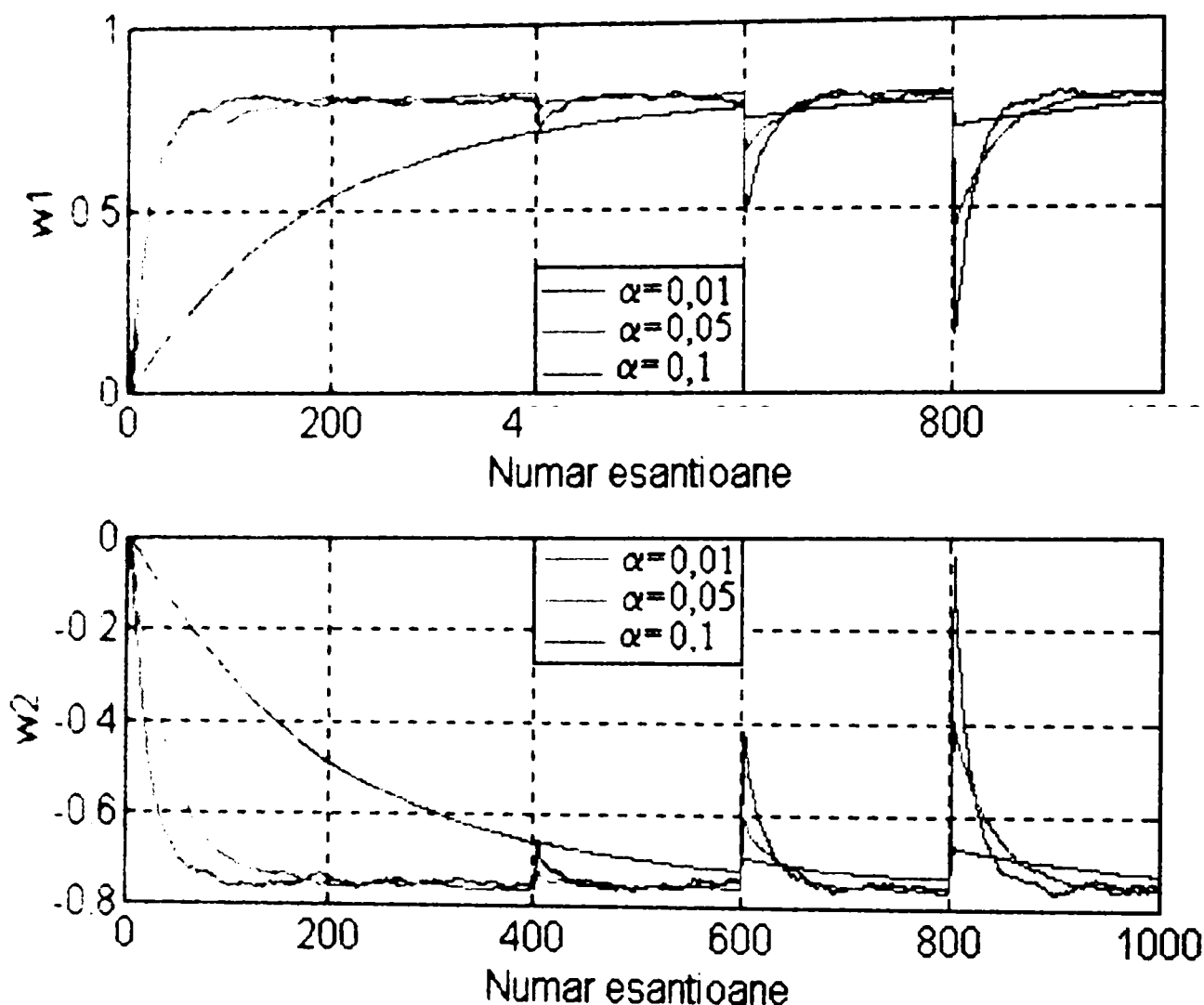


Fig. 4.3. Variația coeficienților de predicție în cazul utilizării unui filtru de tip ALMS.

Ca și în cazul filtrării MLMS, în cazul filtrării ALMS se observă că odată cu creșterea valorii parametrului de adaptare convergența filtrului este mai bună, însă stabilitatea valorilor coeficienților de predicție este mai mică, influența impulsurilor fiind mult mai mare în acest caz.

4.1.3.3. Algoritmul "mediei ordonate" LMS

Acest tip de filtru este definit de relațiile

$$\begin{cases} a(i) = 0, & i = 1, \dots, M \\ a(i) = \frac{1}{L - 2M}, & i = M + 1, \dots, L - M \\ a(i) = 0, & i = L - M + 1, \dots, L \end{cases} \quad (4.10)$$

Acest tip de filtru reprezintă o aplicare a unei operații de mediere asupra estimatorului gradientului cu valori variabile ale parametrului M . Pentru valori particulare ale parametrului M se pot obține precedentele două tipuri de filtre OSLMS. Astfel, pentru $M = \frac{L-1}{2}$ se obține filtrarea MLMS, iar pentru $M=0$ se obține filtrarea ALMS.

4.1.4. Studiul caracteristicilor filtrelor OSLMS

O problemă care apare în cazul folosirii filtrelor OSLMS este alegerea lungimii ferestrei L datorită faptului că transformările de ordonare implicate necesită un volum de calcule suplimentar.

În general, cu excepția procesului de inițializare, se schimbă un singur eșantion la fiecare pas, celelalte rămânând nemodificate. Din acest motiv, operația de ordonare necesară reactualizării fiecărui coeficient necesită un total de $L-1$ comparații. Pentru N coeficienți de predicție rezultă că sunt necesare $N \cdot \log_2(L-1)$ comparații pentru fiecare reactualizare.

Dacă se utilizează algoritmul MLMS această operație de sortare reprezintă singurul volum de calcule suplimentar necesar, în timp ce pentru implementarea filtrului general OSLMS apar suplimentar $N \cdot L$ înmulțiri și $N \cdot (L-1)$ adunări pe iterație. Aceasta duce la o creștere a volumului de calcule suplimentar direct proporțională cu numărul de coeficienți de predicție N și cu lungimea ferestrei L .

Influența alegerii lungimii ferestrei L asupra formei de variație a coeficienților de predicție în situația utilizării filtrului MLMS respectiv ALMS este prezentată în figurile 4.4 respectiv 4.5.

Se observă că în cazul filtrării MLMS odată cu creșterea lungimii ferestrei se obține o mai mică variație a coeficienților de predicție în momentul apariției unui impuls, din acest punct de vedere fiind recomandabilă utilizarea unei ferestre cât mai mari. Aceasta se reflectă, așa cum s-a arătat anterior asupra volumului de calcule și implicit asupra timpului de rulare a programului.

În cazul filtrării ALMS se observă că lungimea ferestrei practic nu influențează amplitudinea variației coeficienților de predicție, aceasta fiind mult mai mare decât în cazul filtrării MLMS și aproximativ aceeași indiferent de lungimea ferestrei. Pe de altă parte se observă o anumită întârziere care depinde de lungimea ferestrei. Rezultă că filtrul ALMS nu este recomandabil a se utiliza în situații în care zgomotul prezent nu are o distribuție gaussiană.

Influența lungimii ferestrei asupra timpului de rulare, exprimat în secunde, a programului este prezentată în Tabelul 4.1 pentru filtrele MLMS, respectiv ALMS.

Se observă că în ambele cazuri odată cu creșterea lungimii ferestrei crește și timpul de rulare, însă timpul necesar implementării algoritmului MLMS este întotdeauna mai mare decât cel necesar implementării algoritmului ALMS datorită operațiilor de comparație suplimentare, necesare determinării valorii mediane.

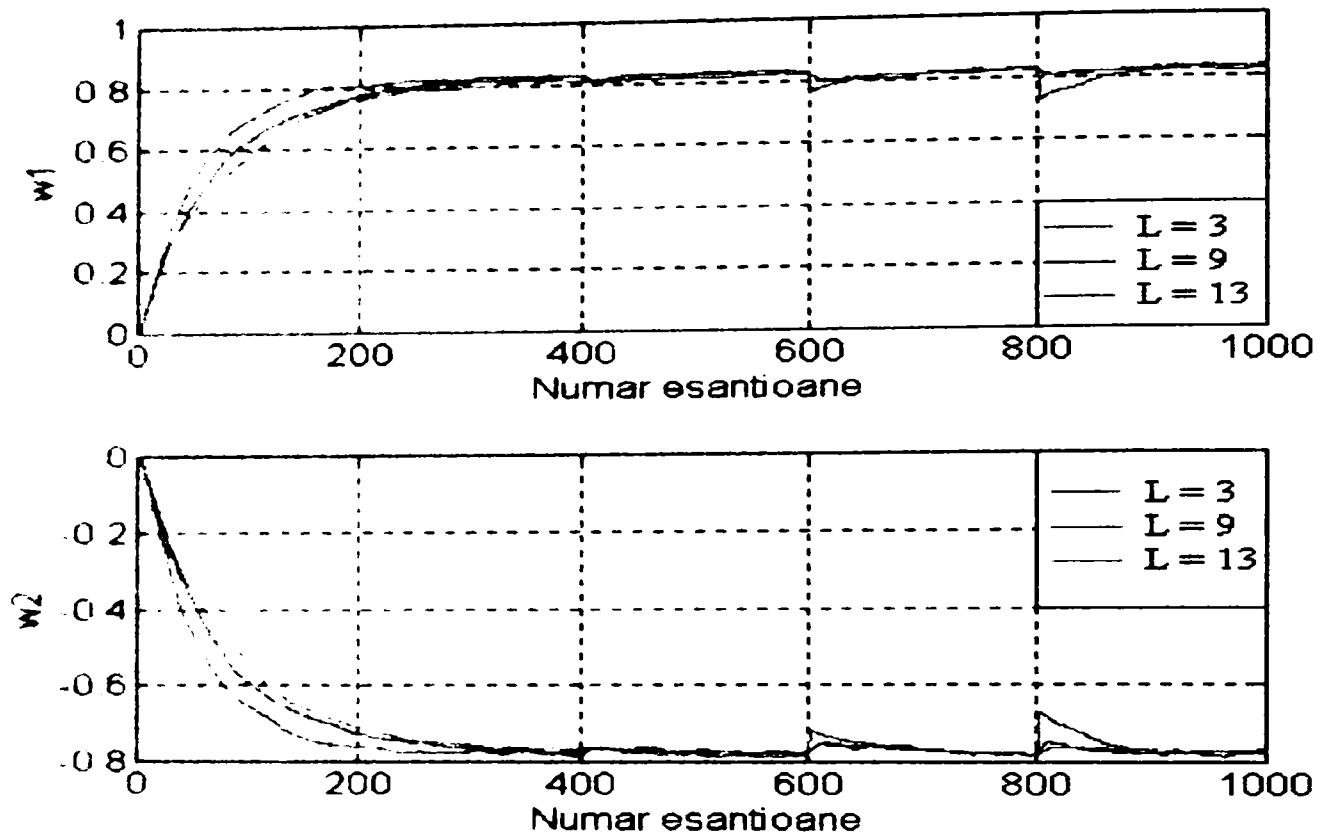


Fig. 4.4. Influența lungimii ferestrei asupra variației coeficienților de predicție în cazul filtrului MLMS.

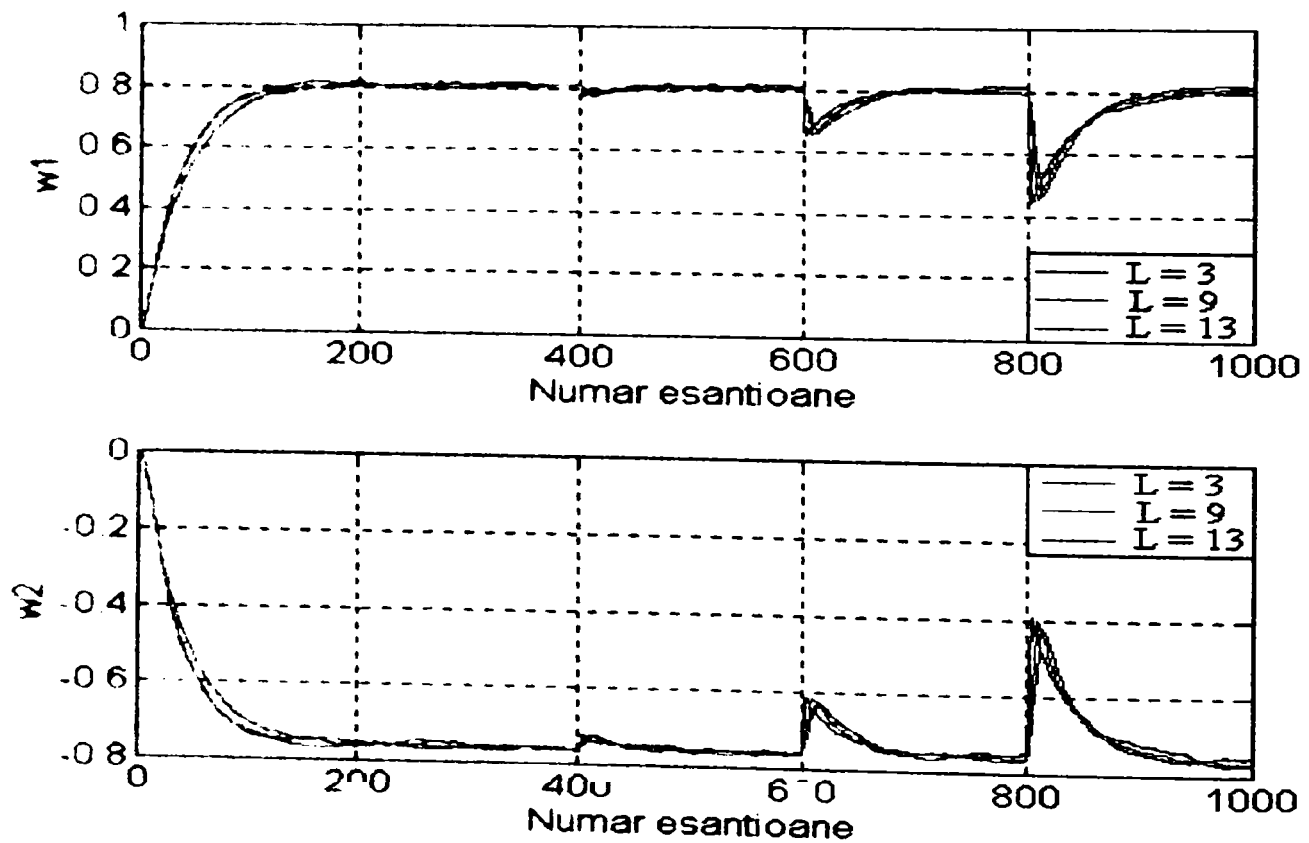


Fig. 4.5. Influența lungimii ferestrei asupra variației coeficienților de predicție în cazul filtrului ALMS.

Tabelul 4.1.

Influența lungimii ferestrei asupra timpului de rulare exprimat în secunde în cazul filtrelor MLMS și ALMS.

	L = 3	L = 9	L = 13
Filtru MLMS	511	555	575
Filtru ALMS	405	429	441

În ceea ce privește convergența, se observă din figurile 4.4 și 4.5 că filtrul ALMS este mai rapid convergent decât filtrul MLMS.

O altă problemă în utilizarea filtrelor OSLMS constă în procesul de inițializare a algoritmului, care implică alegerea valorilor inițiale ale coeficienților de predicție precum și a matricei \mathbf{M}_N [37]. Ca și în cazul filtrării liniare bazată pe algoritmul LMS, inițializarea coeficienților de predicție se poate face cu relația

$$\mathbf{w}_N(0) = 0.$$

Pentru inițializarea matricei \mathbf{M}_N există două procedee utilizate :

- a) alegerea valorilor inițiale nule;
- b) extensia valorilor inițiale.

Ca și în cazul utilizării algoritmului LMS alegerea unuia dintre cele două procedee nu este critică, având o influență esențială doar asupra primelor iterații.

4.1.5. Comutarea algoritmilor OSLMS în funcție de natura zgomotelor perturbatoare

Așa cum s-a prezentat anterior, în funcție de nivelul și tipul zgomotelor prezente la intrarea unui sistem este recomandabilă utilizarea unui anumit filtru. S-a pus problema realizării unui filtru OSLMS care să aibă performanțele îmbunătățite, indiferent de zgomotele prezente la intrare. O astfel de posibilitate este de a adapta tipul filtrului OSLMS în funcție de un parametru care este utilizat în scopul detectării prezenței unui impuls. Acest parametru, notat cu γ , este măsurat la fiecare iterație utilizând gradientul estimat în L puncte ale ferestrei. Parametrul γ indică prezența sau absența unui impuls în interiorul ferestrei și în funcție de decizia luată algoritmul comută între aplicarea algoritmului MLMS dacă se consideră că există un impuls în interiorul ferestrei sau a algoritmului ALMS dacă nu a fost detectată prezența impulsului [31],[41].

Considerând că x_1, x_2, \dots, x_L sunt eșantioanele semnalului de intrare din fereastra de lungime L, iar $x_{(1)} \leq x_{(2)} \leq \dots \leq x_{(L)}$ reprezintă cele L eșantioane ordonate, o primă variantă de definire a parametrului γ , prezentată în [31] este dată de relația

$$\gamma_1 = \frac{\frac{1}{L} \cdot \sum_{i=1}^L (x_i - \bar{x})^4}{\left[\frac{1}{L} \cdot \sum_{i=1}^L (x_i - \bar{x})^2 \right]^2}, \quad (4.11)$$

unde $\bar{x} = \frac{1}{L} \sum_{i=1}^L x_i$ reprezintă media celor L eșantioane.

În figura 4.6 este prezentată forma de variație a coeficienților de predicție precum și a parametrului γ în situația definirii parametrului γ pe baza relației (4.11). Se observă că variația coeficienților de predicție în situația utilizării filtrului adaptiv este similară cu cea din situația utilizării algoritmului MLMS (figura 4.4), iar viteza de convergență este mai mare, ca în cazul utilizării algoritmului ALMS (figura 4.5). Programul realizat de autor care a permis obținerea acestei reprezentări este prezentat în Anexa 4.2 rezultatele fiind obținute în urma medierii statistice a 200 de evenimente.

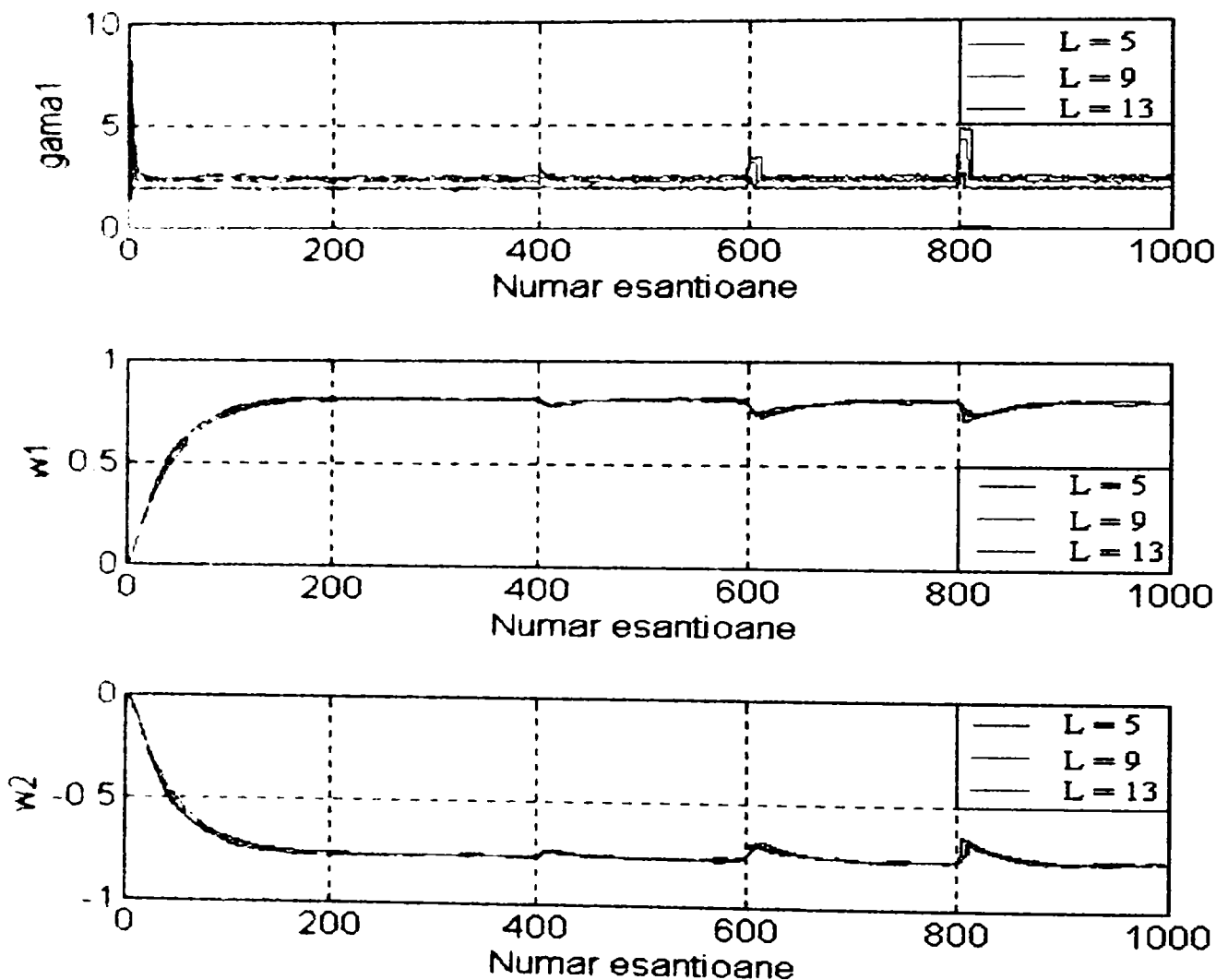


Fig. 4.6. Variația parametrului γ , și a coeficienților de predicție în situația utilizării filtrului adaptiv OSLMS.

A doua variantă de definire a parametrului γ , prezentată în [31], este dată de relația

$$\gamma_2 = \frac{x(i) - x(i)}{\sum_{i=1}^L |x(i) - x(i+1)|} \quad (4.12)$$

Utilizând această variantă de definire a parametrului γ , pe baza programului realizat de autor și prezentat în Anexa 4.3 s-a obținut forma de variație a parametrului γ și a coeficienților de predicție prezentată în figura 4.7.

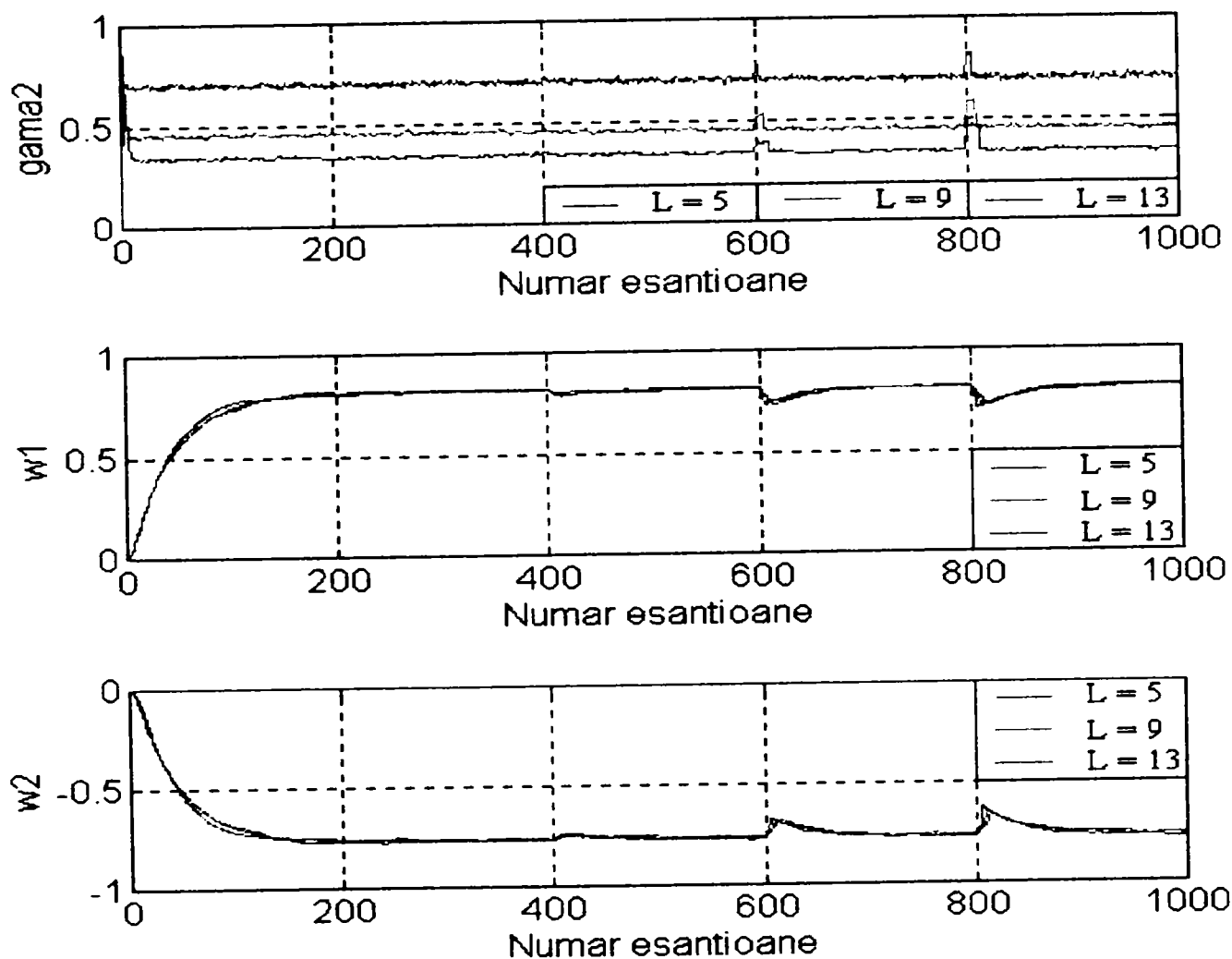


Fig. 4.7. Variația parametrului γ_2 și a coeficienților de predicție în situația utilizării filtrului adaptiv OSLMS.

Se observă că **alegerea uneia din cele două variante de definire a parametrului γ nu aduce îmbunătățiri esențiale în ceea ce privește forma de variație a coeficienților de predicție, dar a doua variantă prezintă două avantaje:**

- **amplitudinea impulsului în situația utilizării lui γ_2 este constantă indiferent de lățimea ferestrei;**

- **calculele necesare implementării acestui algoritm sunt mai simple, fapt rezultat din Tabelul 4.2 în care sunt prezentați timpii necesari implementării ambelor variante, pentru diferite valori ale lățimii ferestrei în situația medierii statistice a 200 de evenimente. Se observă că timpii necesari implementării celei de-a doua variante sunt mai mici decât în situația implementării primei variante, totuși timpii obținuți sunt mult mai mari decât în situația implementării algoritmului LMS datorită operațiilor de comparație suplimentare, precum și determinării valorii parametrului γ .**

Tabelul 4.2.

Influența parametrului γ asupra timpului de rulare în cazul filtrelor adaptive OSLMS .

	L = 5	L = 9	L = 13
Parametrul γ_1	2155	2730	3908
Parametrul γ_2	2134	2693	3786
Parametrul γ_3	1978	2401	3425

Pe baza celor prezentate, autorul a propus o metodă de reducere a volumului de calcule necesar, în situația utilizării parametrului γ_1 . Această metodă se bazează pe calculul mediei eșantioanelor din fereastra de date de intrare la momentul n cunoscând media eșantioanelor din fereastra de date de intrare de la momentul $n-1$.

Considerând că $x(n-L), x(n-L+1), \dots, x(n-1)$ sunt eșantioanele din fereastra de date de la momentul $n-1$, având media $\bar{x}_L(n-1)$ și $x(n-L+1), x(n-L+2), \dots, x(n)$ sunt eșantioanele din fereastra de date de la momentul n , având media $\bar{x}_L(n)$, relația de recurență dintre cele două medii este

$$\bar{x}_L(n) = \bar{x}_L(n-1) - \frac{x(n-L) - x(n)}{L} \quad (4.13)$$

Rezultă că, în cazul metodei propuse de autor, pentru determinarea valorii medii a eșantioanelor din fereastra de lungime L sunt necesare două operații de adunare și o operație de multiplicare. Deoarece în cazul determinării valorii medii a eșantioanelor dintr-o fereastră sunt necesare $L-1$ operații de adunare și o operație de înmulțire, se observă că în situația utilizării ferestrelor de lungimi mari, metoda propusă de autor permite reducerea volumului de calcule necesar determinării valorii parametrului γ_1 . Acest fapt este ilustrat în Tabelul 4.2, în care timpul de rulare obținut utilizând metoda propusă corespunde parametrului γ_1 .

Această metodă permite, de asemenea, reducerea volumului de calcule în situația utilizării unui filtru de tip ALMS. Din relațiile (4.1) și (4.9) se poate deduce că o formă echivalentă a relațiilor de reactualizare a coeficienților de predicție este

$$\mathbf{w}_N(n+1) = \mathbf{w}_N(n) + \alpha \cdot \text{media} \{e(n) \mathbf{x}_N(n)\}_L \quad (4.14)$$

Pentru coeficientul w_i , relația (4.14) poate fi scrisă sub forma

$$w_i(n+1) = w_i(n) + \alpha \cdot \text{media} \{e(n)x(n-i+1), e(n-1)x(n-i), \dots, e(n-L+1)x(n-i-L+2)\}$$

Notând

$$\text{media}(n) = \text{media} \{e(n)x(n-i+1), e(n-1)x(n-i), \dots, e(n-L+1)x(n-i-L+2)\}$$

și

$$\text{media}(n-1) = \text{media} \{e(n-1)x(n-i), e(n-2)x(n-i-1), \dots, e(n-L)x(n-i-L+1)\},$$

relația de recurență între cele două medii este

$$\text{media}(n) = \text{media}(n-1) - \frac{e(n-L)x(n-i-L+1) - e(n)x(n-i+1)}{L} \quad (4.15)$$

Se observă că în situația utilizării metodei prezentate de autor reactualizarea mediei secvenței $\{e(n)x(n-i+1), e(n-1)x(n-i), \dots, e(n-L+1)x(n-i-L+2)\}$, pentru calculul unui coeficient de predicție, necesită un număr de 2 operații de multiplicare și 2 operații de

adunare față de 2 operații de multiplicare și L-1 operații de adunare, produsul $e(n-L)x(n-i-L+1)$ fiind calculat la momentul n. Din cele prezentate rezultă că în cazul filtrării ALMS volumul de calcule poate fi redus utilizând metoda propusă de autor, metoda fiind eficientă în special în situația utilizării ferestrelor de lungime mare.

4.1.6. Filtre adaptive MALMS

Așa cum s-a prezentat anterior, filtrul MLMS se caracterizează prin faptul că influența impulsurilor cu amplitudine mare și durată scurtă asupra formei de variație a coeficienților este mai mică decât în cazul filtrului ALMS, fiind astfel recomandat a se utiliza în situația prezenței acestui tip de impulsuri în semnalul de la intrarea filtrului. În ceea ce privește viteza de convergență, filtrul ALMS este mai rapid convergent decât filtrul MLMS. Datorită acestui fapt este recomandabil a se utiliza filtrarea de tip ALMS în situația în care semnalul de la intrarea filtrului nu este perturbat de impulsuri cu amplitudine mare și durată scurtă.

Metoda de realizare a unui filtru adaptiv prezentată în paragraful anterior se bazează pe utilizarea unui parametru care detecta prezența unui impuls în fereastra de date de la intrarea filtrului și în funcție de prezența sau absența impulsului se comută pe o filtrare de tip MLMS sau de tip ALMS. Această metodă prezintă dezavantajul că pe durata utilizării unui anumit tip de filtru avantajul oferit de celălalt tip de filtru dispare.

Observând din figurile 4.6 și 4.7 că valoarea parametrului de detecție a prezenței impulsului în interiorul ferestrei de date de la intrarea filtrului depinde de amplitudinea impulsului autorul a propus o nouă metodă de obținere a unui filtru adaptiv bazată pe ponderarea celor două tipuri de filtrări în funcție de amplitudinea parametrului de detecție a prezenței impulsului. Metoda utilizează un coeficient de ponderare a fiecărui tip de filtrare, coeficient care își modifică valoarea în funcție de amplitudinea parametrului de detecție a prezenței impulsului în fereastra de date de la intrarea filtrului, γ . Relația de reactualizare a coeficienților de predicție în cazul metodei propuse de autor este

$$\mathbf{w}_N(n+1) = \mathbf{w}_N(n) + \alpha \cdot \left[(1-q) \cdot \text{med}\{e(n) \mathbf{x}_N(n)\}_L + q \cdot \text{media}\{e(n) \mathbf{x}_N(n)\}_L \right]. \quad (4.16)$$

Se observă că pentru valoarea parametrului $q=0$ se obține un filtru MLMS, iar pentru valoarea $q=1$ se obține filtrul ALMS. Pentru valori ale parametrului q cuprinse în intervalul $(0,1)$ filtrul obținut combină caracteristicile filtrelor median și medie fiind denumit de autor filtru MALMS. În figura 4.8 este prezentată forma de variație a coeficienților filtrului adaptiv MALMS pentru diferite valori ale parametrului q , în situația utilizării unei ferestre de lungime 9 eșantioane.

Autorul a propus ca ponderea fiecărui tip de filtru, deci a parametrului q , să se modifice în funcție de valoarea parametrului de detecție a prezenței impulsului, γ . Considerând că se obține o valoare a parametrului $\gamma \geq \gamma_{\max}$, aceasta semnifică faptul că amplitudinea impulsului este mare, deci parametrul q trebuie să aibă valoarea $q=0$. Dacă parametrul $\gamma \leq \gamma_{\min}$ se consideră că în fereastra de date nu a fost detectată prezența impulsului deci parametrul q trebuie să aibă valoarea $q=1$. Între valorile γ_{\min} și γ_{\max} s-a ales o variație liniară a parametrului q în funcție de parametrul γ . În aceste condiții, relația de definire a parametrului q în funcție de parametrul γ este

$$q = \begin{cases} 1 & \text{dacă } \gamma \leq \gamma_{\min} \\ \frac{\gamma_{\max} - \gamma}{\gamma_{\max} - \gamma_{\min}} & \text{dacă } \gamma \in (\gamma_{\min}, \gamma_{\max}) \\ 0 & \text{dacă } \gamma \geq \gamma_{\max} \end{cases} \quad (4.17)$$

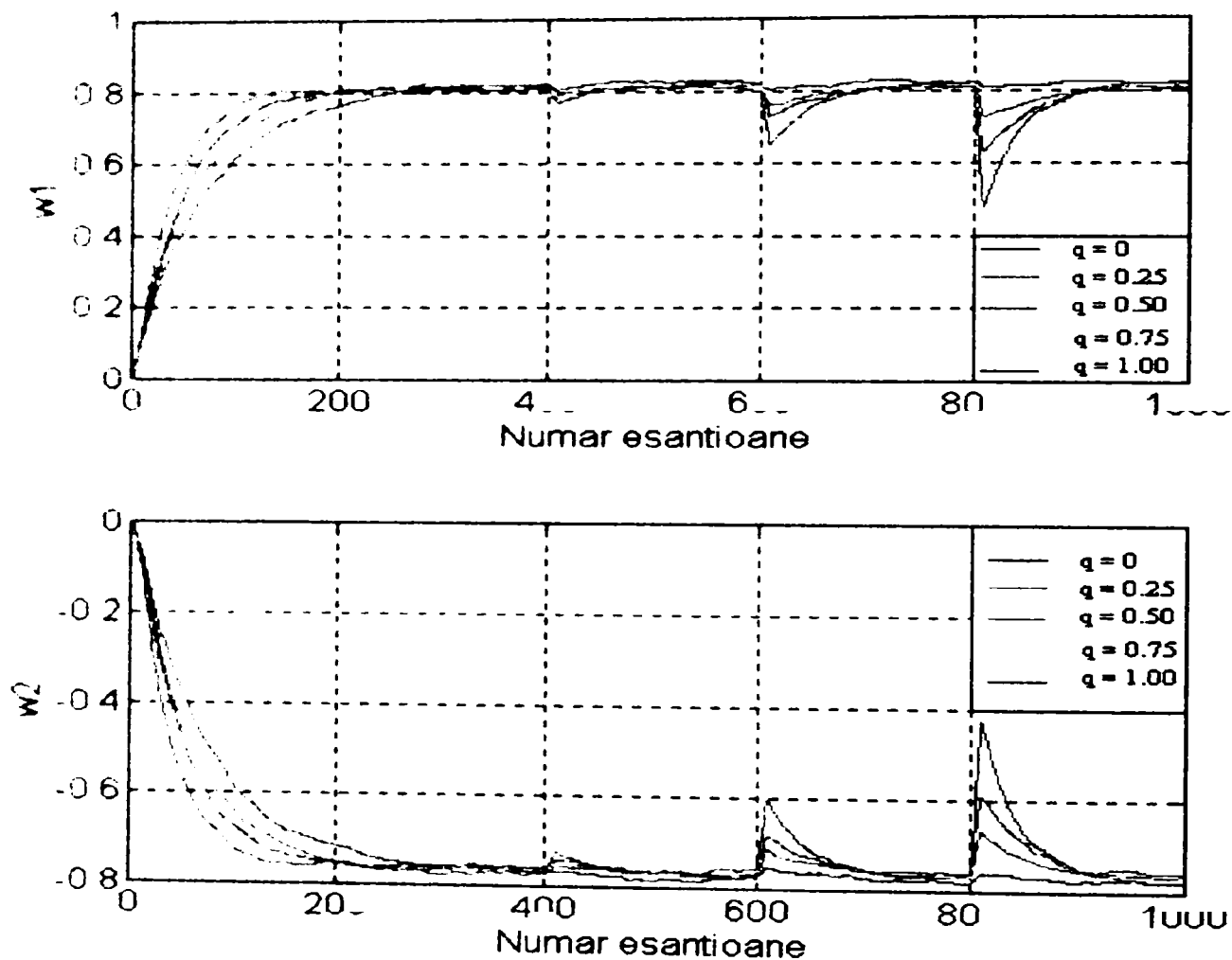


Fig. 4.8. Variația coeficienților de predicție în cazul utilizării unui filtru de tip MALMS.

Valorile γ_{\min} și γ_{\max} se aleg în funcție de amplitudinea estimată a impulsurilor care pot influența semnalul de la intrarea filtrului, precum și de lungimea ferestrei de date.

Avantajul oferit de acest tip de filtru constă în faptul că pentru valori ale parametrului γ cuprinse între γ_{\min} și γ_{\max} este realizată atât o creștere a vitezei de convergență față de filtrul MLMS, cât și o influență mai mică a impulsurilor de amplitudine mare și durată scurtă asupra variației coeficienților de predicție decât în cazul filtrului ALMS.

Dezavantajul filtrului MALMS constă în volumul mai mare de calcule necesar, datorită faptului că trebuie calculată atât valoarea medie cât și valoarea mediană a secvenței $\{e(n) \times_N(n)\}$. Aceasta duce la creșterea timpului de calcul necesar realizării unei iterații, deci

filtrul va putea fi utilizat în timp real pentru procese mai lente decât în situația utilizării filtrelor MLMS și ALMS.

4.2. Filtre IIR-OSLMS

4.2.1. Definirea filtrelor IIR-OSLMS

Așa cum s-a prezentat în capitolul 1, una dintre aplicațiile filtrelor adaptive este identificarea. În acest caz, rolul filtrului adaptiv este de a furniza un model care să reprezinte cea mai bună aproximare a sistemului necunoscut. Există situații când modelul asociat sistemului necunoscut este caracterizat printr-o funcție rațională ai cărei coeficienți se dorește a fi determinați, utilizând filtrul adaptiv. În această situație filtrul adaptiv va fi un filtru IIR, descris de relația (3.16).

Filtrele adaptive IIR pot fi implementate atât în forma directă cât și utilizând structuri lattice, această din urmă metodă fiind prezentată de autor în capitolul precedent. Autorul a prezentat trei algoritmi care pot fi utilizați pentru reactualizarea coeficienților filtrului adaptiv, algoritmul bazat pe utilizarea gradientului, algoritmul Steiglitz-McBride și algoritmul SHARF. Acești algoritmi pot fi utilizați dacă peste semnalul de la ieșirea sistemului necunoscut se suprapune un zgomot cu caracter gaussian.

Dacă peste semnalul de la ieșirea sistemului necunoscut se suprapun impulsuri cu amplitudine mare și durată scurtă, valorile coeficienților filtrului adaptiv IIR pot prezenta oscilații. S-a arătat în paragraful precedent că o posibilitate de a reduce influența impulsurilor cu amplitudine mare și durată scurtă este de a utiliza un filtru OSLMS.

Pe baza celor prezentate, autorul a definit un tip de filtru adaptiv IIR care reactualizează coeficienții pe baza unuia din cei trei algoritmi de identificare, combinat cu o operație de ordonare.

Deoarece la fiecare din cei trei algoritmi filtrul IIR poate fi implementat atât în forma directă, cât și utilizând structuri lattice, rezultă 6 posibilități de reactualizare a coeficienților filtrului adaptiv. Cu notațiile utilizate anterior, relațiile de reactualizare ale coeficienților filtrului adaptiv propuse de autor sunt

1.1. Algoritmul gradientului, forma de implementare directă

$$\begin{pmatrix} \mathbf{a}(n+1) \\ \mathbf{b}(n+1) \end{pmatrix} = \begin{pmatrix} \mathbf{a}(n) \\ \mathbf{b}(n) \end{pmatrix} + \mu \cdot \mathbf{T} \begin{Bmatrix} -e(n)\mathbf{Y}_A(n) \\ e(n)\mathbf{X}_A(n) \end{Bmatrix} \cdot \mathbf{a} \quad (4.18)$$

1.2. Algoritmul gradientului, forma de implementare lattice

$$\begin{pmatrix} \mathbf{k}(n+1) \\ \mathbf{b}(n+1) \end{pmatrix} = \begin{pmatrix} \mathbf{k}(n) \\ \mathbf{b}(n) \end{pmatrix} + \mu \cdot \mathbf{T} \begin{Bmatrix} -e(n)\hat{\mathbf{U}}_A(n) \\ e(n)\mathbf{X}_A(n) \end{Bmatrix} \cdot \mathbf{a} \quad (4.19)$$

2.1. Algoritmul SHARF, forma de implementare directă

$$\begin{pmatrix} \mathbf{a}(n+1) \\ \mathbf{b}(n+1) \end{pmatrix} = \begin{pmatrix} \mathbf{a}(n) \\ \mathbf{b}(n) \end{pmatrix} + \mu \cdot \mathbf{T} \begin{Bmatrix} -c(n)\mathbf{y}(n) \\ c(n)\mathbf{x}(n) \end{Bmatrix} \cdot \mathbf{a} \quad (4.20)$$

2.2. Algoritmul SHARF, forma de implementare lattice

$$\begin{pmatrix} \mathbf{k}(n+1) \\ \mathbf{b}(n+1) \end{pmatrix} = \begin{pmatrix} \mathbf{k}(n) \\ \mathbf{b}(n) \end{pmatrix} + \mu \cdot \mathbf{T} \begin{Bmatrix} -c(n)\mathbf{u}(n) \\ c(n)\mathbf{x}(n) \end{Bmatrix} \cdot \mathbf{a} \quad (4.21)$$

3.1. Algoritmul Steiglitz Mc-Bride, forma de implementare directă

$$\begin{pmatrix} \mathbf{a}(n+1) \\ \mathbf{b}(n+1) \end{pmatrix} = \begin{pmatrix} \mathbf{a}(n) \\ \mathbf{b}(n) \end{pmatrix} + \mu \cdot \mathbf{T} \begin{Bmatrix} -e(n)\mathbf{D}_A(n) \\ e(n)\mathbf{X}_A(n) \end{Bmatrix} \cdot \mathbf{a} \quad (4.22)$$

3.2. Algoritmul Steiglitz Mc-Bride, forma de implementare lattice

$$\begin{pmatrix} \mathbf{k}(n+1) \\ \mathbf{b}(n+1) \end{pmatrix} = \begin{pmatrix} \mathbf{k}(n) \\ \mathbf{b}(n) \end{pmatrix} + \mu \cdot \mathbf{T} \begin{Bmatrix} -e(n)\mathbf{U}_A(n) \\ e(n)\mathbf{X}_A(n) \end{Bmatrix} \cdot \mathbf{a} \quad (4.23)$$

Dacă vectorul \mathbf{a} este definit pe baza relației (4.7), filtrul obținut este un filtru IIR-Median, iar dacă este definit pe baza relației (4.9), filtrul obținut este un filtru IIR-Medie.

4.2.2. Studiul caracteristicilor filtrelor IIR-OSLMS

Pentru studiul caracteristicilor filtrelor IIR-OSLMS, autorul a verificat experimental cei trei algoritmi în cele două forme de implementare. În acest scop, s-au identificat parametrii sistemului necunoscut cu funcția de transfer

$$H(z) = \frac{-0,1 - 1,3z^{-1} + 1,6z^{-2}}{1 - 1,19z^{-1} + 0,7z^{-2}}$$

Acest sistem are coeficienții de reflexie de valori $k_1 = -0,7$ și $k_2 = 0,7$. La intrarea sistemului necunoscut a fost aplicat zgomot cu distribuție gaussiană. Toate simulările au fost realizate cu aceeași valoare a parametrului de adaptare $\mu = 0,05$ și aceeași lungime a ferestrei, de 9 eșantioane.

Autorul a constatat experimental că, pentru aceste valori ale parametrului de adaptare, respectiv ale lungimii ferestrei, un număr de 5000 de eșantioane este suficient pentru ca toți coeficienții filtrului adaptiv să convergă la valorile coeficienților sistemului necunoscut.

Pentru a studia comportarea filtrelor IIR-OSLMS în situația în care apar perturbații sub formă de impulsuri cu amplitudine mare și durată scurtă, la ieșirea sistemului necunoscut s-a considerat că apar 3 impulsuri de amplitudini diferite, 2,3 respectiv 4 unități, situate în pozițiile eșantioanelor cu numerele 2000, 3000 respectiv 4000.

Programele scrise de autor, care permit determinarea formei de variație a coeficienților filtrului adaptiv sunt prezentate în Anexele 4.4 - 4.9. Pentru primii doi algoritmi s-a reprezentat forma de variație a coeficienților filtrului adaptiv IIR, a coeficienților filtrului adaptiv IIR-Medie, respectiv IIR-Median, implementați atât în forma directă, cât și în structura lattice (figurile 4.9 - 4.20).

Autorul a constatat următoarele:

1. În cazul formei de implementare directă, filtrul IIR și filtrul Medie-IIR au proprietăți similare atât în ceea ce privește viteza de convergență, cât și în influența impulsurilor de amplitudine mare și durată scurtă asupra formei de variație a coeficienților. Filtrul Median-IIR prezintă o viteză de convergență mai mică, însă influența impulsurilor de amplitudine mare și durată scurtă asupra formei de variație a coeficienților este mult redusă.

2. În cazul formei de implementare lattice, filtrul IIR și filtrul Medie-IIR au proprietăți similare în ceea ce privește viteza de convergență dar influența impulsurilor de amplitudine mare și durată scurtă asupra formei de variație a coeficienților este mai mare în cazul filtrului

Medie-IIR. Și în acest caz, filtrul Median-IIR prezintă o viteză de convergență mai mică, însă influența impulsurilor de amplitudine mare și durată scurtă asupra formei de variație a coeficienților este mult redusă.

3. Comparând caracteristicile celor doi algoritmi, se observă că viteza de convergență este aproximativ aceeași, indiferent de forma de implementare și de tipul de filtru, în timp ce, în ceea ce privește influența impulsurilor de amplitudine mare și durată scurtă asupra formei de variație a coeficienților, aceasta este mai redusă în cazul algoritmului SHARF.

4. În cazul algoritmului Steiglitz-McBride, autorul a constatat experimental că filtrele IIR-OSLMS propuse nu sunt convergente pentru valorile alese ale parametrului de adaptare, respectiv ale lungimii ferestrei. În figurile 4.21, respectiv 4.22, s-a reprezentat variația coeficienților filtrului adaptiv IIR în cazul utilizării acestui algoritm, în ambele forme de implementare. Formele de variație a coeficienților reprezentate au fost obținute în urma unor încercări repetate deoarece în multe cazuri acest algoritm nu a fost convergent

5. În ceea ce privește volumul de calcule necesar implementării acestor tipuri de filtre, implementarea filtrului IIR necesită cel mai mic volum de calcule, în timp ce, datorită operațiilor de comparare necesare implementării filtrului Median-IIR, acesta necesită cel mai mare volum de calcule.

6. Cele mai bune performanțe în ceea ce privește influența impulsurilor de amplitudine mare și durată scurtă asupra formei de variație a coeficienților le are filtrul Median-IIR, indiferent de de algoritmul utilizat și de forma de implementare. Rezultă că, deși timpul de calcul necesar implementării acestui filtru este cel mai mare, există aplicații în care utilizarea acestui tip de filtru prezintă avantaje în eliminarea influenței impulsurilor de amplitudine mare și durată scurtă. De asemenea, acest tip de filtru poate fi utilizat la identificarea în timp real a parametrilor proceselor lente.

4.3. Filtre combinaționale

4.3.1. Definirea filtrelor combinaționale

În multe domenii de prelucrare a semnalelor, inclusiv prelucrarea imaginilor, o categorie de semnale de mare interes este cea a semnalelor cu nivelul mediei variabil în timp. În particular, schimbările abrupte în nivelul mediu al semnalelor (de exemplu contururile) trebuie menținute și la ieșirea filtrului, caz în care filtrele liniare nu mai sunt recomandabile [30], [31]. De asemenea, dacă zgomotul care poate influența semnalul respectiv are o altă distribuție decât cea gaussiană (de exemplu impulsuri cu amplitudine mare și durată scurtă), filtrele liniare nu permit reconstituirea semnalului dorit.

O clasă de filtre neliniare care îmbunătățesc performanțele obținute sunt filtrele combinaționale (C) prezentate sub această denumire în [29] și întâlnite pentru prima dată sub denumirea de filtre LI [35].

Se consideră vectorul de date

$$\mathbf{x} = (x_1 \ x_2 \ \dots \ x_N)^T,$$

cu N componente aranjate în ordinea achiziționării în timp, aranjare purtând denumirea de ordonare temporală. Uzual parametrul N se alege impar.

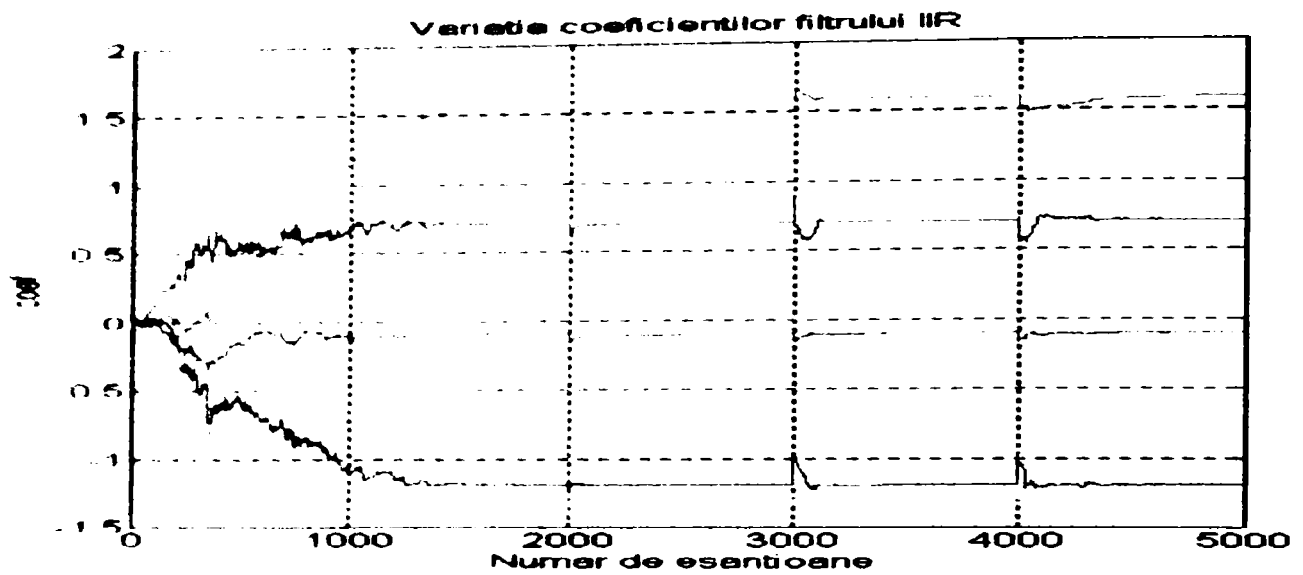


Fig. 4.9. Variația coeficienților filtrului IIR, algoritmul gradientului, forma directă.

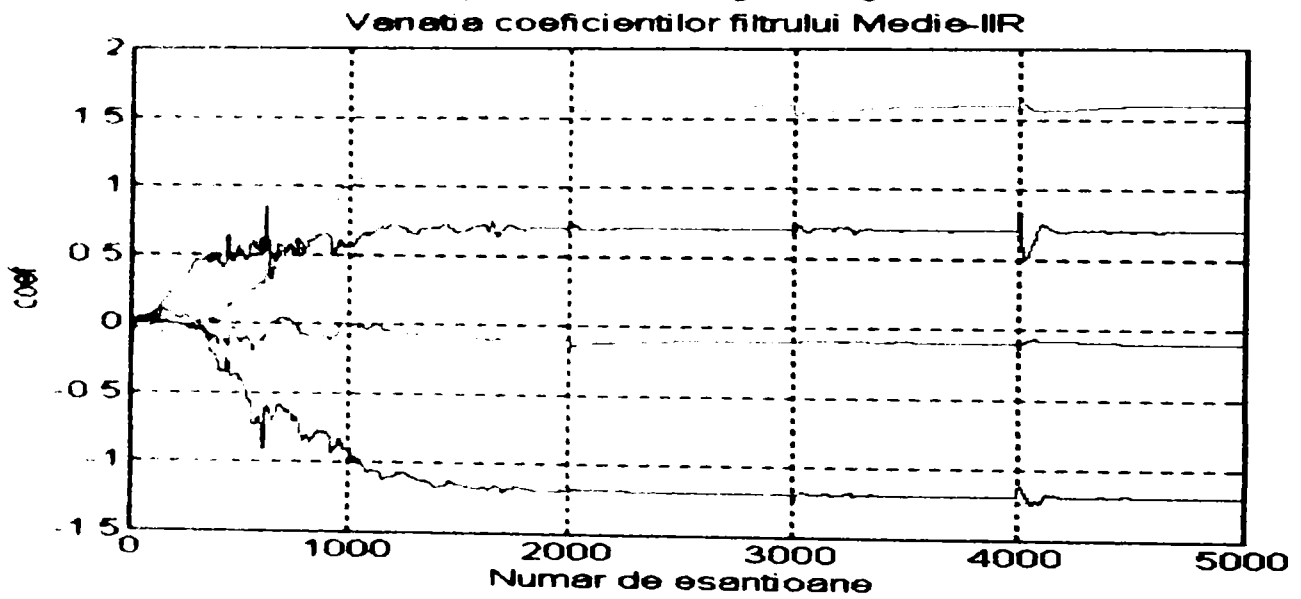


Fig. 4.10. Variația coeficienților filtrului Medie-IIR, algoritmul gradientului, forma directă.

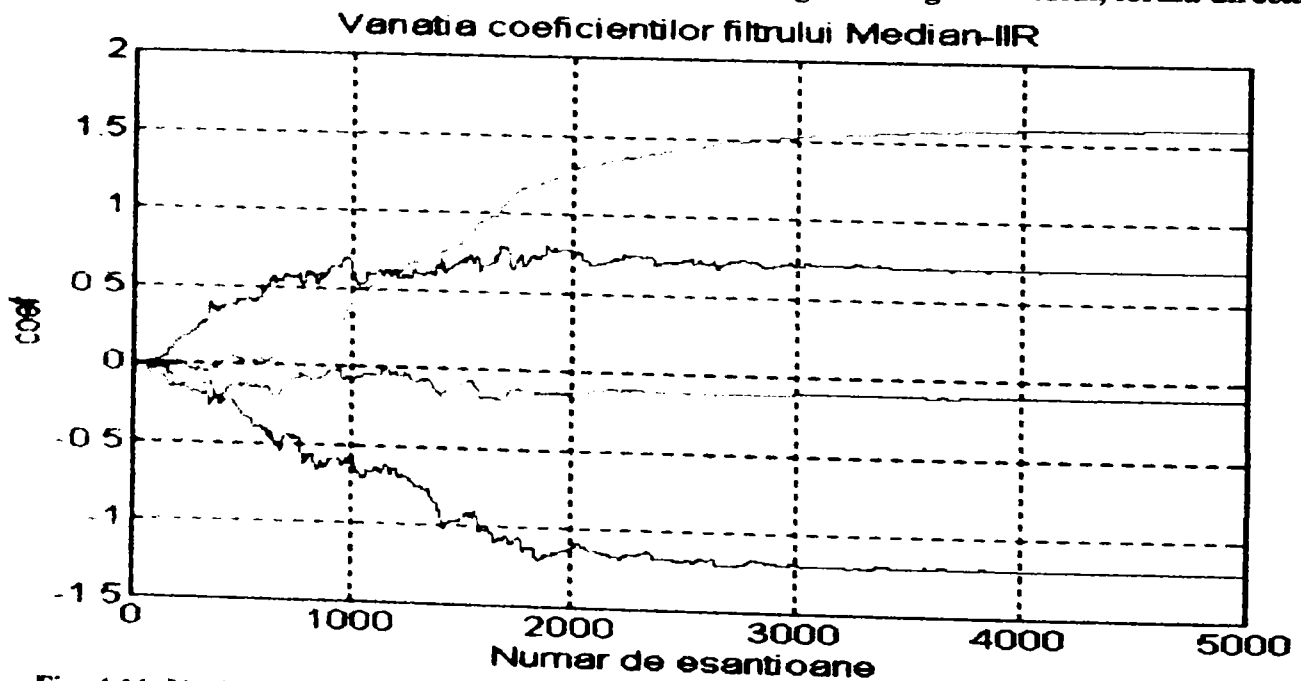


Fig. 4.11. Variația coeficienților filtrului Median-IIR, algoritmul gradientului, forma directă.

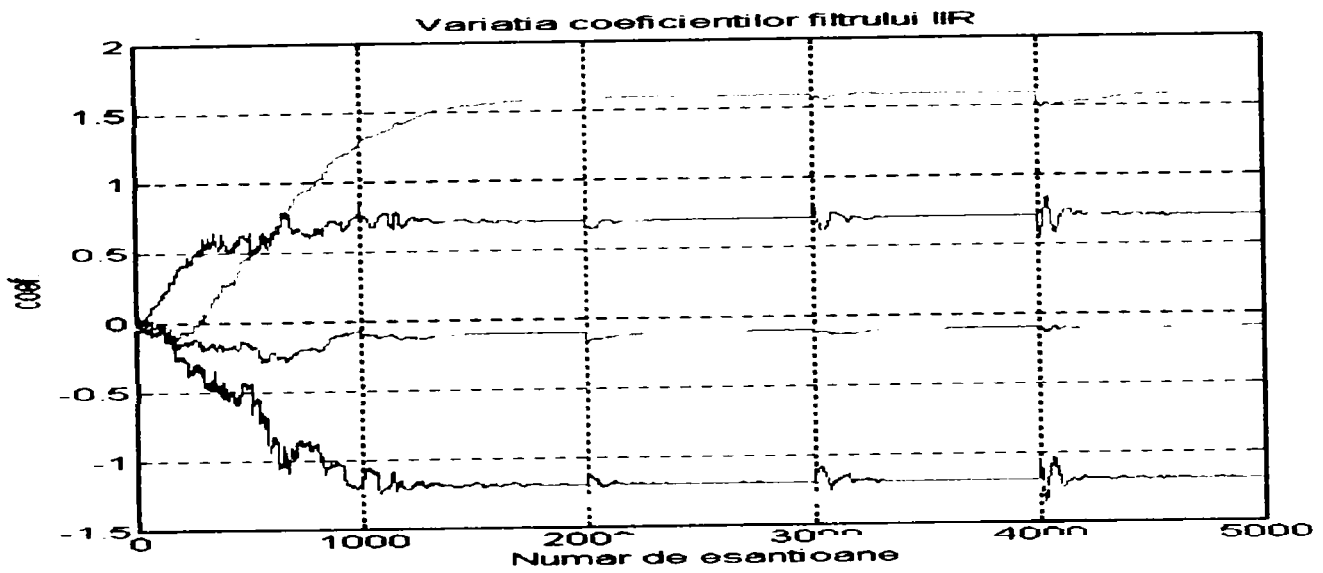


Fig. 4.12. Variația coeficienților filtrului IIR, algoritmul gradientului, forma lattice.

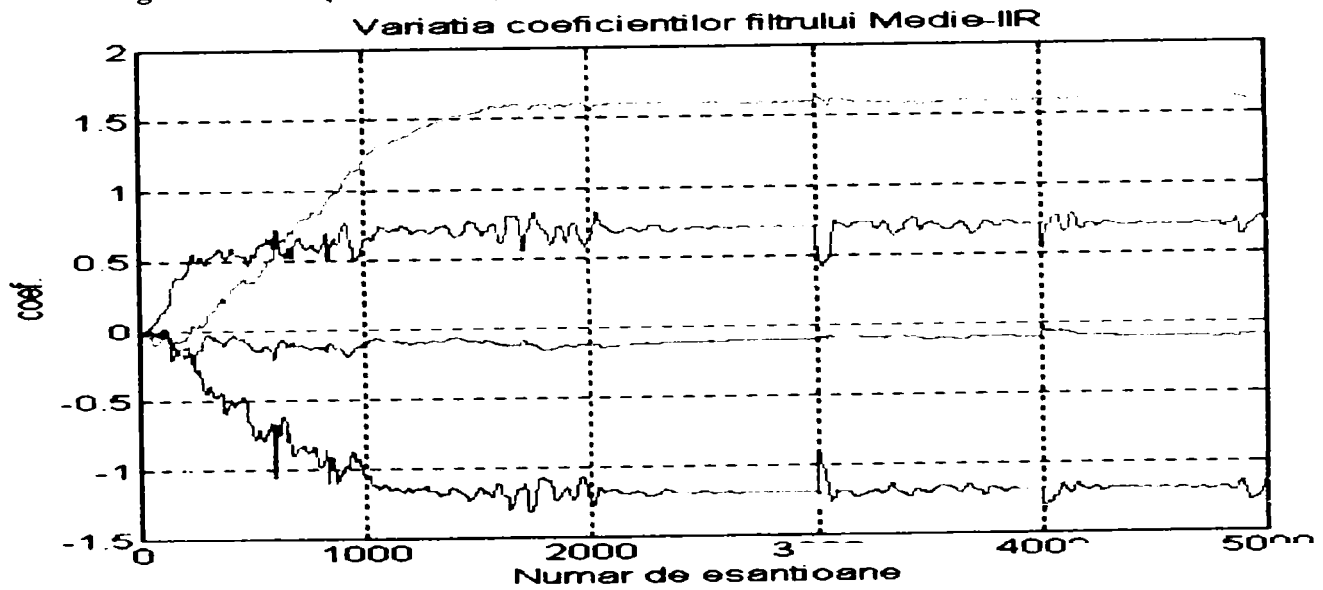


Fig. 4.13. Variația coeficienților filtrului Medie-IIR, algoritmul gradientului, forma lattice.

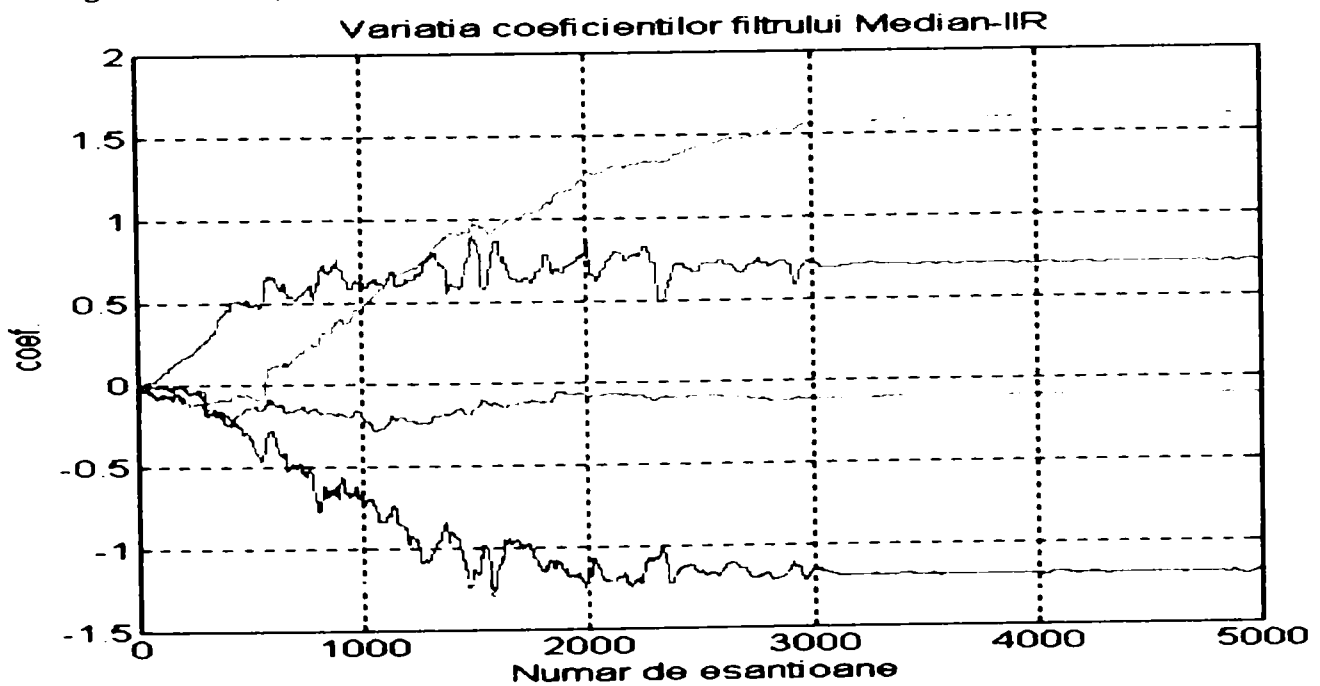


Fig. 4.14. Variația coeficienților filtrului Median-IIR, algoritmul gradientului, forma lattice.

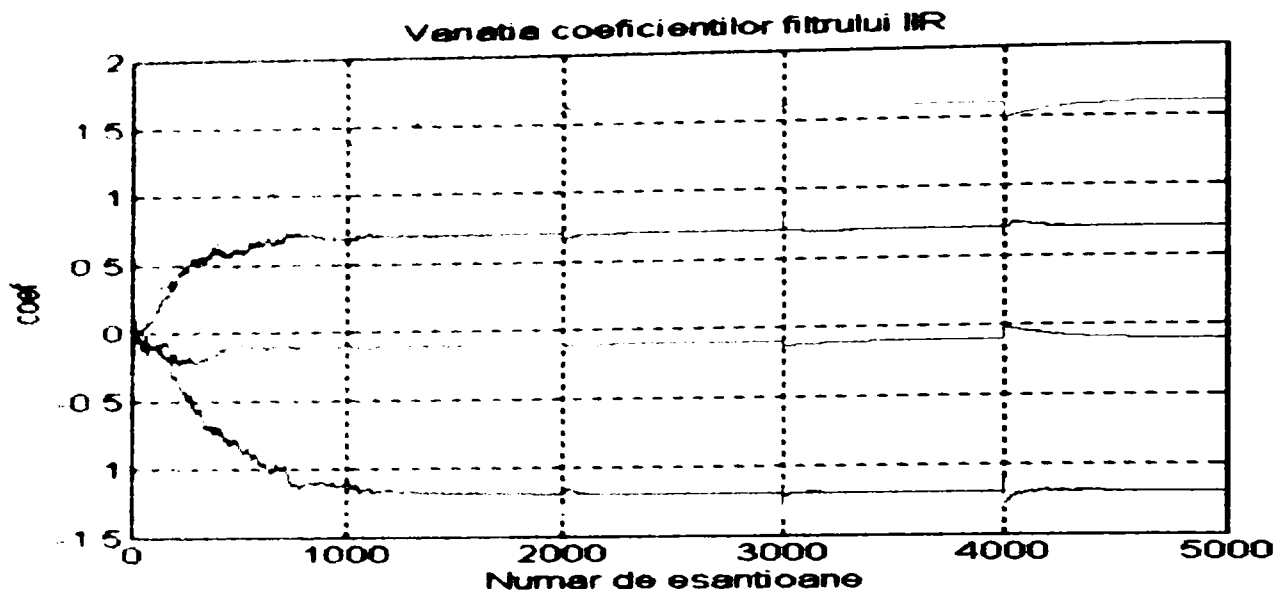


Fig. 4.15. Variația coeficienților filtrului IIR, algoritmul SHARF, forma directă.

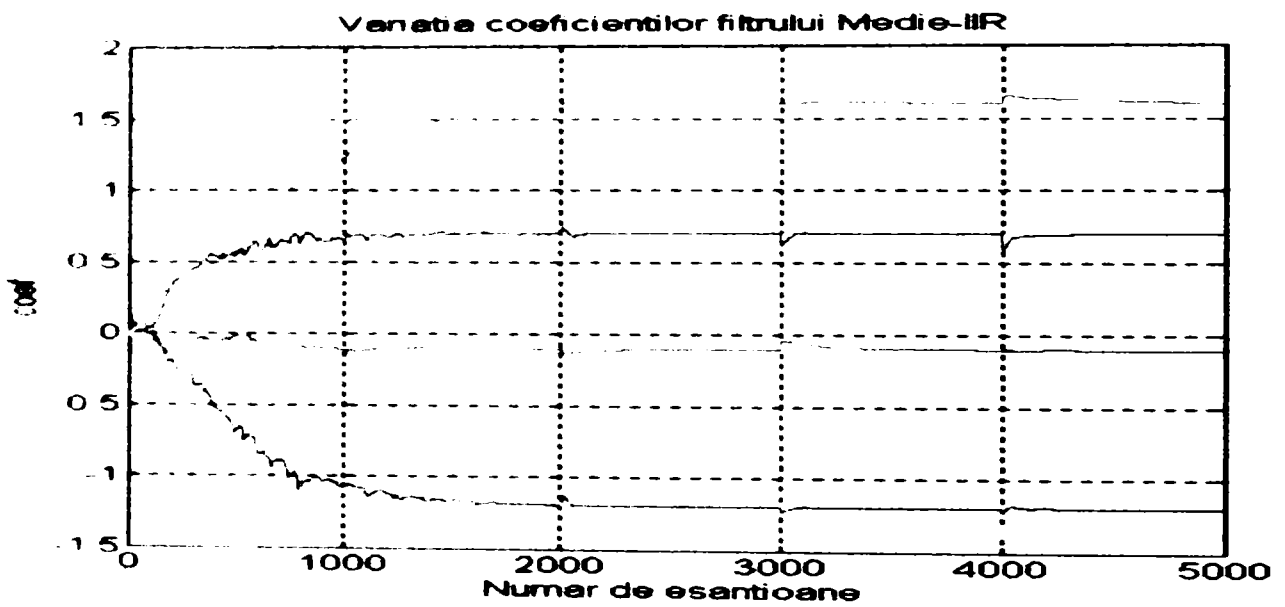


Fig. 4.16. Variația coeficienților filtrului Medie-IIR, algoritmul SHARF, forma directă.

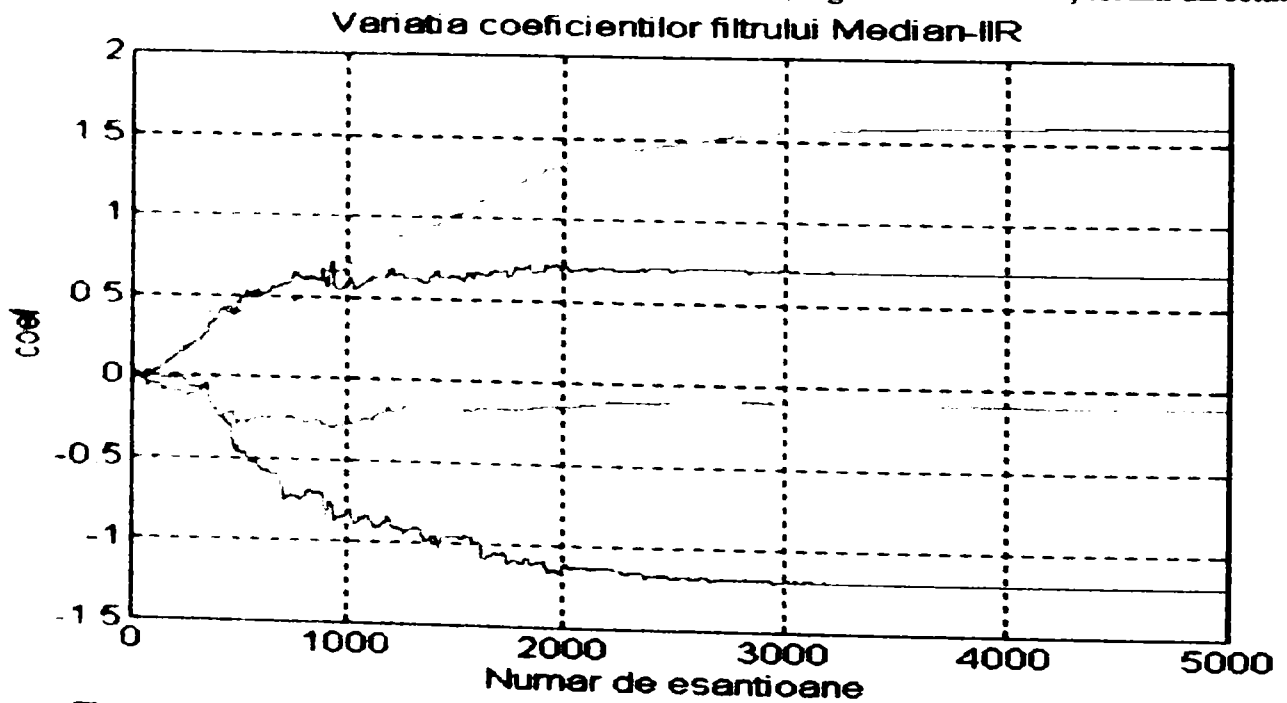


Fig. 4.17. Variația coeficienților filtrului Median-IIR, algoritmul SHARF, forma directă.

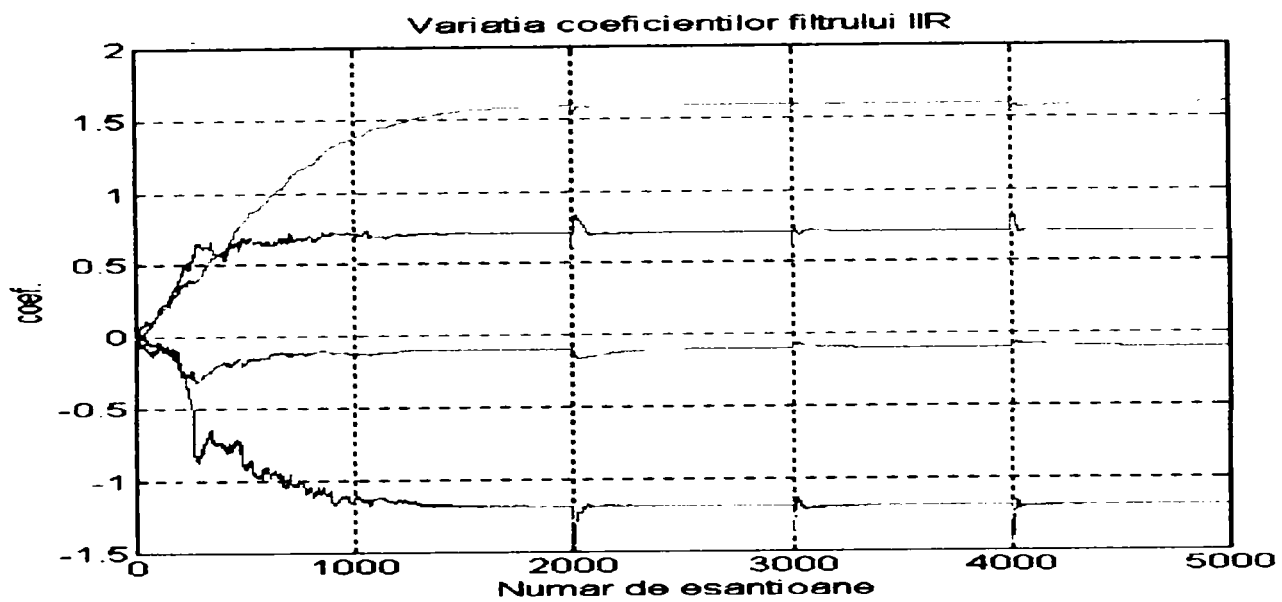


Fig. 4.18. Variația coeficienților filtrului IIR, algoritmul SHARF, forma lattice.

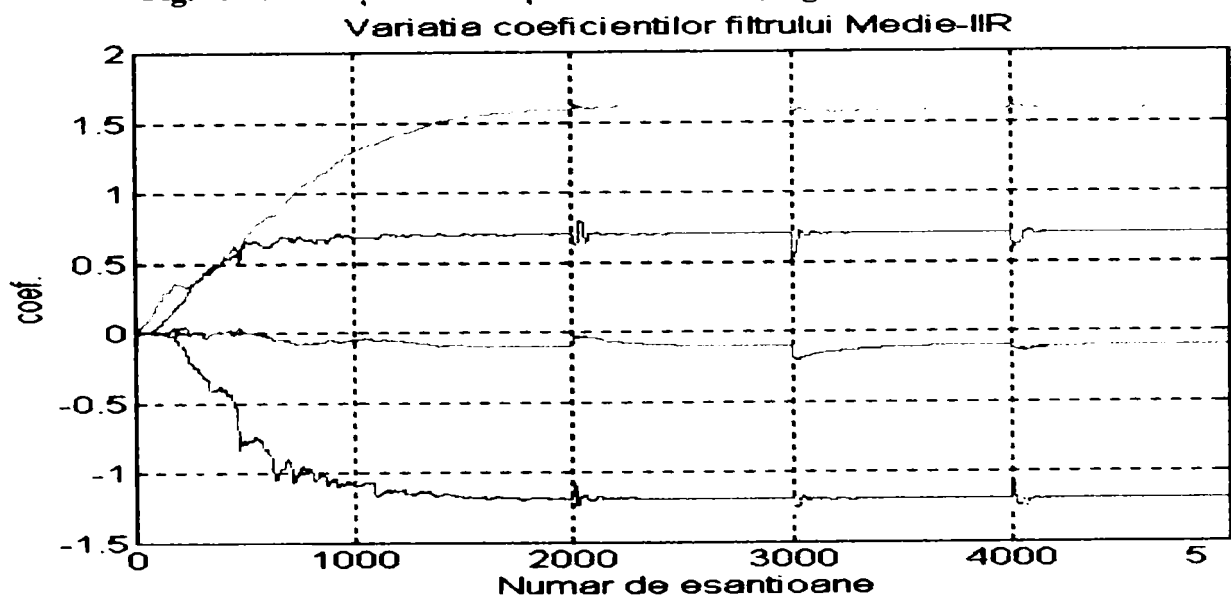


Fig. 4.19. Variația coeficienților filtrului Medie-IIR, algoritmul SHARF, forma lattice.

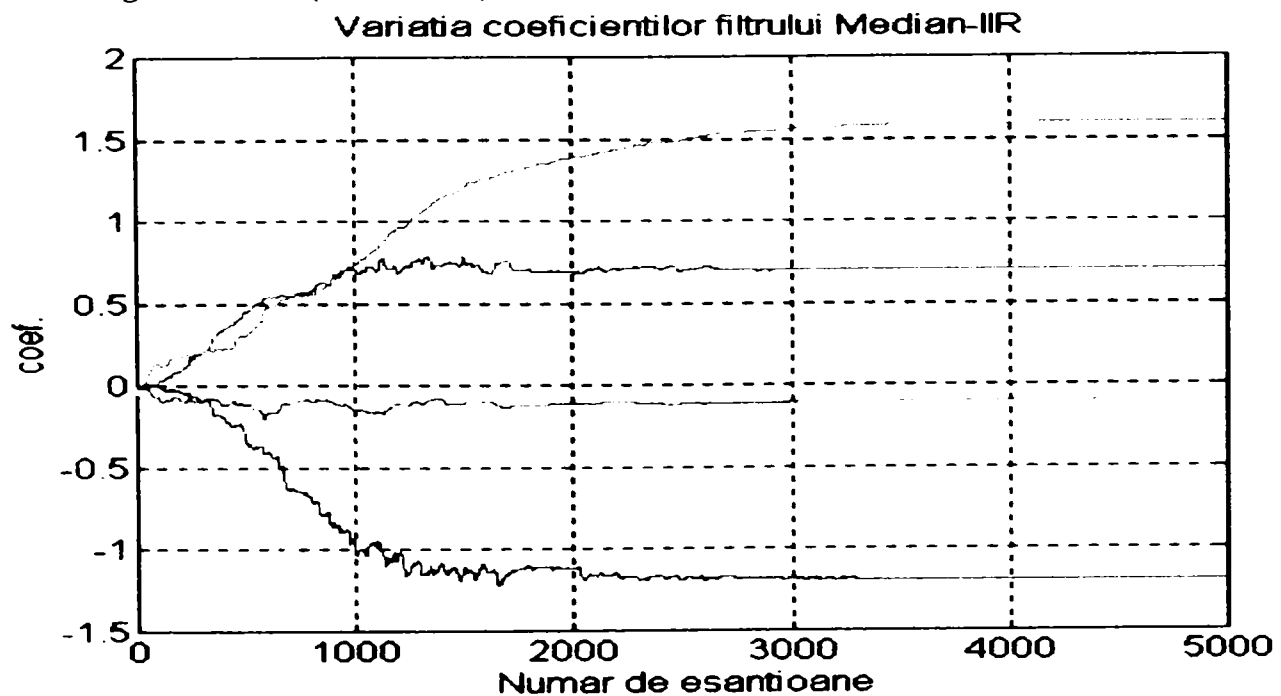


Fig. 4.20. Variația coeficienților filtrului Median-IIR, algoritmul SHARF, forma lattice.

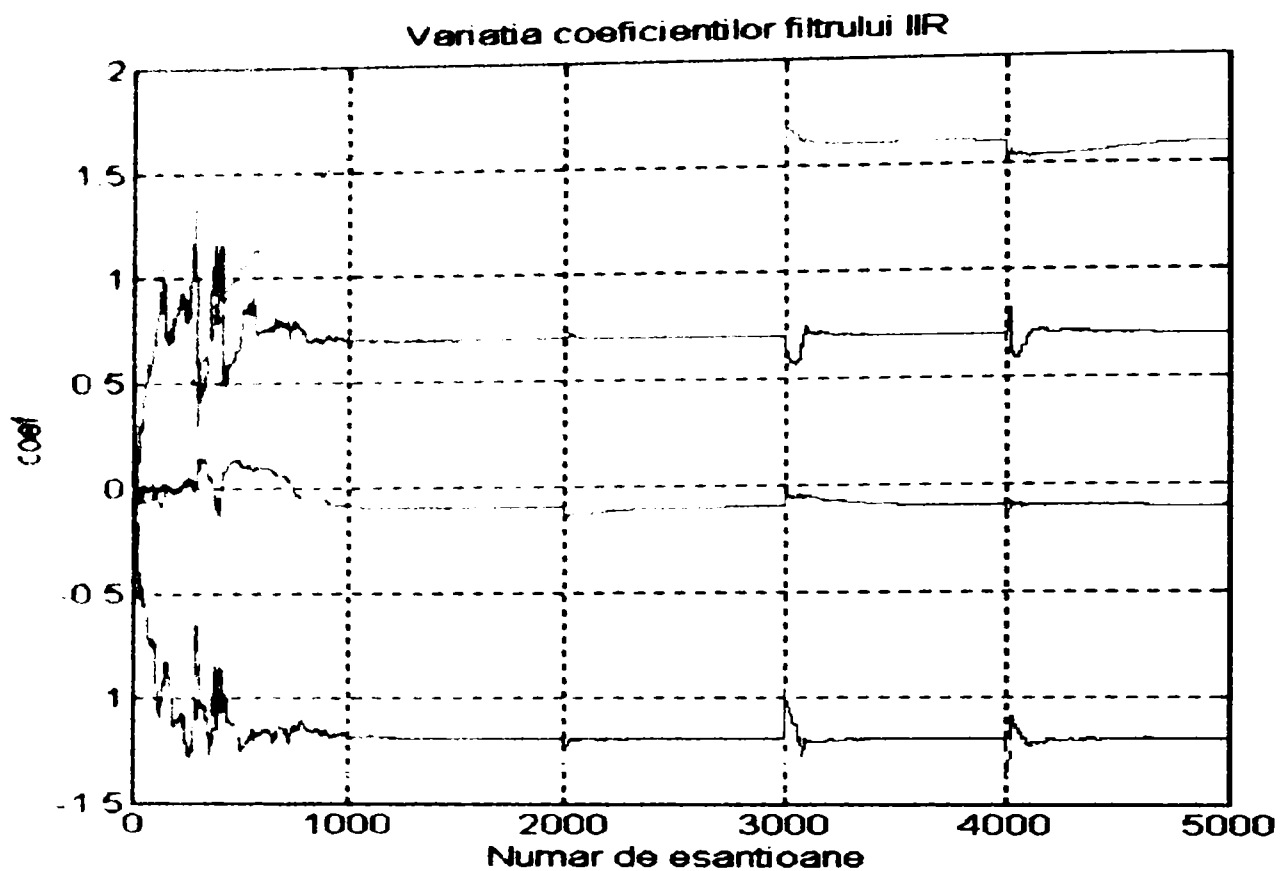


Fig. 4.21. Variatia coeficientilor filtrului IIR, algoritmul Steiglitz-McBride, forma directă.

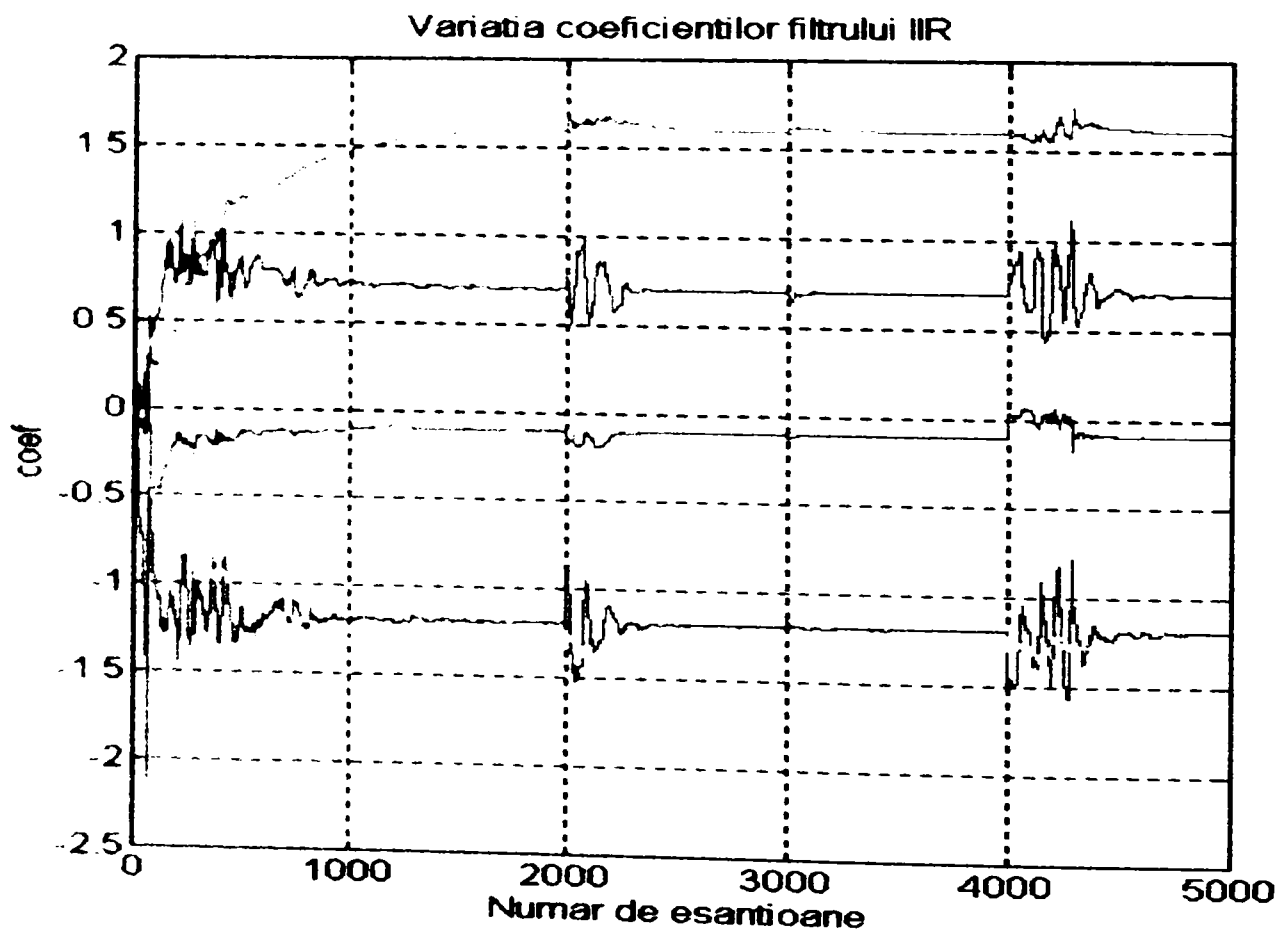


Fig. 4.22. Variatia coeficientilor filtrului IIR, algoritmul Steiglitz-McBride, forma lattice.

Se notează cu \mathbf{x}_{os} vectorul

$$\mathbf{x}_{os} = (x_{(1)} \ x_{(2)} \ \dots \ x_{(N)})^T,$$

cu elementele aranjate în ordinea crescătoare a valorilor astfel încât

$$x_{(1)} \leq x_{(2)} \leq \dots \leq x_{(N)} .$$

Vectorul \mathbf{x}_{os} este denumit ordonarea statistică a vectorului \mathbf{x} . Orice vector de forma

$$\mathbf{x}_j = (x_{j1} \ x_{j2} \ \dots \ x_{jN})^T,$$

unde $(j_1 \ j_2 \ \dots \ j_N)$ este o permutare a lui $(1 \ 2 \ \dots \ N)$ are același vector ordonat statistic \mathbf{x}_{os} .

Vectorul ordonat statistic \mathbf{x}_{os} pierde ordinea temporară în care x_j a fost achiziționat.

Considerând eșantionul $x_j = x_{(i)}$, rangul r_j al eșantionului x_j este definit de relația

$$r_j = i$$

și în mod similar ordinea temporală s_i a lui $x_{(i)}$ este definită de relația

$$s_i = j .$$

S-a demonstrat în [29] că pentru orice vector \mathbf{x} căruia îi corespunde un vector statistic \mathbf{x}_{os} există un unic vector de ordonare după rang

$$\mathbf{r} = (r_1 \ r_2 \ \dots \ r_N)^T$$

și un unic vector de ordonare în timp

$$\mathbf{s} = (s_1 \ s_2 \ \dots \ s_N)^T .$$

Rezultă că un vector \mathbf{x} este complet caracterizat de vectorul \mathbf{x}_{os} și \mathbf{r} sau de \mathbf{x}_{os} și \mathbf{s} . Deoarece $r_{s_i} = i$ și $s_{r_j} = j$, rezultă că dacă se cunoaște vectorul \mathbf{r} se poate determina vectorul \mathbf{s} și invers.

Considerând că vectorul \mathbf{x} reprezintă intrarea filtrului, iar y reprezintă valoarea corespunzătoare a ieșirii, filtrele L1 sunt definite în [35] ca acele filtre pentru care ieșirea este dată de relația

$$y = \sum_{i=1}^N b_i \cdot x_{(i)} , \quad (4.24)$$

unde $\mathbf{b} = (b_1 \ b_2 \ \dots \ b_N)^T$ reprezintă vectorul de coeficienți ai filtrului. În acest caz ieșirea este bazată doar pe utilizarea vectorului \mathbf{x}_{os} și nu depinde de vectorii \mathbf{s} sau \mathbf{r} .

Relația (4.24) mai poate fi scrisă și sub forma

$$y = \sum_{j=1}^N b_{r_j} \cdot x_j . \quad (4.25)$$

În acest mod filtrele L1 țin seama de valoarea eșantionului x_j în generarea semnalului de ieșire prin înmulțirea cu un coeficient care depinde de rangul eșantionului (deci de matricea \mathbf{r}), rezultând că informația despre ordonarea în timp a eșantioanelor devine irelevantă. Pe de altă parte, filtrele liniare FIR generează semnalul de la ieșire potrivit relației

$$y = \sum_{j=1}^N a_j \cdot x_j , \quad (4.26)$$

bazată pe observațiile temporale și unde vectorul $\mathbf{a} = (a_1 \ a_2 \ \dots \ a_N)^T$ reprezintă coeficienții filtrului FIR. Ieșirea filtrului FIR poate fi scrisă sub forma

$$y = \sum_{i=1}^N a_{s_i} \cdot x_{(i)}, \quad (4.27)$$

cu coeficienții a_{s_i} depinzând de poziția în timp a datelor.

Combinând relațiile (4.25) și (4.26) sau (4.24) și (4.27) se pot defini relațiile care determină ieșirea filtrelor combinaționale

$$y = \sum_{j=1}^N c(r_j, j) \cdot x_j, \quad (4.28)$$

sau

$$y = \sum_{i=1}^N c(i, s_i) \cdot x_{(i)}, \quad (4.29)$$

unde coeficienții $c(i, j)$ depind atât de rangul cât și de ordinea în timp a eșantioanelor. Din acest motiv rezultă că filtrele C au ieșirea dată fie de datele ordonate în timp și ponderate în funcție de rang, fie de datele ordonate după rang și ponderate în funcție de ordinea în timp.

Din cele prezentate rezultă că un filtru combinațional necesită un set de N^2 coeficienți $c(i, j)$, dar pentru a produce o valoare a ieșirii la un moment dat utilizează doar un set de N coeficienți $c(r_j, j)$ sau $c(i, s_i)$.

Ieșirea filtrului se poate scrie matriceal sub forma

$$y = \mathbf{w}_N^T \cdot \mathbf{x} = \mathbf{1}^T (\mathbf{C} \circ \mathbf{P}) \cdot \mathbf{x}, \quad (4.30)$$

unde

- \mathbf{w}_N este un vector de coeficienți ponderare de dimensiune N ;
- $\mathbf{1}$ este un vector de dimensiune N cu elementele egale cu 1;
- \mathbf{P} este o matrice care se determină în funcție de datele achiziționate;
- \mathbf{C} este matricea coeficienților filtrului combinațional;
- \circ reprezintă produsul Schur al matricelor \mathbf{C} și \mathbf{P} .

Elementele matricei \mathbf{P} sunt definite de relația

$$p_{ij} = \begin{cases} 1 & \text{dacă } r_j = i \\ 0 & \text{în rest} \end{cases},$$

care arată că matricea \mathbf{P} conține atât informații despre vectorul \mathbf{r} cât și despre vectorul \mathbf{s} , linia i caracterizând observațiile cu rangul i , iar coloana j caracterizând observațiile cu ordinul în timp j . Vectorul pondere $\mathbf{w}_N = (w_1 \ w_2 \ \dots \ w_N)^T$ este o funcție de vectorul de eșantioane \mathbf{x} întrucât pentru diferite realizări ale matricei \mathbf{r} (sau \mathbf{s}) se obțin diferiți coeficienți ai matricei \mathbf{C} în determinarea ieșirii.

Pentru reducerea influenței zgomotelor se înlătură influența eșantioanelor de ranguri inferioare respectiv superioare prin introducerea parametrilor de trunchiere T_1 și T_2 , matricea \mathbf{C} se construindu-se pe baza relațiilor

$$c_{ij} = 0 \text{ pentru } i < T_1 + 1 \text{ și } i > N - T_2, \quad (4.31)$$

în care parametrii T_1 și T_2 trebuie să îndeplinească condițiile

$$0 \leq T_1; T_2 < N \text{ și } T_1 + T_2 < N.$$

Filtrele definite de relațiile (4.31) poartă denumirea de *filtre C trunchiate*, iar parametrii T_1 și T_2 poartă denumirea de parametrii de trunchiere.

Se notează cu Ω_C mulțimea tuturor vectorilor coeficienților de predicție care se pot obține pentru o matrice C dată. Numărul de elemente al mulțimii Ω_C va fi

$$\text{card}(\Omega_C) = \frac{N!}{(T_1 + T_2)!}, \text{ cel mult.}$$

În cazuri particulare filtrele C se reduc la structuri de filtre cunoscute. Considerând cazul că $T_1 = T_2 = 0$ se observă că dacă $c_{ij} = a_j, \forall i$, filtrul C se reduce la un filtru FIR, iar dacă $c_{ij} = b_i, \forall j$ filtrul C se reduce la un filtru LI.

Dacă $c_{ij} = \frac{1}{N}$ pentru $i \geq 1, j \leq N$ se obține filtrul medie (ALMS), iar dacă $c_{ij} = 1$ pentru $i = \frac{N+1}{2}$ și 0 în rest se obține filtrul median (MLMS).

Numărul total de coeficienți necesari a fi definiți pentru un filtru C este N^2 , spre deosebire de doar N coeficienți necesari pentru definirea unui filtru LI sau FIR. Numărul coeficienților poate fi redus la $N(N - T_1 - T_2)$ în cazul utilizării filtrelor C trunchiate și în unele cazuri la un sfert dacă se impune ca atât în funcție de rang cât și în timp coeficienții să fie simetrici adică

$$c_{ij} = c_{N-i+1, j} = c_{i, N-j+1} = c_{N-i+1, N-j+1} \text{ pentru } i \leq \frac{N+1}{2} \text{ și } j \geq 1.$$

4.3.2. Filtre C cu parametrii de trunchiere variabili (VTC)

În aplicații ca restaurarea semnalelor netaționare performanțele filtrelor C se pot îmbunătăți în continuare dacă parametrii T_1 și T_2 se modifică. Dacă apare o variație bruscă (de exemplu un contur) a semnalului în fereastră este necesară o valoare mai mare a parametrilor T_1 și T_2 . Relația (4.19) care permite exprimarea ieșirii în funcție de intrare poate fi rescrisă sub forma

$$y = \mathbf{t}^T (\mathbf{C} \circ \mathbf{P}) \cdot \mathbf{x}, \quad (4.32)$$

unde vectorul $\mathbf{t} = (t_1 \ t_2 \ \dots \ t_N)^T$ conține informația de trunchiere și din acest motiv este denumit vector de trunchiere.

Se cunosc mai multe strategii în alegerea parametrilor T_1 și T_2 dintre care cea mai utilizată este dată de relația

$$\begin{aligned} T_1 &= \text{card}\{j: x_j < \text{med}(x) - q; 1 \leq j \leq N\} \\ T_2 &= \text{card}\{j: x_j > \text{med}(x) + q; 1 \leq j \leq N\} \end{aligned}, \quad (4.33)$$

unde $\text{med}(x)$ este valoarea din mijloc (mediana) a setului de date $\{x_1 \ x_2 \ \dots \ x_N\}$, iar q este valoarea unui parametru real care depinde de gama dinamică estimată a semnalului raportată la lungimea ferestrei de date, de energia zgomotului și de amplitudinea estimată a conturilor.

Ieșirea filtrului VTC este calculată printr-o combinație liniară a $N-T_1-T_2$ eșantioane x_j ale căror valori se găsesc în jurul valorii mediane într-un interval de lungime $2q$. Când fereastra de date nu conține impulsuri sau contururi aproape toate eșantioanele se vor găsi în această fereastră rezultând o valoare mică pentru parametrii T_1 și T_2 . S-a ales mediana ca centrul acestui interval pentru că aceasta urmărește în general cel mai bine contururile.

4.3.3. Filtre C generalizate (GC)

Filtrele C prezentate se caracterizează prin faptul că pentru un vector de trunchiere t dat, fiecare vector w_x corespunde unei realizări particulare P (sau r). Această descriere duce la o structură complexă a filtrului necesitând $\frac{N!}{(T_1+T_2)!}$ vectori independenți pentru un vector t fix.

Filtrele C generalizate (GC) se definesc explicit prin alegerea vectorilor w_x pe baza posibilităților de apariție a zerourilor în structură. Pentru un vector t fixat pot apărea doar $C_N^{T_1+T_2}$ posibilități distincte de apariție a valorilor zero, fiecare furnizând un vector w_x . Această definiție a filtrelor generalizate a permis o reducere de aproximativ $(N-T_1-T_2)!$ ori a numărului maxim de vectori w_x .

Dacă parametrii T_1 și T_2 sunt variabili, numărul total de vectori este dat de relația

$$M = \sum_{j=0}^N C_N^j = 2^N - 1 \quad (4.34)$$

Structura unui filtru GC, în care se consideră parametrii T_1 și T_2 fiși este prezentată în figura 4.23.

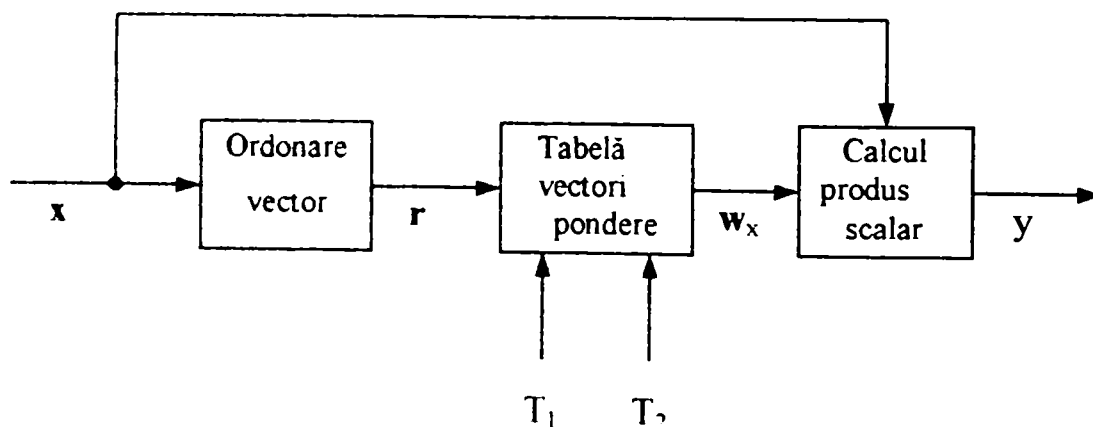


Fig. 4.23. Structura bloc a unui filtru GC cu parametrii de trunchiere fiși.

4.3.4. Implementarea filtrelor C adaptive

În acest paragraf este prezentată o posibilitate de a elimina secvența de zgomot precum și secvența de interferență utilizând un filtru C adaptiv. În figura 4.24 este prezentată structura filtrului C adaptiv.

Modelul general al zgomotului analizat este dat de relația

$$V_k = U_k + N_k + I_k,$$

unde U_k este ieșirea unui filtru liniar și invariant în timp cu răspunsul la impulsul unitar h_k la intrarea căruia se aplică semnalul D_k presupus aleator și nestaționar. N_k reprezintă secvența de

zgomot iar I_k secvența de interferență. Scopul propus este de a estima secvența D_k din setul finit

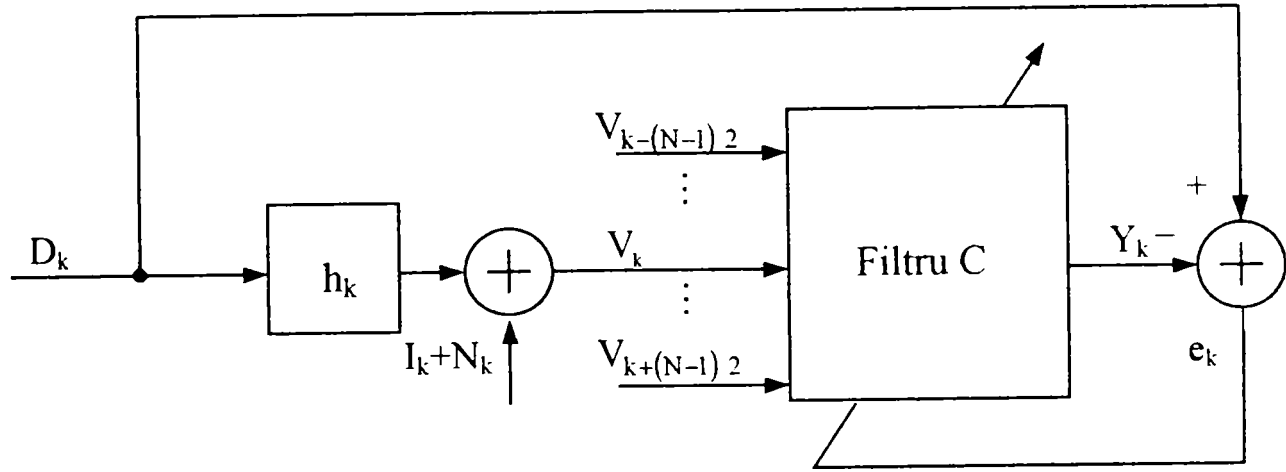


Fig. 4.24. Structura bloc a filtrului C adaptiv.

$$\{V_{k-(N-1)/2}, \dots, V_k, \dots, V_{k+(N-1)/2}\}$$

de N observații în jurul lui V_k . Aceste N eșantioane din fereastra de date procesată la momentul de timp k se notează

$$\mathbf{x}_k = (x_1^k \ x_2^k \ \dots \ x_N^k)^T, \text{ unde}$$

$$V_{k+j-1-(N-1)/2} = x_j^k.$$

Definind variabila aleatoare z_{ij}^k

$$z_{ij}^k = \begin{cases} x_j^k & \text{dacă } r_j = i \text{ (sau } s_i = j) \\ 0 & \text{in rest} \end{cases},$$

există două posibilități de a implementa un filtru C adaptiv:

1. Definind eroarea instantanee e_k obținută ca diferență între Y_k și D_k reactualizarea coeficienților filtrului se poate face potrivit relației

$$c_{k+1}(i,j) = c_k(i,j) + 2 \cdot \mu \cdot e_k \cdot z_{ij}^k, 1 \leq i, j \leq N, \quad (4.35)$$

unde $e_k = D_k - Y_k$, iar μ este o constantă scalară. Această formă este similară implementării algoritmului LMS, structura filtrului C adaptiv fiind prezentată în figura 4.24.

2. Dacă se dorește a se implementa aceeași structură dar utilizând un filtru GC cu factor de trunchiere nenul, relațiile care exprimă setul de coeficienți sunt

$$c_{k+1}(i,j) = \begin{cases} 0 & \text{dacă } 1 \leq i \leq T_1 \text{ și } 1 \leq j \leq (N+1)/2 \\ c_k(i,j) + 2\mu e_k \cdot \bar{z}_{ij}^k & \text{dacă } T_1 + 1 \leq i \leq (N+1)/2 \text{ și } 1 \leq j \leq (N+1)/2 \end{cases} \quad (4.36)$$

unde

$$c_{k+1}(N-i+1, j) = c_{k+1}(i, N-j+1) = c_{k+1}(N-i+1, N-j+1) = c_{k+1}(i, j)$$

$$\text{și } \bar{z}_{ij}^k = \frac{1}{\eta} (z_{ij}^k + z_{N-i+1, j}^k + z_{i, N-j+1}^k + z_{N-i+1, N-j+1}^k)$$

Parametrul η poate avea valorile 1 dacă cel mult unul dintre coeficienții z este diferit de 0 și 2 dacă doi coeficienți sunt diferiți de 0.

4.4. Filtre bazate pe criteriul erorii medii absolute

4.4.1. Filtre WOS

O categorie de filtre neliniare utilizate în filtrarea adaptivă a semnalelor influențate de impulsuri cu amplitudine mare și durată scurtă sau care conțin contururi o reprezintă filtrele WOS (Weighted Order Statistic), a căror structură se bazează pe criteriul erorii medii absolute (MAE)

În [33] a fost prezentată structura acestor tipuri de filtre care se definește pornind de la secvența de N eșantioane $x(n)$, unde $N=N_1+N_2+1$. Vectorul \mathbf{x} ale cărui elemente sunt eșantioanele ordonate în timp este dat de relația

$$\mathbf{x}(n) = [x(n-N_1) \ x(n-N_1+1) \ \dots \ x(n) \ \dots \ x(n+N_2)]^T = [x_1(n) \ x_2(n) \ \dots \ x_N(n)]^T$$

Vectorul de date $\mathbf{x}(n)$ permite obținerea estimatului $\hat{D}(n)$ a valorii dorite $D(n)$ utilizând un filtru WOS a cărei structură este prezentată în figura 4.25.

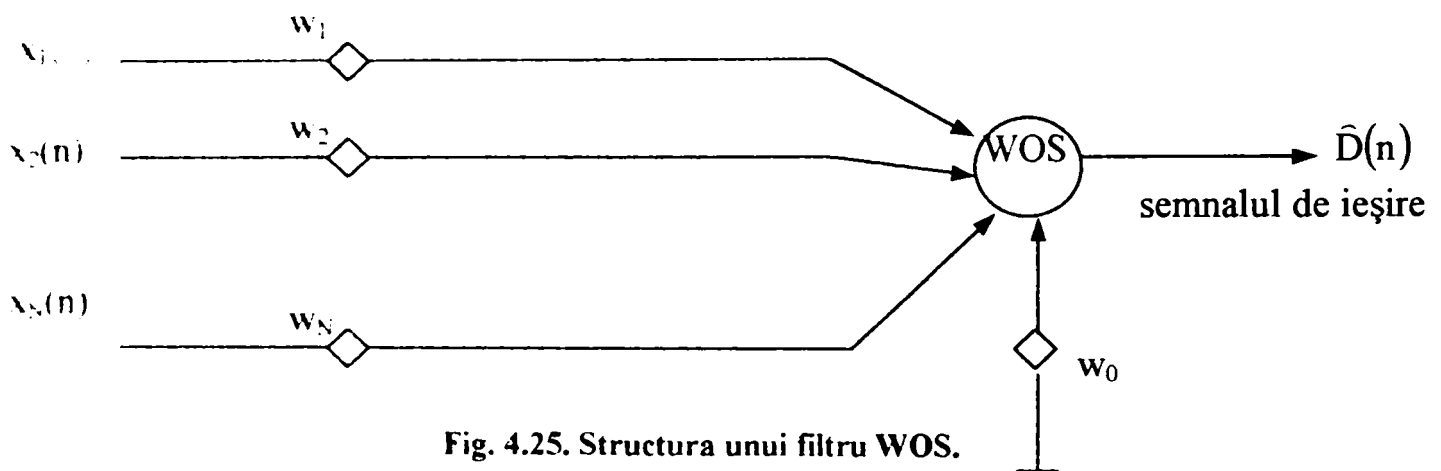


Fig. 4.25. Structura unui filtru WOS.

Vectorul coeficienților de pondere conține valori pozitive (sau nule) fiind dat de relația

$$\mathbf{w} = [w_0 \ w_1 \ \dots \ w_N]^T$$

Ieșirea filtrului WOS se obține considerând cunoscută valoarea coeficientului w_0 și ordonând setul de valori

$$\mathbf{w} \diamond \mathbf{x} = [w_1 \diamond x_1(n) \ w_2 \diamond x_2(n) \ \dots \ w_N \diamond x_N(n)], \quad (4.37)$$

în mod descrescător în funcție de valorile eșantioanelor $x_1(n), \dots, x_N(n)$, unde operația \diamond specifică faptul că w_i și x_i formează un cuplu. În urma operației de sortare se obțin cupluri formate din datele sortate și ponderile corespunzătoare.

În continuare se adună ponderile începând cu cea a celui mai mare eșantion până când suma devine mai mare decât w_0 , în acest caz eșantionul corespunzător ultimei ponderi adunate considerându-se a fi ieșirea filtrului. Se observă că gradul de complexitate al filtrului WOS este proporțional cu lungimea N a filtrului și nu cu suma ponderilor.

4.4.2. Filtre FIR - Median -Hibride (FMH)

Filtrele hibride FMH reprezintă o combinație între k filtre FIR și un filtru median. Structura unui filtru FMH este prezentată în figura 4.26.

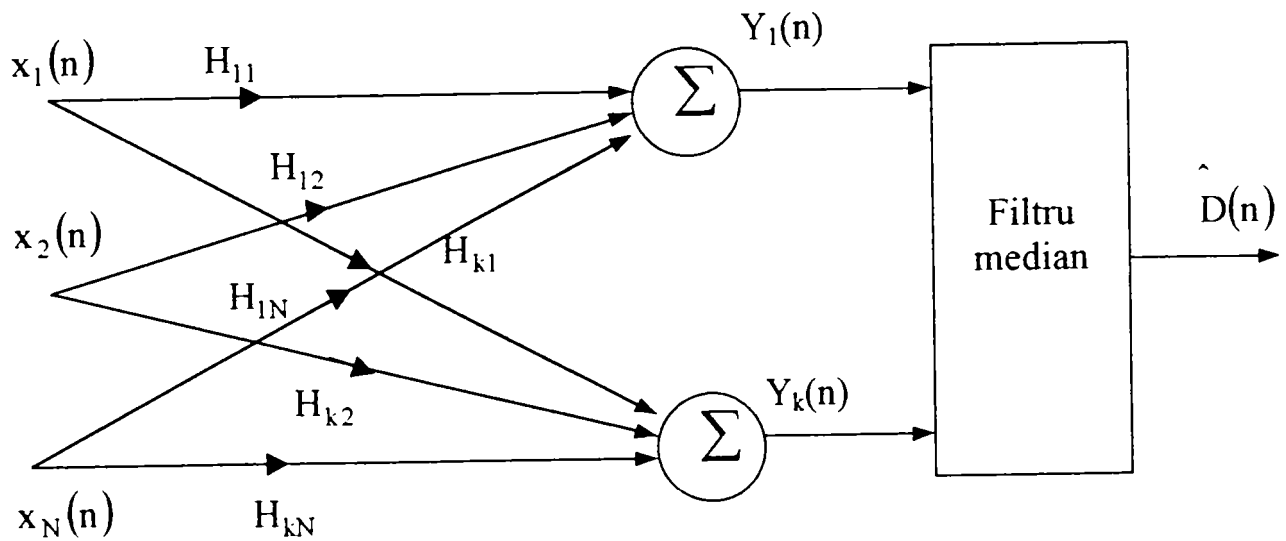


Fig. 4.26. Structura generală a unui filtru FMH.

Prima etapă constă în filtrarea semnalului de intrare $x(n)$ cu ajutorul a k filtre de tip FIR de lungime N . Ieșirile filtrelor FIR sunt date de relația generală

$$y_i(n) = \mathbf{H}_i^T \cdot \mathbf{x}(n), \quad i = 1, 2, \dots, k,$$

unde \mathbf{H}_i reprezintă vectorul coeficienților pentru filtrul FIR cu numărul i .

$$\mathbf{H}_i = [\mathbf{H}_{i1} \ \mathbf{H}_{i2} \ \dots \ \mathbf{H}_{iN}]^T$$

Cele k ieșiri ale filtrelor FIR sunt filtrate cu ajutorul unui filtru median, ieșirea acestuia reprezentând ieșirea filtrului FMH.

4.4.3. Filtre FIR-WOS-Hibride (FWH)

Filtrele FIR-WOS hibride (FWH) se obțin prin înlocuirea filtrului median din structura unui filtru FMH cu un filtru WOS.

Structura generală a unui filtru FWH este prezentată în figura 4.27.

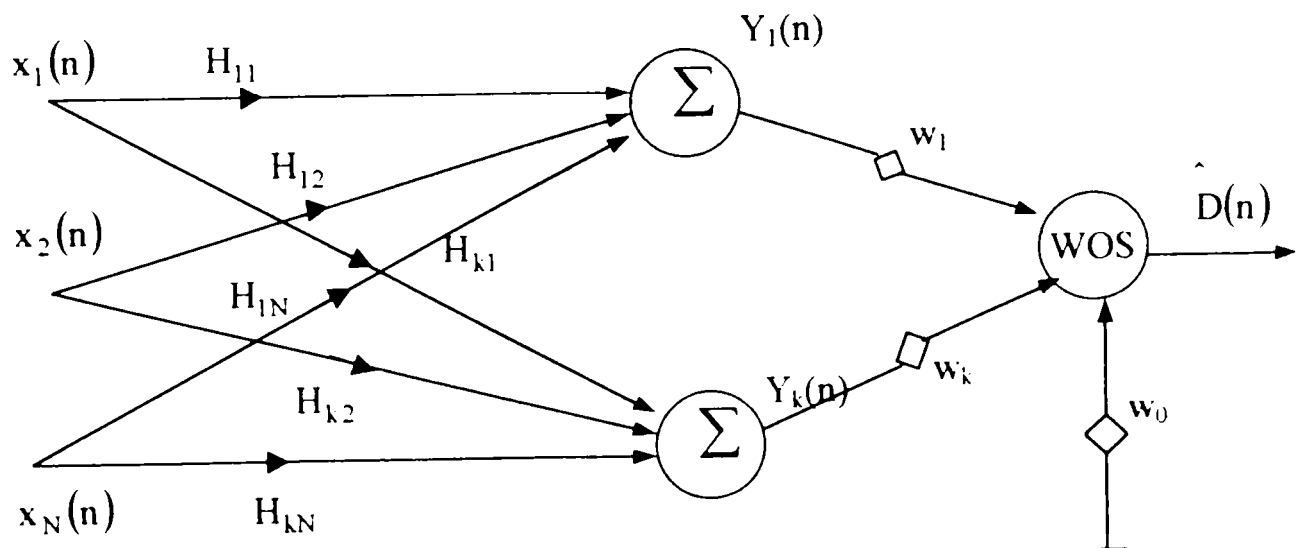


Fig. 4.27. Structura generală a unui filtru FWH.

Ca și în cazul filtrului FMH, prima etapă constă în filtrarea semnalului $x(n)$ prin intermediul a k filtre FIR, iar în etapa a doua cele k ieșiri ale filtrelor FIR sunt considerate semnale de intrare ale unui filtru WOS cu coeficienți nenegativi de ponderare

$$\mathbf{w} = [w_0 \ w_1 \ \dots \ w_k]^T$$

S-a arătat în [33] că filtrele FWH reprezintă o clasă de filtre mai largă, incluzând clasele filtrelor FIR, WOS și FMH.

4.4.4. Arhitectura și implementarea filtrelor WOS

Considerând că semnalele prezente la intrarea unui filtru WOS sunt reprezentate de numere întregi rezultă că

$$x(n) \in \{0, 1, \dots, M-1\}, \forall n$$

Orice vector de date de intrare va putea fi descompus în $M-1$ secvențe binare notate $\{x^m(n)\}$ $m = 1, \dots, M-1$ printr-o operație de comparație cu un prag, m reprezentând pragul de comparație. Operația de comparație cu pragul m se notează cu T^m , astfel încât se poate scrie

$$x^m(n) = T^m(x(n)) = U(x(n) - m) = \begin{cases} 1 & \text{dacă } x(n) \geq m \\ 0 & \text{în rest} \end{cases}, \quad (4.38)$$

unde $U(\cdot)$ reprezintă funcția treaptă unitară.

Vectorul de date $\mathbf{x}(n)$ poate fi obținut din vectorii cu componente binare potrivit relației

$$\{\mathbf{x}(n)\} = \sum_{m=0}^{M-1} \{T^m(x(n))\} = \sum_{m=0}^{M-1} \{x^m(n)\} \quad (4.39)$$

Utilizând arhitectura bazată pe această descompunere filtrele WOS pot fi implementate utilizând porți logice cu prag a căror intrări sunt semnale binare, ieșirea filtrului WOS fiind egală cu suma ieșirilor filtrelor, adică

$$\hat{D}(n) = \sum_{m=1}^{M-1} U(\mathbf{w}^T \cdot \tilde{\mathbf{x}}^m(n)), \quad (4.40)$$

unde

$$\tilde{\mathbf{x}}^m(n) = [-1, x_1^m(n), \dots, x_N^m(n)]^T,$$

$x^m(n)$ putând avea valorile 0 sau 1.

În cazul filtrelor FWH ieșirea poate fi de asemenea exprimată ca

$$\hat{D}(n) = \sum_{m=1}^{M-1} U(\mathbf{w}^T \cdot \tilde{\mathbf{y}}^m(n)), \quad (4.41)$$

unde \mathbf{w} reprezintă vectorul coeficienților de ponderare iar $\tilde{\mathbf{y}}^m(n)$ este dat de relația

$$\tilde{\mathbf{y}}^m(n) = [-1, y_1^m(n), \dots, y_k^m(n)]^T,$$

unde

$$y_i^m(n) = T^m(y_i(n)).$$

4.4.5. Implementarea filtrelor adaptive rapide WOS și FMH

Ca și în cazul filtrării liniare, scopul unui filtru neliniar optimal este de a minimiza un criteriu de performanță, în cazul filtrelor WOS și FMH acest criteriu fiind eroarea medie absolută (MAE).

Dacă semnalele $D(n)$ și $x(n)$ sunt ambele staționare, trebuie minimizată eroarea

$$J(\mathbf{w}) = E\left[|D(n) - \hat{D}(n)|\right] \quad (4.42)$$

Se definește operatorul de proiecție $P[\cdot]$ cu ajutorul relației

$$P(x) = \begin{cases} x, & \text{dacă } x \geq 0 \\ 0, & \text{în rest} \end{cases} \quad (4.43)$$

Relațiile de recurență simplificate care determină setul de coeficienți de ponderare în cazul implementării filtrelor WOS sunt

$$\begin{aligned} w_0(n+1) &= P[w_0(n) - 2\mu(D(n) - \hat{D}(n))] \\ w_j(n+1) &= P[w_j(n) + 2\mu(D(n) - \hat{D}(n)) \cdot U(x_j(n) - \hat{D}(n))] \end{aligned} \quad (4.44)$$

unde $j=1,2,\dots,N$ iar μ este un parametru de adaptare a cărui valoare influențează rapiditatea convergenței algoritmului. Relațiile (4.44) reprezintă forma cea mai simplă în care se poate implementa un filtru adaptiv WOS dar prezintă dezavantajul că necesită un vector inițial al coeficienților de pondere bine ales. Funcționarea acestui filtru adaptiv se bazează pe faptul că dacă ieșirea filtrului este mai mică decât ieșirea dorită, ponderile corespunzătoare vor crește astfel încât ieșirea să crească.

Alegerea inițială a vectorului coeficienților de ponderare se poate face prin două metode. Prima metodă se bazează pe utilizarea relațiilor exacte, prezentate în [33], dar care necesită un volum de calcul mai mare însă este insensibilă la alegerea vectorului inițial iar după încheierea perioadei de învățare se comută pe utilizarea relațiilor (4.44). A doua metodă se bazează pe "ghicirea" valorilor inițiale ale coeficienților de ponderare.

Structura unui filtru WOS adaptiv este prezentată în figura 4.28.

În mod asemănător cu relațiile utilizate la realizarea filtrelor adaptive WOS, relațiile de recurență pentru filtrele adaptive FWH sunt

$$\begin{aligned} w_0(n+1) &= P[w_0(n) - 2\mu_2 \cdot (D(n) - \hat{D}(n))] \\ w_i(n+1) &= P[w_i(n) + 2\mu_2(D(n) - \hat{D}(n)) \cdot U(y_i(n) - \hat{D}(n))] \quad \text{unde } i = 1, 2, \dots, k \\ H_i(n+1) &= H_i(n) + \mu_1(D(n) - \hat{D}(n)) \cdot w_i(n) \cdot U_s(y_i(n) - \hat{D}(n)) \cdot (1 - U_s(y_i(n) - \hat{D}(n))) \cdot x(n), \end{aligned} \quad (4.45)$$

unde $i = 1, 2, \dots, k$

În relațiile (4.45) s-a utilizat funcția sigmoidală definită de relația

$$U_s(x) = \frac{1}{1 + e^{-x}}$$

În cazul filtrelor adaptive FWH algoritmul este convergent dacă parametrii de adaptare μ_1 și μ_2 sunt suficient de mici. Ca și în cazul filtrelor adaptive WOS există două posibilități de alegere a setului inițial de coeficienți.

Prima posibilitate constă în alegerea unui filtru FWH cu o astfel de structură care să prezinte mai multe minime locale. A doua posibilitate este aceea de a face mai multe porniri de antrenament cu diferite seturi de valori inițiale ale coeficienților, alegându-se ulterior setul optim

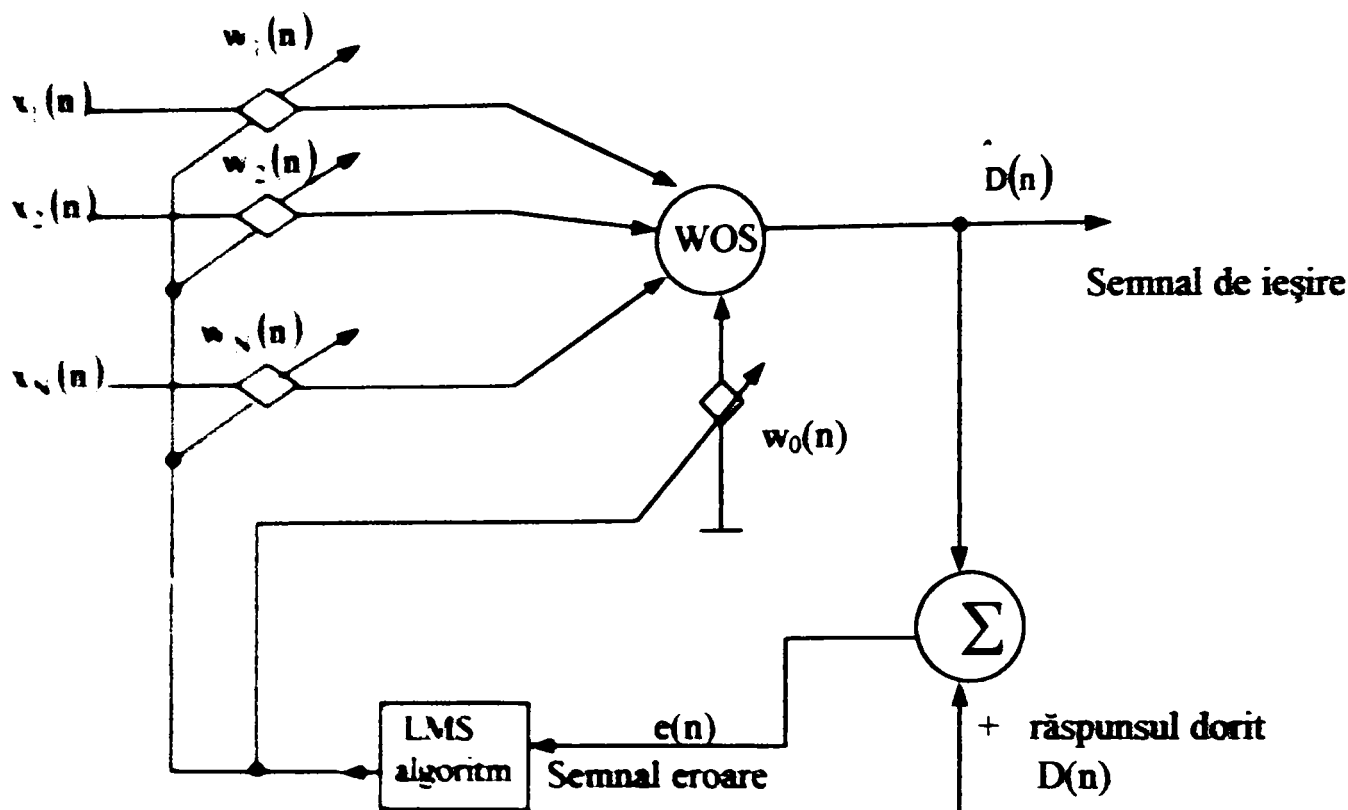


Fig. 4.28. Structura unui filtru WOS adaptiv.

4.5. Concluzii

În acest capitol autorul a realizat un studiu privitor la posibilitățile de utilizare a filtrelor neliniare în scopul eliminării perturbațiilor cu caracter negaussian precum și în scopul prelucrării semnalelor cu nivelul mediei nestaționar. Aceste filtre se bazează pe criteriul minimizării erorii pătratice medii sau pe criteriul minimizării erorii absolute minime. Pentru fiecare tip de filtru neliniar a fost prezentat modul de implementare a unui filtru adaptiv precum și relațiile de determinare a coeficienților filtrului. De asemenea, autorul a propus noi tipuri de filtre adaptive neliniare studiind proprietățile acestora și subliniind principalele lor avantaje.

În primul paragraf autorul a studiat proprietățile filtrelor MLMS și ALMS, ajungând la următoarele concluzii.

1. Odată cu creșterea valorii parametrului de adaptare α crește și viteza de convergență a ambelor tipuri de filtre, însă se observă că pentru aceeași valoare a lui α filtrul ALMS este mai rapid convergent decât filtrul MLMS, fapt rezultat din figurile 4.2 și 4.3.

2. Variația coeficienților de predicție în situația apariției unui impuls este mai mică în cazul filtrului MLMS decât în cazul filtrului ALMS, rezultând că filtrarea MLMS este recomandabilă în situația în care se dorește eliminarea impulsurilor cu amplitudine mare și

durată scurtă.

3. În cazul filtrării MLMS, odată cu creșterea lungimii ferestrei amplitudinea variației coeficienților de predicție este mai mică, în timp ce în cazul filtrării ALMS lungimea ferestrei practic nu influențează amplitudinea variației coeficienților, fapt rezultat din figurile 4.4 și 4.5..

4. Datorită volumului suplimentar de calcule necesar implementării filtrului MLMS timpul de rulare pentru acest tip de filtru este mai mare decât în cazul filtrării ALMS pentru aceeași lungime a ferestrei, fapt rezultat din Tabelul 4.1.

5. Implementarea filtrului adaptiv bazat pe avantajele acestor două tipuri de filtre s-a făcut utilizând un parametru de detecție a unui impuls în interiorul ferestrei, parametru definit în două moduri (relațiile 4.11, respectiv 4.12). S-a arătat că ambele moduri de definire duc la performanțe ale filtrului adaptiv similare (figurile 4.6. și 4.7.), însă timpul de calcul necesar implementării filtrului adaptiv este mai mare în situația utilizării relației (4.11), fapt rezultat din Tabelul 4.2.

6. Autorul a propus o metodă de reducere a volumului de calcule necesar, în situația utilizării parametrului de detecție a impulsului pe baza relației (4.11). Metoda se bazează pe calculul mediei eșantioanelor din fereastra de date de intrare la momentul n cunoscând media eșantioanelor din fereastra de date de intrare de la momentul $n-1$.

7. Utilizând aceeași metodă, autorul a prezentat posibilitatea reducerii volumului de calcule necesar în situația implementării unui filtru ALMS, în special în situația utilizării ferestrelor de lungime mare.

8. În paragraful 4.1.6 autorul a propus un nou tip de filtru adaptiv denumit MALMS, filtru care combină avantajele filtrelor MLMS și ALMS. Definirea acestui tip de filtru s-a făcut utilizând un coeficient de ponderare a celor două tipuri de filtre în funcție de amplitudinea parametrului de detecție a prezenței unui impuls în fereastra de date, coeficient de ponderare introdus de autor. S-au prezentat, de asemenea avantajele și dezavantajele utilizării acestui tip de filtru adaptiv.

9. În paragraful 4.2. autorul a definit un nou tip de filtru adaptiv IIR care reactualizează coeficienții pe baza unuia din cei 3 algoritmi de identificare prezentați în capitolul precedent, combinat cu o operație de ordonare.

Autorul a definit relațiile de reactualizare ale coeficienților noului tip de filtru și a verificat experimental caracteristicile filtrului adaptiv în procesul de identificare a parametrilor unui sistem necunoscut. În încheierea paragrafului, a prezentat concluziile referitoare la caracteristicile filtrului adaptiv propus, arătând principalele avantaje ale utilizării acestuia.

10. În paragrafele 4.3 și 4.4 autorul a realizat o prezentare a principalelor tipuri de filtre neliniare întâlnite în literatura de specialitate (filtrele combinaționale, filtre WOS, FMH și FWH) care pot fi utilizate ca filtre adaptive, prezentând de asemenea modul de implementare a filtrului adaptiv respectiv.

5. REGLAREA ADAPTIVĂ A TEMPERATURII LA CUPTORUL CU REZISTENȚĂ ELECTRICĂ

În acest capitol autorul prezintă o metodă de identificare în timp real a parametrilor procesului de încălzire a cuptorului cu rezistență electrică. În [46] și [47] s-a considerat că acest proces poate fi bine aproximat considerând că modelul încălzirii cuptorului cu rezistență electrică este cel al unui sistem cu o constantă de timp și timp mort. Verificarea metodei propuse s-a făcut prin compararea rezultatelor cu cele obținute prin metoda de identificare on-line

În primul paragraf, autorul a studiat metoda de identificare on-line prezentând concluzii referitoare la erorile ce intervin în măsurarea parametrilor.

În paragraful următor autorul a prezentat metoda propusă pornind de la alegerea celei mai bune aproximări a modelului sistemului, prezentând apoi modul de obținere a funcției sistem pentru sistemul discret. În continuare, autorul a studiat cei trei algoritmi de identificare prezentați în capitolul 3, implementând fiecare algoritm atât în forma directă, cât și în forma lattice. În urma rezultatelor obținute prin simulare autorul a ales algoritmul utilizat precum și forma de implementare. Etapa următoare a constat în verificarea faptului că aproximația făcută a condus la obținerea unui filtru numeric care să se comporte cât mai apropiat de modelul ales.

În paragraful următor au fost prezentate criteriile de alegere a reguletoarelor adaptive precum și posibilitățile de implementare ale acestora.

Următorul paragraf prezintă metodele de eliminare a zgomotelor cu caracter negaussian apărute în procesul de măsurare a temperaturii, iar în ultimul paragraf este prezentat sistemul de reglare a temperaturii realizat de autor, precum și rezultatele experimentale obținute.

În încheierea capitolului sunt prezentate concluziile autorului privitoare la metoda propusă comparativ cu metoda de identificare on-line.

5.1. Metoda de identificare on-line a parametrilor proceselor lente cu timp mort

În [46] s-a arătat că modelul cuptorului cu rezistență electrică este cel al unui sistem cu o constantă de timp și timp mort, definit de relația

$$H(s) = \frac{K \cdot e^{-\tau s}}{1 + Ts} \quad , \quad \text{unde } T, \tau > 0 \quad . \quad (5.1)$$

În relația (5.1) K reprezintă factorul de amplificare a sistemului, τ reprezintă timpul mort, iar T reprezintă constanta de timp.

Metoda de identificare on-line, prezentată în [46], se bazează pe aplicarea la intrarea unui sistem aflat în starea de echilibru descrisă de punctul (X_0, Y_0) , a unui semnal de intrare dat de relația

$$x(t) = \begin{cases} X_1 & 0 \leq t \leq T_0 \\ X_0 & t > T_0 \end{cases}, \quad (5.2)$$

și a cărei formă este prezentată în figura 5.1.

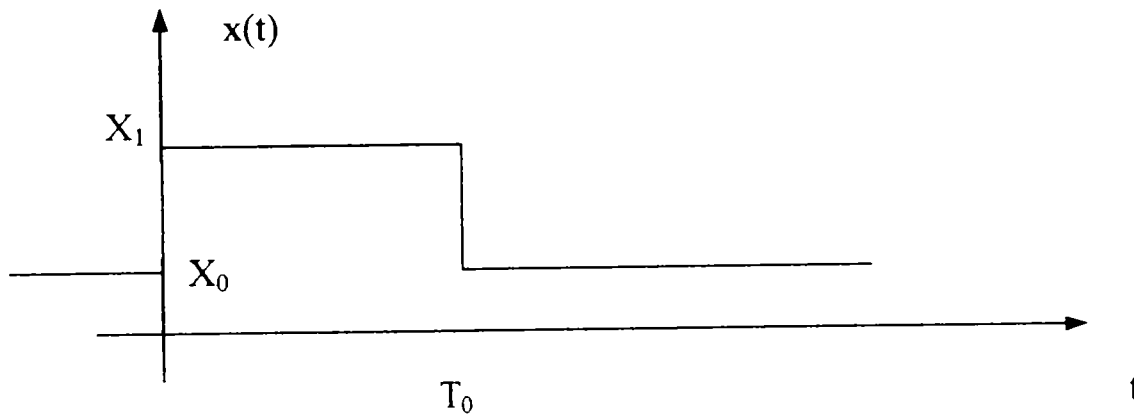


Fig. 5.1. Forma semnalului de intrare.

Aplicarea acestui semnal de intrare în locul semnalului treaptă prezintă două avantaje. Primul avantaj constă în faptul că se poate observa dacă ieșirea y se poate stabiliza în același punct staționar Y_0 sau nu. Dacă nu este atinsă valoarea Y_0 înseamnă fie că procesul este nestaționar, fapt ce poate conduce la o alegere mai bună a modelului procesului, fie că a apărut o perturbație în timpul experimentului, fapt ce implică reluarea experimentului. Al doilea avantaj constă în aceea că diferențele

$$x(t) - X_0 \quad \text{și} \quad y(t) - Y_0,$$

sunt practic nule după un interval de timp mai mare decât timpul de stabilizare al procesului, fapt ce presupune că integralele

$$I_{XK} = \int_0^{\infty} (-t)^K [x(t) - X_0] \cdot dt \quad (5.3)$$

și

$$I_{YK} = \int_0^{\infty} (-t)^K [y(t) - Y_0] \cdot dt, \quad (5.4)$$

unde $K = 0, 1, \dots$, vor fi finite.

Considerând că modelul ales depinde de N parametri, ideea identificării este de a evalua funcția $H(s)$ și primele $N-1$ derivate în origine, rezultând un sistem de N ecuații cu N necunoscute, sistem de ecuații din care se pot determina cei N parametri ai sistemului.

Cele $N-1$ derivate se pot determina recursiv pe baza relației

$$Y(s) = H(s) \cdot X(s),$$

din care rezultă

$$Y(0) = H(0) \cdot X(0)$$

și prin derivări succesive se obține relația generală pentru derivata de ordin k în origine

$$H^{(k)}(0) = \frac{Y^{(k)}(0) - \sum_{i=1}^k C_K^i H^{(k-i)}(0) \cdot X^{(i)}(0)}{X(0)}, \quad (5.5)$$

unde

$$Y^{(k)}(0) = \int_0^x (-t)^k [y(t) - Y_0] \cdot dt, \quad (5.6)$$

$$X^{(k)}(0) = \int_0^x (-t)^k [x(t) - X_0] \cdot dt, \quad K = 0, 1, \dots, N-1. \quad (5.7)$$

Integralele date de relațiile (5.6) și (5.7) pot fi calculate pe un interval finit $0 \leq t \leq T_i$, unde T_i reprezintă durata de integrare. Aplicând relația (5.5) pentru modelul dat de relația (5.1) s-au obținut succesiv

$$\begin{aligned} H(0) &= K \\ H'(0) &= -K \cdot (T + \tau) \\ H''(0) &= K \cdot [(T + \tau)^2 + T^2], \end{aligned} \quad (5.8)$$

relații care au permis evaluarea parametrilor modelului

$$\begin{aligned} K &= H(0) \\ T &= \sqrt{\frac{H'(0)}{H(0)} - \left(\frac{H'(0)}{H(0)}\right)^2} \\ \tau &= -\frac{H'(0)}{H(0)} - T. \end{aligned} \quad (5.9)$$

Autorul a verificat prin simulare această metodă, determinând erorile care pot apărea în măsurarea parametrilor procesului, utilizând programul prezentat în Anexa 5.1. Durata impulsului de test a fost aleasă 50 de secunde, durata de integrare de 100 de secunde, iar perioada de eșantionare de 0.1 secunde. Autorul a studiat influența variației constantei de amplificare, a constantei de timp și a duratei timpului mort asupra rezultatelor obținute în urma măsurării. Rezultatele obținute sunt prezentate în tabelele 5.1, 5.2, respectiv 5.3.

Din Tabelul 5.1. se observă că influența variației factorului de amplificare asupra rezultatelor obținute în urma măsurării parametrilor este mică, valorile obținute pentru constanta de timp și timpul mort fiind aceleași indiferent de valoarea factorului de amplificare. Din Tabelul 5.2, respectiv 5.3, se observă că variațiile constantei de timp și a timpului mort au o mult mai mare influență asupra valorilor măsurate ale parametrilor, în special la valori mai mari ale acestor parametri.

Metoda prezintă astfel dezavantajul că, dacă durata impulsului de test și a duratei de integrare nu sunt alese în corelație cu valorile reale ale parametrilor sistemului, valorile măsurate ale parametrilor prezintă erori mari. Reducerea erorilor de măsurare se poate face crescând durata semnalului de test și a duratei de integrare.

Tabelul 5.1. Influența modificării factorului de amplificare asupra valorilor măsurate ale parametrilor procesului ($T=7$, $\tau=4$).

k	k_m	T_m	τ_m
1	0,9998	6,9324	4,0039
2	1,9996	6,9324	4,0039
3	2,9994	6,9324	4,0039
4	3,9992	6,9324	4,0039
5	4,999	6,9324	4,0039
6	5,9988	6,9324	4,0039
7	6,9986	6,9324	4,0039
8	7,9984	6,9324	4,0039
9	8,9982	6,9324	4,0039
10	9,998	6,9324	4,0039

Tabelul 5.2. Influența modificării constantei de timp asupra valorilor măsurate ale parametrilor procesului ($k=5$, $\tau=4$).

T	k_m	T_m	τ_m
1	5,000	0,9996	3,9512
3	5,000	2,9999	3,9504
5	4,9999	4,9951	3,9544
7	4,999	6,9324	4,0039
9	4,9946	8,7054	4,1682
11	4,9835	10,2409	4,4733
13	4,9632	11,5243	4,8985
15	4,9329	12,5771	5,4058
17	4,8927	13,4338	5,9585
19	4,8438	14,1302	6,5284

Tabelul 5.3. Influența modificării duratei timpului mort asupra valorilor măsurate ale parametrilor procesului ($T=7$, $k=5$).

τ	k_m	T_m	τ_m
1	4,9994	6,9520	0,9887
2	4,9993	6,9462	1,9932
4	4,999	6,9324	4,0039
6	4,9987	6,9151	6,0171
8	4,9983	6,8936	8,0334
10	4,9977	6,8669	10,0534
12	4,9970	6,8337	12,0779
14	4,9959	6,7926	14,1079
16	4,9946	6,7418	16,1445
18	4,9928	6,6791	18,1890

5.2. Metoda adaptivă de identificare a parametrilor procesului

5.2.1. Aproximări ale modelului funcției de transfer a sistemului

În unele aplicații, ca de exemplu procesul de încălzire a cuptorului cu rezistență electrică, apar întârzieri între semnalul de la intrare și cel de la ieșire de forma

$$y(t) = x(t - \tau) \quad , \tau > 0 \quad .$$

unde τ poartă denumirea de timp mort sau timp de propagare.

Funcția de transfer asociată unui proces cu timp mort este

$$H_{\tau}(s) = e^{-s\tau} \quad (5.10)$$

Una din problemele legate de asemenea procese este problema realizabilității funcției de transfer printr-un sistem finit dimensional, fapt care se reduce la aproximarea printr-o funcție rațională a funcției $e^{-s\tau}$. Funcția de transfer care aproximează pe $H_{\tau}(s)$ se notează $H_{\tau a}(s)$, unde

$$H_{\tau a}(s) = \frac{1 + c_1 \cdot s + c_2 \cdot s^2 + \dots + c_n \cdot s^n}{1 + d_1 \cdot s + d_2 \cdot s^2 + \dots + d_n \cdot s^n} \quad , b_n \neq 0. \quad (5.11)$$

Coefficienții funcției de transfer $H_{\tau a}(s)$ pot fi determinați egalând dezvoltarea în jurul originii a funcției $H_{\tau a}(s)$ cu dezvoltarea în jurul originii a funcției $H_{\tau}(s)$. O astfel de aproximare este cunoscută sub denumirea de aproximare Padé de ordinul $(n+k)$, unde n reprezintă gradul polinomului de la numitor, iar k gradul polinomului de la numărătorul lui $H_{\tau a}(s)$

În [46] s-au prezentat aproximațiile Padé uzuale de ordinul $(2+0)$, $(2+1)$, $(1+1)$ respectiv $(2+2)$

$$\begin{aligned} H_{\tau a1}(s) &= \frac{1}{1 + \tau s + \frac{\tau^2}{2} \cdot s^2} \quad , & H_{\tau a2}(s) &= \frac{1 - \frac{1}{3} \tau s}{1 + \frac{2}{3} \tau s + \frac{\tau^2}{2} s^2} \quad , \\ H_{\tau a3}(s) &= \frac{1 - \frac{\tau}{2} s}{1 + \frac{\tau}{2} s} \quad , & H_{\tau a4}(s) &= \frac{1 - \frac{\tau}{2} s + \frac{\tau^2}{12} \cdot s^2}{1 + \frac{\tau}{2} s + \frac{\tau^2}{12} \cdot s^2} \end{aligned} \quad (5.12)$$

5.2.2. Determinarea funcției sistem a sistemului discret

Pentru obținerea funcției sistem a sistemului discret se pornește de la forma aproximativă a funcției de transfer a sistemului, obținută prin utilizarea uneia dintre aproximațiile Padé ale funcției $e^{-s\tau}$ și prezentate în paragraful anterior

$$H_a(s) = \frac{K}{1 + s \cdot T} \cdot H_{\tau a}(s) \quad .$$

Funcția sistem a sistemului discret se obține prin utilizarea unei metode de echivalare a filtrului analogic cu un filtru numeric. Cele două metode de echivalare analizate de autor sunt:

1. metoda aproximării ecuației diferențiale prin diferențe finite, cunoscută și sub denumirea de metoda aproximării integralei prin metoda dreptunghiului, în care funcția sistem se obține pe baza relației

$$H(z) = H_a(s) \Big|_{s = \frac{1}{T_e} \cdot (1-z^{-1})} ;$$

2. metoda transformării biliniare, cunoscută și sub denumirea de metoda aproximării integralei prin metoda trapezelor, în care funcția sistem se obține pe baza relației

$$H(z) = H_a(s) \Big|_{s = \frac{2}{T_e} \cdot \frac{1-z^{-1}}{1+z^{-1}}}$$

Indiferent de aproximația Padé utilizată, expresia generală a funcției sistem poate fi scrisă sub forma

$$H(z) = \frac{b_0 + b_1 \cdot z + b_2 \cdot z^2 + b_3 \cdot z^3}{1 + a_1 \cdot z + a_2 \cdot z^2 + a_3 \cdot z^3}.$$

Autorul a determinat coeficienții funcției sistem în cazul utilizării ambelor metode de echivalare, rezultatele fiind prezentate în tabelul 5.4, pentru metoda aproximării integralei prin metoda dreptunghiului, respectiv în tabelele 5.5 și 5.6, pentru metoda aproximării integralei prin metoda trapezelor.

Se observă că numai în situația utilizării aproximației Padé (1+1) se obțin doi poli ai funcției sistem. De asemenea, se observă că valorile coeficienților funcției sistem depind de perioada de eșantionare, fapt care face ca poziția polilor funcției sistem să depindă de perioada de eșantionare, stabilitatea sistemului fiind, deci, influențată de aceasta.

Autorul a simulat comportarea filtrului obținut pentru cele 4 aproximații Padé, pentru diferite valori ale parametrilor sistemului și a ajuns la concluzia că pentru asigurarea convergenței coeficienților filtrului adaptiv la valorile coeficienților filtrului care ar rezulta pentru valorile alese ale parametrilor sistemului este necesară creșterea perioadei de eșantionare, odată cu creșterea numărului de poli ai funcției sistem.

În continuare s-a pus problema ca, în situația în care coeficienții filtrului adaptiv ating valorile corespunzătoare parametrilor sistemului, să se poată deduce valorile parametrilor sistemului pornind de la valorile coeficienților filtrului adaptiv. În tabelele 5.7 și 5.8 autorul a prezentat modul de obținere a parametrilor sistemului pornind de la coeficienții filtrului adaptiv, atât în situația utilizării metodei aproximării integralei prin metoda dreptunghiurilor, cât și prin metoda trapezelor. În cazul aproximației Padé (2+0) autorul nu a găsit relații simple de determinare a parametrilor sistemului pornind de la coeficienții filtrului adaptiv.

Tabelul 5.4. Valorile coeficienților funcției sistem a sistemului discret obținute utilizând cazul aproximării integralei prin metoda dreptunghiurilor.

	b_0	b_1	b_2	a_1	a_2	a_3
$s = \frac{1}{T_e}(1 - z^{-1})$						
$H_{r_1}(s) = \frac{1 - \frac{\tau}{2} s}{1 + \frac{\tau}{2} s}$ Padé (1+1)	$-\frac{kT_e(\tau - 2T_e)}{(T_e + T_e)(\tau + 2T_e)}$	$\frac{k\tau T_e}{(T_e + T_e)(2T_e + \tau)}$	0	$-\frac{2\tau T_e + 2T_e T_e + T_e \tau}{(T_e + T_e)(2T_e + \tau)}$	$\frac{\tau T_e}{(T_e + T_e)(\tau + 2T_e)}$	0
$H_{r_1}(s) = \frac{1}{1 + \tau s + \frac{\tau^2}{2} s^2}$ Padé (2+0)	$\frac{2kT_e^2}{(T_e + T_e)(2T_e^2 + 2\tau T_e + \tau^2)}$	0	0	$-\left(\frac{T_e}{T_e + T_e} + \frac{2\tau(T_e + \tau)}{2T_e^2 + 2\tau T_e + \tau^2} \right)$	$\frac{2\tau T_e(T_e + \tau)}{(T_e + \tau)(2T_e^2 + 2\tau T_e + \tau^2)} + \frac{\tau^2}{2T_e^2 + 2\tau T_e + \tau^2}$	$\frac{\tau T_e^2}{(T_e + T_e)(2T_e^2 + 2\tau T_e + \tau^2)}$
$H_{r_1}(s) = \frac{1 - \frac{1}{3} \tau s}{1 + \frac{2}{3} \tau s + \frac{\tau^2}{2} s^2}$ Padé (2+1)	$\frac{2kT_e^2(3T_e - \tau)}{(T_e + T_e)(6T_e^2 + 4\tau T_e + 3\tau^2)}$	$\frac{2k\tau T_e^2}{(T_e + T_e)(6T_e^2 + 4\tau T_e + 3\tau^2)}$	0	$-\left(\frac{4T_e + \tau}{6T_e^2 + 4\tau T_e + 3\tau^2} + \frac{T_e}{T_e + T_e} \right)$	$\frac{3\tau^2}{6T_e^2 + 4\tau T_e + 3\tau^2} + \frac{\tau^2}{4\tau T_e + 6\tau^2}$	$\frac{-3\tau^2 T_e}{(T_e + T_e)(6T_e^2 + 4\tau T_e + 3\tau^2)}$
$H_{r_1}(s) = \frac{1 - \frac{\tau}{2} s + \frac{\tau^2}{2} s^2}{1 + \frac{\tau}{2} s + \frac{\tau^2}{2} s^2}$ Padé (2+2)	$\frac{kT_e(6T_e^2 - 6\tau T_e + \tau^2)}{(T_e + T_e)(2T_e^2 + 6\tau T_e + \tau^2)}$	$\frac{kT_e(6\tau T_e - 2\tau^2)}{(T_e + T_e)(2T_e^2 + 6\tau T_e + \tau^2)}$	$\frac{kT_e\tau^2}{(T_e + T_e)(2T_e^2 + 6\tau T_e + \tau^2)}$	$\left(\frac{1}{(T_e + T_e)} + \frac{6\tau T_e + 2\tau^2}{12T_e^2 + 6\tau T_e + \tau^2} \right)$	$\frac{11(\tau T_e + 2\tau^2) + \tau^2(T_e + T_e)}{(T_e + T_e)(12T_e^2 + 6\tau T_e + \tau^2)}$	$\frac{\tau T_e}{(T_e + T_e)(6T_e^2 + 6\tau T_e + \tau^2)}$

Tabelul 5.5. Valorile coeficienților numărătorului funcției sistem a sistemului discret obținute utilizând cazul aproximării integralei prin metoda trapezelor (transformării biliniare).

$s = \frac{2}{T_0} \left(\frac{1-z^{-1}}{1+z^{-1}} \right)$	b_0	b_1	b_2	b_3
$H_{r1}(s) = \frac{1-\frac{\tau}{2}s}{1+\frac{\tau}{2}s}$ Padé (1+1)	$\frac{kT_0(T_0-\tau)}{(T_0+2T)(T_0+\tau)}$	$\frac{2kT_0^2}{(T_0+2T)(T_0+\tau)}$	$\frac{kT_0}{T_0+2T}$	0
$H_{r2}(s) = \frac{1}{1+\tau s + \frac{\tau^2}{2}s^2}$ Padé(2+0)	$\frac{kT_0^3}{(T_0+2T)[(T_0+\tau)^2+\tau^2]}$	$\frac{3kT_0^3}{(T_0+2T)[(T_0+\tau)^2+\tau^2]}$	$\frac{3kT_0^3}{(T_0+2T)[(T_0+\tau)^2+\tau^2]}$	$\frac{kT_0^3}{(T_0+2T)[(T_0+\tau)^2+\tau^2]}$
$H_{r3}(s) = \frac{1-\frac{1}{3}\tau s}{1+\frac{2}{3}\tau s + \frac{\tau^2}{2}s^2}$ Padé(2+1)	$\frac{kT_0^2(3T_0-2\tau)}{(T_0+2T)[(T_0+2\tau)^2+2(T_0^2+\tau^2)]}$	$\frac{kT_0^2(9T_0-2\tau)}{(T_0+2T)[(T_0+2\tau)^2+2(T_0^2+\tau^2)]}$	$\frac{kT_0^2(9T_0+2\tau)}{(T_0+2T)[(T_0+2\tau)^2+2(T_0^2+\tau^2)]}$	$\frac{kT_0^2(3T_0+2\tau)}{(T_0+2T)[(T_0+2\tau)^2+2(T_0^2+\tau^2)]}$
$H_{r4}(s) = \frac{1-\frac{\tau}{2}s + \frac{\tau^2}{12}s^2}{1+\frac{\tau}{2}s + \frac{\tau^2}{12}s^2}$ Padé (2+2)	$\frac{kT_0(3T_0^2-3\tau T_0+\tau^2)}{(T_0+2T)(3T_0^2+3\tau T_0+\tau^2)}$	$\frac{kT_0(9T_0^2-3\tau T_0-\tau^2)}{(T_0+2T)(3T_0^2+3\tau T_0+\tau^2)}$	$\frac{kT_0(9T_0^2+3\tau T_0-\tau^2)}{(T_0+2T)(3T_0^2+3\tau T_0+\tau^2)}$	$\frac{kT_0}{T_0+2T}$

Tabelul 5.6. Valorile coeficienților numitorului funcției sistem a sistemului discret obținute utilizând cazul aproximării integralei prin metoda trapezelor (transformării biliniare).

$s = \frac{2}{T_0} \left(\frac{1-z^{-1}}{1+z^{-1}} \right)$	a_1	a_2	a_3
$H_1(s) = \frac{1 - \frac{\tau}{2} s}{1 + \frac{\tau}{2} s}$ Padé (1+1)	$\frac{2(T_0^2 - 2T_0\tau)}{(T_0 + 2T_0)(T_0 + \tau)}$	$\frac{(T_0 - 2T_0)(T_0 - \tau)}{(T_0 + 2T_0)(T_0 + \tau)}$	0
$H_2(s) = \frac{1}{1 + \tau s + \frac{\tau^2}{2} s^2}$ Padé(2+0)	$\frac{T_0 - 2T_0}{T_0 + 2T_0} + \frac{2(T_0^2 - 2\tau^2)}{(T_0 + \tau)^2 + \tau^2}$	$\frac{T_0 - 2T_0}{T_0 + 2T_0} \cdot \frac{2(T_0^2 - 2\tau^2)}{(T_0 + \tau)^2 + \tau^2} + \frac{(T_0 - \tau)^2 + \tau^2}{(T_0 + \tau)^2 + \tau^2}$	$\frac{(T_0 - 2T_0)(T_0 - \tau)^2 + \tau^2}{(T_0 + 2T_0)(T_0 + \tau)^2 + \tau^2}$
$H_3(s) = \frac{1 - \frac{1}{3}\tau s}{1 + \frac{2}{3}\tau s + \frac{\tau^2}{2} s^2}$ Padé(2+1)	$\frac{(T_0 - 2T_0)}{(T_0 + 2T_0)} + \frac{6(T_0^2 - 2\tau^2)}{(T_0 + 2\tau)^2 + 2(T_0^2 + \tau^2)}$	$\frac{(T_0 - 2T_0)}{(T_0 + 2T_0)} \cdot \frac{6(T_0^2 - 2\tau^2)}{(T_0 + 2\tau)^2 + 2(T_0^2 + \tau^2)} + \frac{(T_0 - 2\tau)^2 + 2(T_0^2 + \tau^2)}{(T_0 + 2\tau)^2 + 2(T_0^2 + \tau^2)}$	$\frac{(T_0 - 2T_0)}{(T_0 + 2T_0)} \cdot \frac{(T_0 - 2\tau)^2 + 2(T_0^2 + \tau^2)}{(T_0 + 2\tau)^2 + 2(T_0^2 + \tau^2)}$
$H_4(s) = \frac{1 - \frac{\tau}{2} s + \frac{\tau^2}{12} s^2}{1 + \frac{\tau}{2} s + \frac{\tau^2}{2} s^2}$ Padé (2+2)	$\frac{T_0 - 2T_0}{T_0 + 2T_0} + \frac{2(3T_0^2 - \tau^2)}{3T_0^2 + 3\tau T_0 + \tau^2}$	$\frac{2(T_0 - 2T_0)(3T_0^2 - \tau^2)}{(T_0 + 2T_0)(3T_0^2 + 3\tau T_0 + \tau^2)} + \frac{3T_0^2 - 3\tau T_0 + \tau^2}{3T_0^2 + 3\tau T_0 + \tau^2}$	$\frac{2(T_0 - 2T_0)(3T_0^2 - 3\tau T_0 + \tau^2)}{(T_0 + 2T_0)(3T_0^2 + 3\tau T_0 + \tau^2)}$

Tabelul 5.7. Determinarea valorilor parametrilor sistemului în cazul utilizării metodei aproximării integralei prin metoda dreptunghiurilor.

$s = \frac{1}{T_c}(1-z^{-1})$	τ	T	k
$H_{\tau}(s) = \frac{1 - \frac{\tau}{2}s}{1 + \frac{\tau}{2}s}$ Padé (1+1)	$\frac{2T_c}{\frac{b_0}{b_1} + 1}$	$\frac{T_c}{-\frac{a_1}{a_2} - 2 - \frac{2T_c}{\tau}}$	$\frac{b_1}{a_2} \cdot \frac{T}{T_c}$
Padé (2+0)	-	-	-
$H_{\tau}(s) = \frac{1 - \frac{1}{3}\tau s}{1 + \frac{2}{3}\tau s + \frac{\tau^2}{2}s^2}$ Padé (2+1)	$\frac{3T_c}{\frac{b_0}{b_1} + 1}$	$\frac{-T_c}{\frac{3\tau^2}{(6T_c^2 + 4\tau T_c + 3\tau^2) \cdot a_3} + 1}$	$-\frac{3}{2} \frac{b_1}{a_3} \cdot \frac{T}{T_c^2} \cdot \tau$
$H_{\tau}(s) = \frac{1 - \frac{\tau}{2}s + \frac{\tau^2}{12}s^2}{1 + \frac{\tau}{2}s + \frac{\tau^2}{12}s^2}$ Padé (2+2)	$\frac{3T_c}{\frac{1}{2} \frac{b_1}{b_2} + 1}$	$\frac{-T_c}{\frac{\tau^2}{(12T_c^2 + 6\tau T_c + \tau^2) \cdot a_1} + 1}$	$\frac{b_0(T_c + T)(12T_c^2 + 6\tau T_c + \tau^2)}{T_c(12T_c^2 - 6\tau T_c + \tau^2)}$

Tabelul 5.8. Determinarea valorilor parametrilor sistemului în cazul utilizării metodei aproximării integralei prin metoda trapezelor.

$s = \frac{2}{T_c} \left(\frac{1-z^{-1}}{1+z^{-1}} \right)$	τ	T	k
$H_{\tau}(s) = \frac{1 - \frac{\tau}{2}s}{1 + \frac{\tau}{2}s}$ Padé (1+1)	$\left(1 - \frac{2b_0}{b_1}\right) T_c$	$\frac{T_c}{2} \cdot \frac{2a_2 T_c - a_1 T_c + a_1 \tau}{2a_2 \tau - a_1 T_c + a_1 \tau}$	$\frac{2b_0(T_c^2 - 2T\tau)}{a_1 T_c(T_c - \tau)}$
Padé (2+0)	-	-	-
$H_{\tau}(s) = \frac{1 - \frac{1}{3}\tau s}{1 + \frac{2}{3}\tau s + \frac{\tau^2}{2}s^2}$ Padé (2+1)	$\frac{3}{2} T_c \frac{\frac{b_1}{b_0} - 3}{\frac{b_1}{b_0} - 1}$	$\frac{T_c}{2} \cdot \frac{1 - a_1 + \frac{6(T_c^2 - 2\tau^2)}{(T_c + 2\tau)^2 + 2(T_c^2 + \tau^2)}}{1 + a_1 - \frac{6(T_c^2 - 2\tau^2)}{(T_c + 2\tau)^2 + 2(T_c^2 + \tau^2)}}$	$\frac{b_0(T_c + 2T)(T_c + 2\tau)^2 + 2(T_c^2 + \tau^2)}{T_c^2(3T_c - 2\tau)}$
$H_{\tau}(s) = \frac{1 - \frac{\tau}{2}s + \frac{\tau^2}{12}s^2}{1 + \frac{\tau}{2}s + \frac{\tau^2}{12}s^2}$ Padé (2+2)	-	$\frac{T_c}{2} \cdot \frac{b_0 - b_1 a_3}{b_0 + b_1 a_3}$	$b_1 \frac{T_c + 2T}{T_c}$

5.2.3. Studiul algoritmilor de identificare și a formelor de implementare

Deoarece, așa cum s-a prezentat anterior, valorile parametrilor sistemului pot fi determinate cunoscând valorile coeficienților filtrului adaptiv, autorul și-a pus problema alegerii algoritmului de identificare optim din punct de vedere al vitezei de convergență, precum și al stabilității. De asemenea și-a pus problema alegerii formei de implementare, directă sau lattice, precum și a metodei de echivalare a filtrului analogic cu un filtru numeric.

În acest scop autorul a implementat cei trei algoritmi de identificare prezentați în capitolul trei, atât în formă directă cât și în forma lattice și a testat algoritmi în condiții identice. Valorile alese pentru parametrii sistemului simulat au fost $\tau=4$ secunde, $T=7$ secunde, $k=5$, iar perioada de eșantionare aleasă a fost $T_e=2$ secunde. Testarea s-a făcut considerând sistemul necunoscut ca fiind un filtru fix, cu coeficienții obținuți pe baza relațiilor prezentate în tabelele 5.4. 5.6.

În continuare sunt prezentați cei trei algoritmi de identificare, în cele două forme de implementare, aplicați în situația utilizării aproximației Padé (1+1). În fiecare caz s-au utilizat ambele metode de echivalare a filtrelor analogice cu filtre numerice.

1.1. Algoritmul gradientului, forma de implementare directă

$$\begin{pmatrix} \mathbf{a}(n+1) \\ \mathbf{b}(n+1) \end{pmatrix} = \begin{pmatrix} \mathbf{a}(n) \\ \mathbf{b}(n) \end{pmatrix} + \mu \cdot \begin{pmatrix} -\mathbf{Y}_A(n) \\ \mathbf{X}_A(n) \end{pmatrix} \cdot e(n).$$

Variația coeficienților filtrului adaptiv este prezentată în figura 5.2 pentru metoda aproximării integralei prin metoda dreptunghiurilor, respectiv în figura 5.8 pentru metoda aproximării integralei prin metoda trapezelor.

1.2. Algoritmul gradientului, forma de implementare lattice

$$\begin{pmatrix} \mathbf{k}(n+1) \\ \mathbf{b}(n+1) \end{pmatrix} = \begin{pmatrix} \mathbf{k}(n) \\ \mathbf{b}(n) \end{pmatrix} + \mu \cdot \begin{pmatrix} -\mathbf{U}_A(n) \\ \mathbf{X}_A(n) \end{pmatrix} \cdot e(n).$$

Variația coeficienților filtrului adaptiv este prezentată în figura 5.3 pentru metoda aproximării integralei prin metoda dreptunghiurilor, respectiv în figura 5.9 pentru metoda aproximării integralei prin metoda trapezelor.

2.1. Algoritmul SHARF, forma de implementare directă

$$\begin{pmatrix} \mathbf{a}(n+1) \\ \mathbf{b}(n+1) \end{pmatrix} = \begin{pmatrix} \mathbf{a}(n) \\ \mathbf{b}(n) \end{pmatrix} + \mu \cdot \begin{pmatrix} -\mathbf{y}(n) \\ \mathbf{x}(n) \end{pmatrix} \cdot c(n),$$

unde

$$c(n) = e(n) - 0.6 \cdot e(n-1).$$

Variația coeficienților filtrului adaptiv este prezentată în figura 5.4 pentru metoda aproximării integralei prin metoda dreptunghiurilor, respectiv în figura 5.10 pentru metoda aproximării integralei prin metoda trapezelor.

2.2. Algoritmul SHARF, forma de implementare lattice

$$\begin{pmatrix} \mathbf{k}(n+1) \\ \mathbf{b}(n+1) \end{pmatrix} = \begin{pmatrix} \mathbf{k}(n) \\ \mathbf{b}(n) \end{pmatrix} + \mu \cdot \begin{pmatrix} -\mathbf{u}(n) \\ \mathbf{x}(n) \end{pmatrix} \cdot c(n),$$

unde

$$c(n) = e(n) - 0.6 \cdot e(n-1).$$

Variația coeficienților filtrului adaptiv este prezentată în figura 5.5 pentru metoda aproximării integralei prin metoda dreptunghiurilor, respectiv în figura 5.11 pentru metoda aproximării integralei prin metoda trapezelor.

3.1. Algoritmul Steiglitz Mc-Bride, forma de implementare directă

$$\begin{pmatrix} \mathbf{a}(n+1) \\ \mathbf{b}(n+1) \end{pmatrix} = \begin{pmatrix} \mathbf{a}(n) \\ \mathbf{b}(n) \end{pmatrix} + \mu \cdot \begin{pmatrix} -\mathbf{D}_A(n) \\ \mathbf{X}_A(n) \end{pmatrix} \cdot e(n).$$

Variația coeficienților filtrului adaptiv este prezentată în figura 5.6 pentru metoda aproximării integralei prin metoda dreptunghiurilor, respectiv în figura 5.12 pentru metoda aproximării integralei prin metoda trapezelor.

3.2. Algoritmul Steiglitz Mc-Bride, forma de implementare lattice

$$\begin{pmatrix} \mathbf{k}(n+1) \\ \mathbf{b}(n+1) \end{pmatrix} = \begin{pmatrix} \mathbf{k}(n) \\ \mathbf{b}(n) \end{pmatrix} + \mu \cdot \begin{pmatrix} -\mathbf{U}_A(n) \\ \mathbf{X}_A(n) \end{pmatrix} \cdot e(n).$$

Variația coeficienților filtrului adaptiv este prezentată în figura 5.7 pentru metoda aproximării integralei prin metoda dreptunghiurilor, respectiv în figura 5.13 pentru metoda aproximării integralei prin metoda trapezelor.

Autorul a constatat că în situația utilizării algoritmului Steiglitz Mc-Bride, pentru a se asigura convergența coeficienților filtrului adaptiv este necesară o perioadă de eşantionare mai mare decât în cazul utilizării celorlalți algoritmi, prin perioadă de eşantionare înțelegându-se intervalul de timp între două reactualizări succesive ale valorilor coeficienților filtrului adaptiv. De asemenea, s-a constatat experimental că, repetând simularea de mai multe ori, nu întotdeauna se obține o convergență a coeficienților filtrului adaptiv. Acest fapt se datorează unei variații mari a coeficienților la începutul procesului de adaptare, aceasta ducând de multe ori la instabilitatea algoritmului. Din acest motiv autorul nu a utilizat acest algoritm în procesul de reglare a temperaturii la cuptorul cu rezistență electrică.

În ceea ce privește atât algoritmul gradientului, cât și algoritmul SHARF, se observă că, indiferent de modul de echivalare a filtrului analogic cu un filtru numeric, forma de implementare lattice prezintă o viteză de convergență mai mare decât forma de implementare directă. Față de algoritmul gradientului, algoritmul SHARF prezintă oscilații ale valorilor coeficienților filtrului adaptiv mai mici, indiferent de forma de implementare sau de metoda de echivalare a filtrului analogic cu un filtru numeric.

Autorul a constatat că, utilizând valorile finale ale coeficienților filtrului adaptiv, valorile obținute pentru parametrii modelului de încălzire al cuptorului sunt aceleași cu cele pe baza cărora au fost calculați coeficienții filtrului care aproximează modelul sistemului.

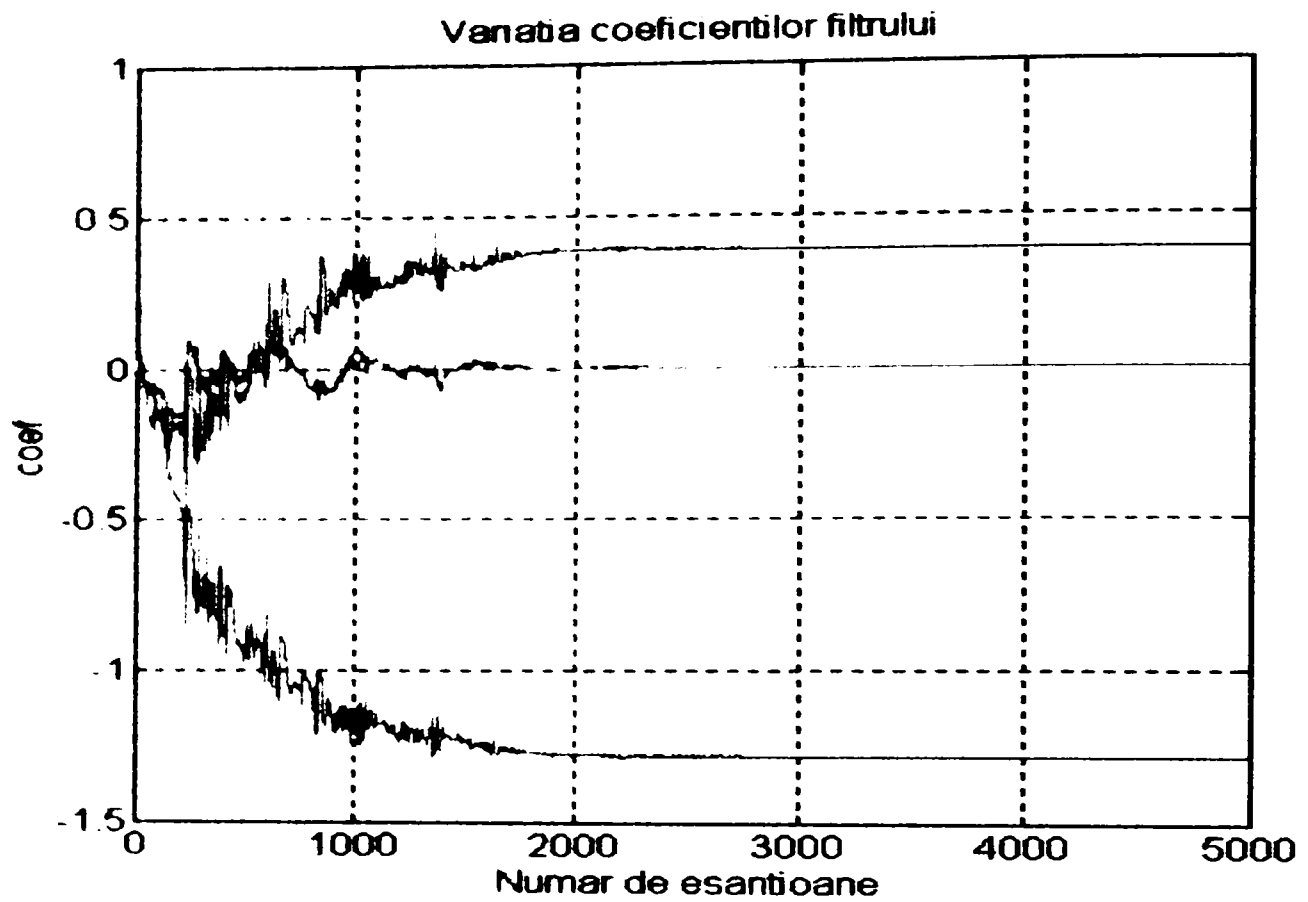


Fig. 5.2. Variația coeficienților filtrului adaptiv utilizând aproximația Padé (1+1), algoritmul gradientului implementat în forma directă ($\alpha = 0.2$; $T_e = 2s$).

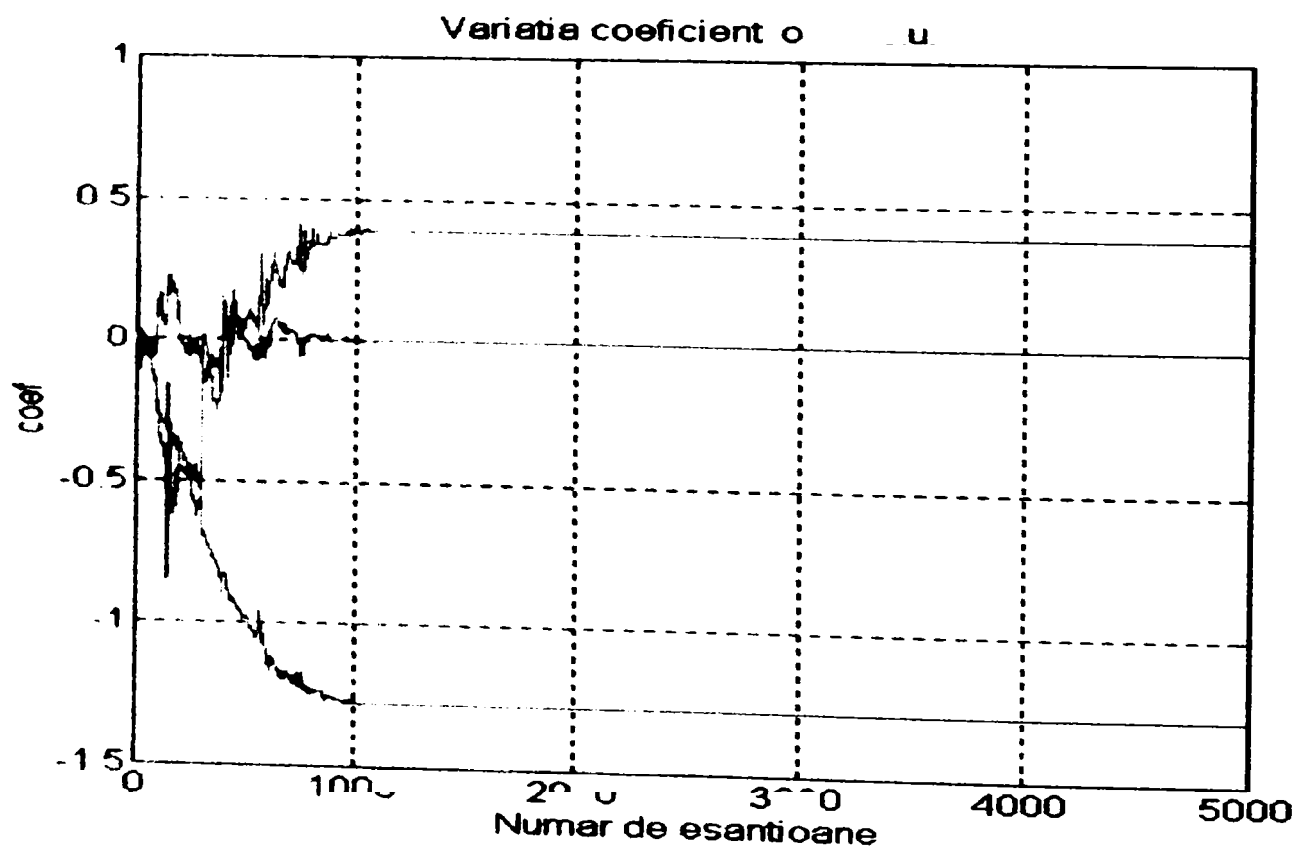


Fig. 5.3. Variația coeficienților filtrului adaptiv utilizând aproximația Padé (1+1), algoritmul gradientului implementat în forma lattice ($\alpha = 0.2$; $T_e = 2s$).

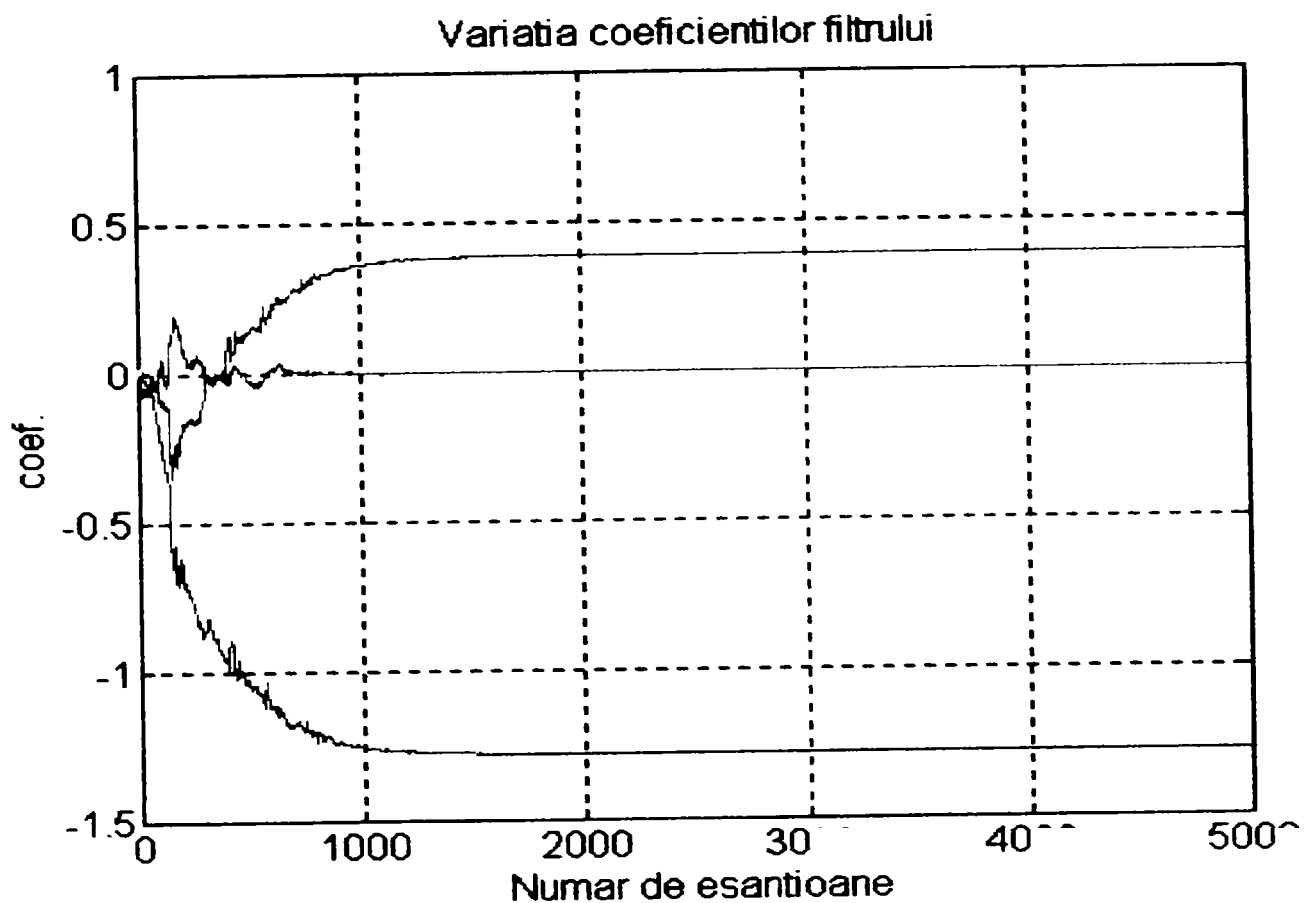


Fig. 5.4. Variația coeficienților filtrului adaptiv utilizând aproximația Padé (1+1), algoritmul SHARF implementat în forma directă ($\alpha = 0.2$; $T_c = 2s$).

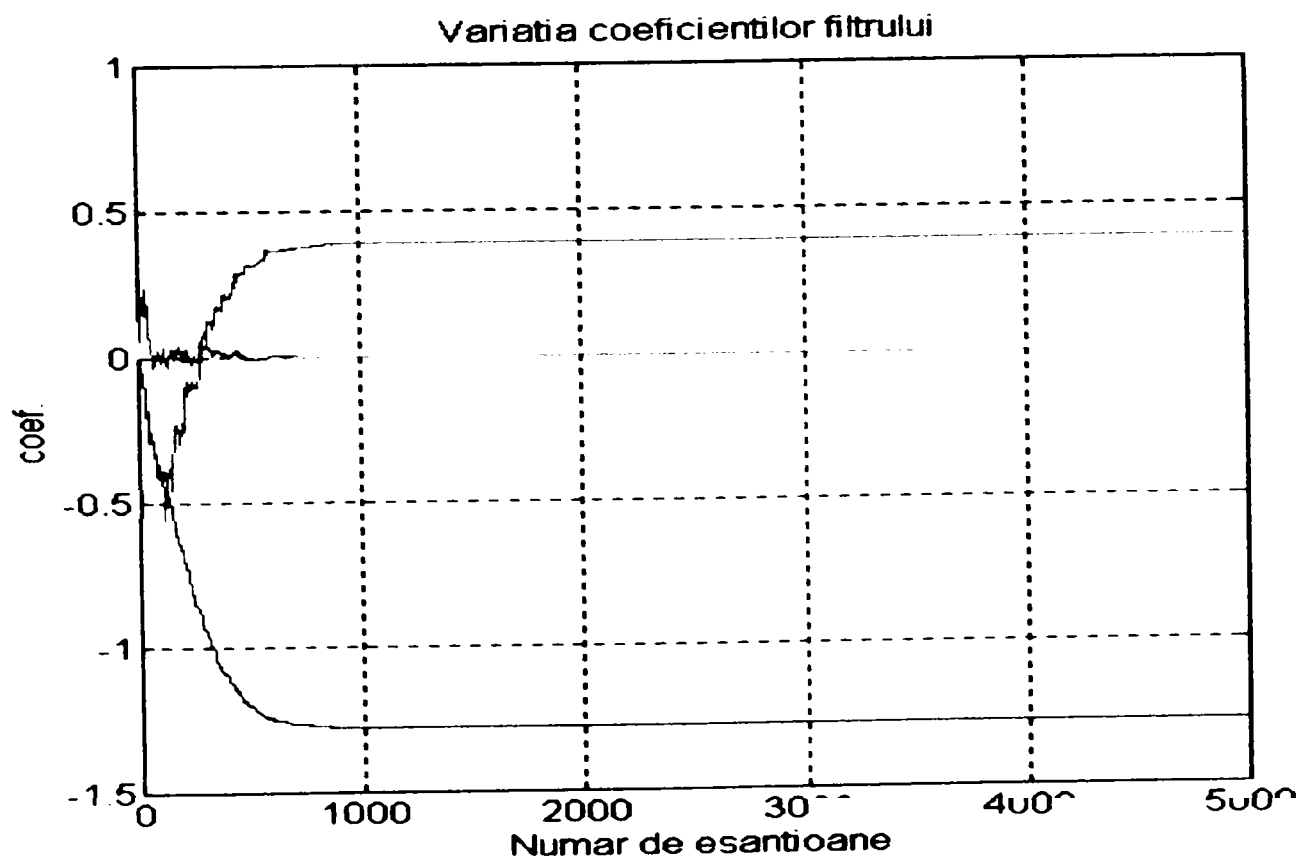


Fig. 5.5. Variația coeficienților filtrului adaptiv utilizând aproximația Padé (1+1), algoritmul SHARF implementat în forma lattice ($\alpha = 0.2$; $T_c = 2s$).

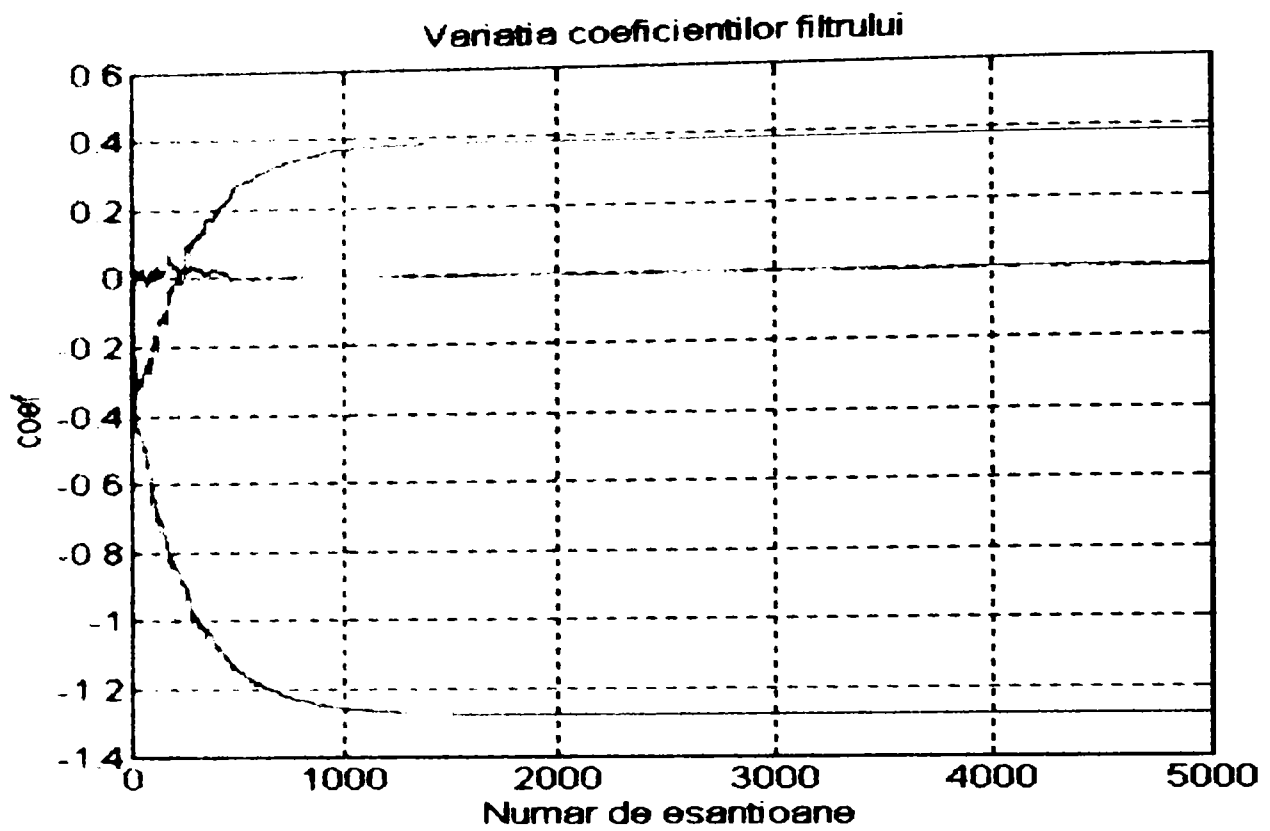


Fig. 5.6. Variația coeficienților filtrului adaptiv utilizând aproximația Padé (1+1), algoritmul Steiglitz Mc-Bride implementat în forma directă ($\alpha = 0.2$; $T_c = 2s$).

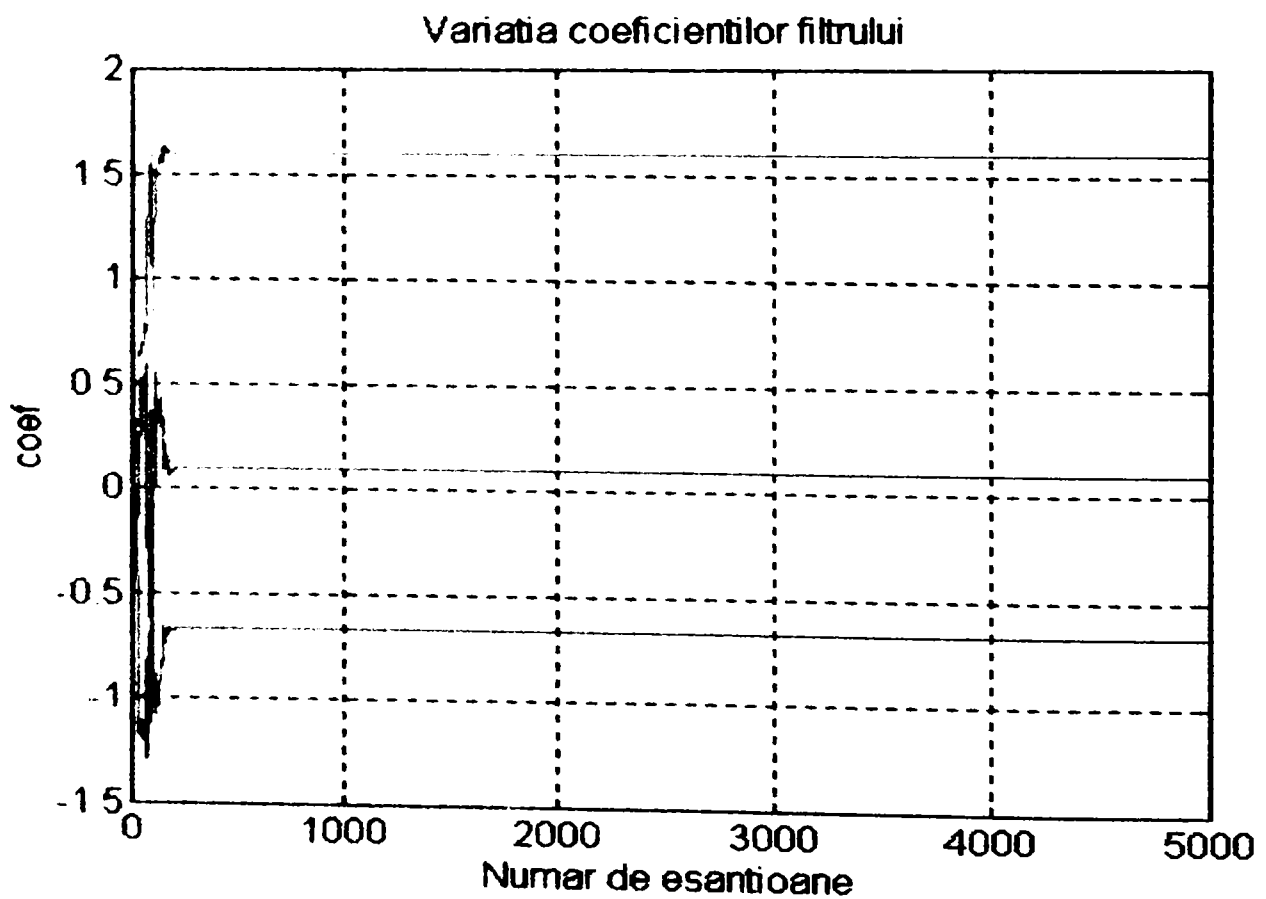


Fig. 5.7. Variația coeficienților filtrului adaptiv utilizând aproximația Padé (1+1), algoritmul Steiglitz Mc-Bride implementat în forma lattice ($\alpha = 0.2$; $T_c = 8s$).

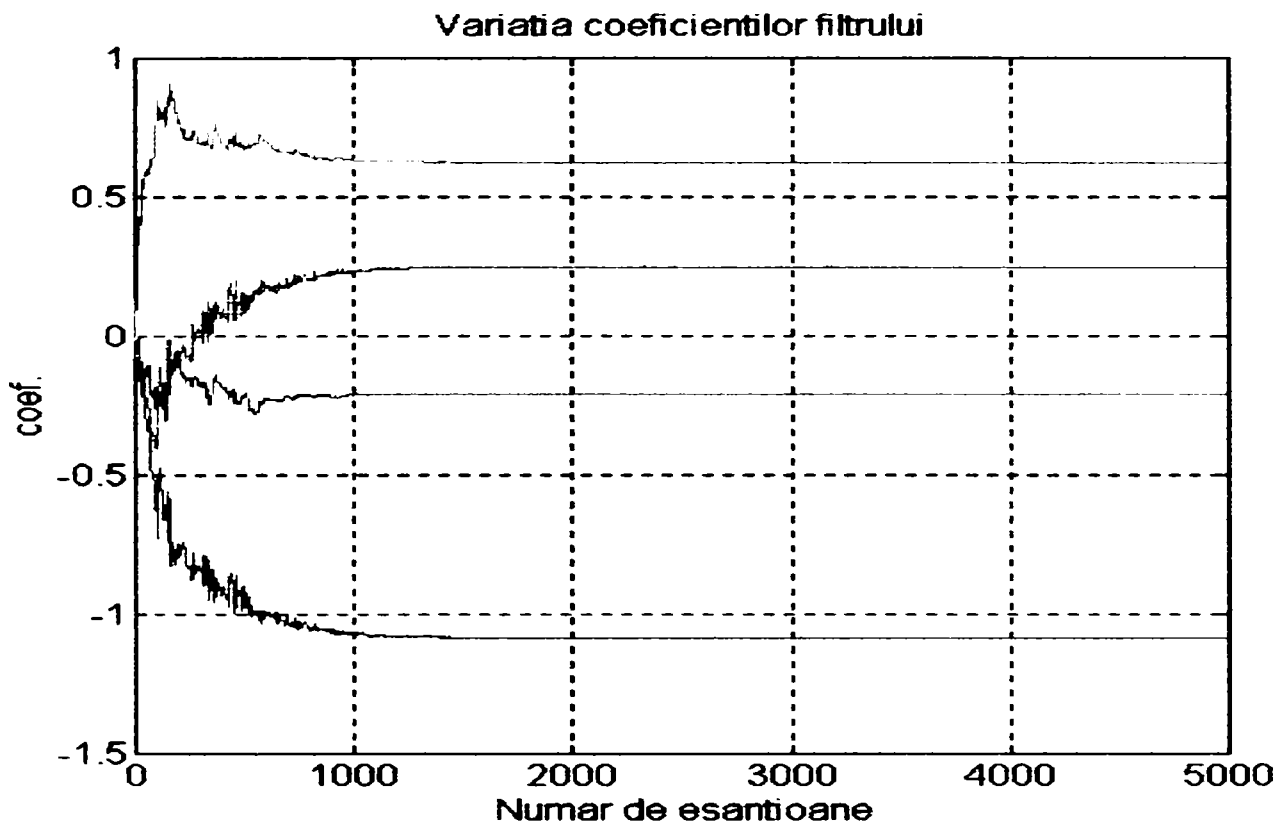


Fig. 5.8. Variația coeficienților filtrului adaptiv utilizând aproximația Padé (1+1), algoritmul gradientului implementat în forma directă ($\alpha = 0.2$; $T_c = 2s$).

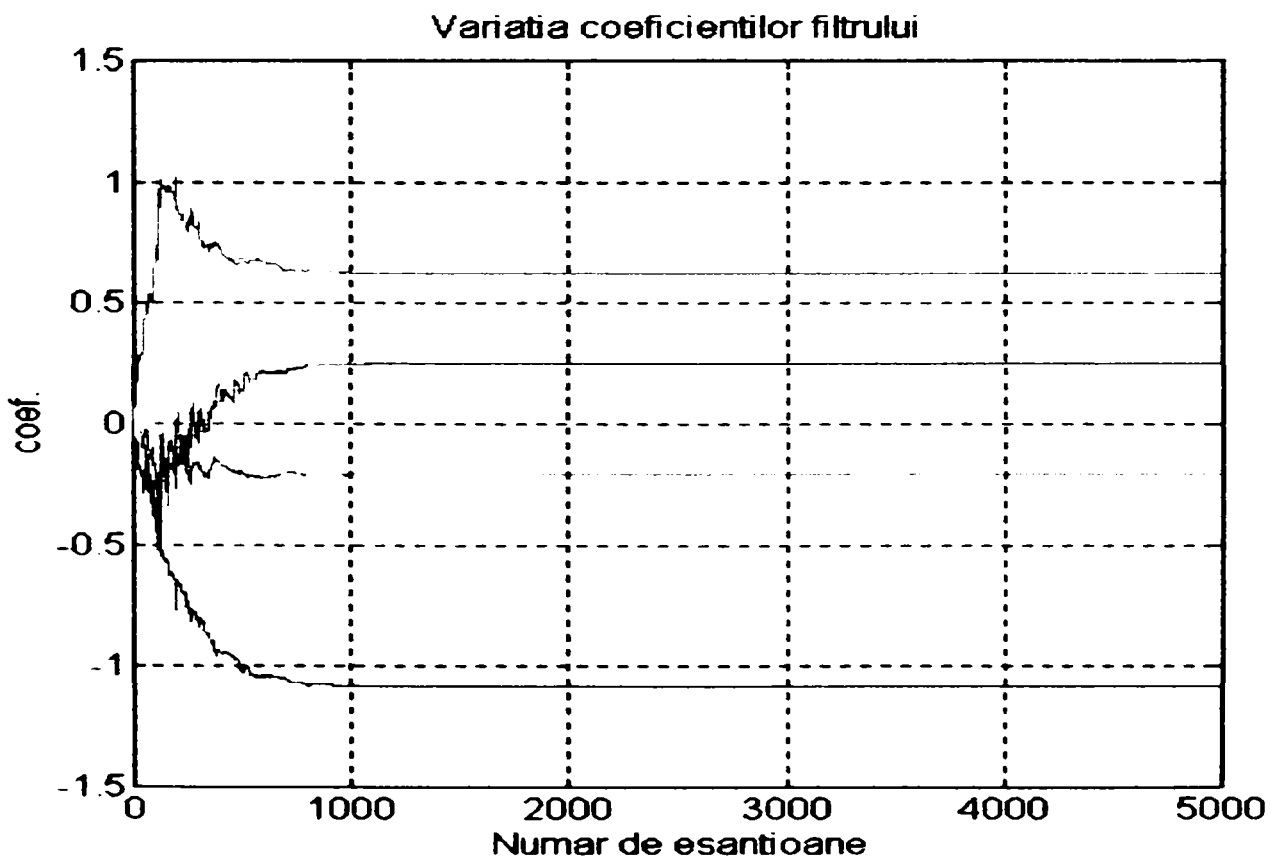


Fig. 5.9. Variația coeficienților filtrului adaptiv utilizând aproximația Padé (1+1), algoritmul gradientului implementat în forma lattice ($\alpha = 0.2$; $T_c = 2s$).

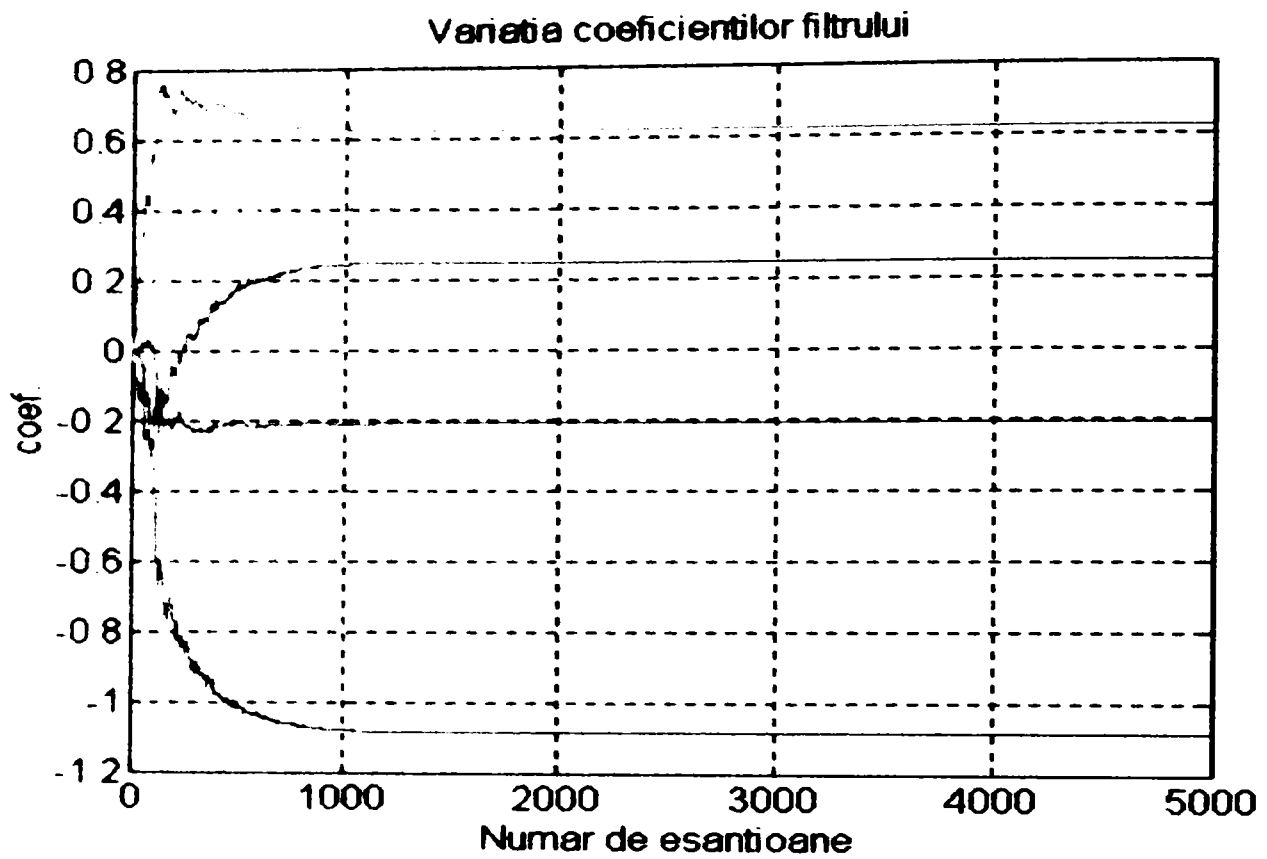


Fig. 5.10. Variația coeficienților filtrului adaptiv utilizând aproximația Padé (1+1), algoritmul SHARF implementat în forma directă ($\alpha = 0.2$; $T_e = 2s$).

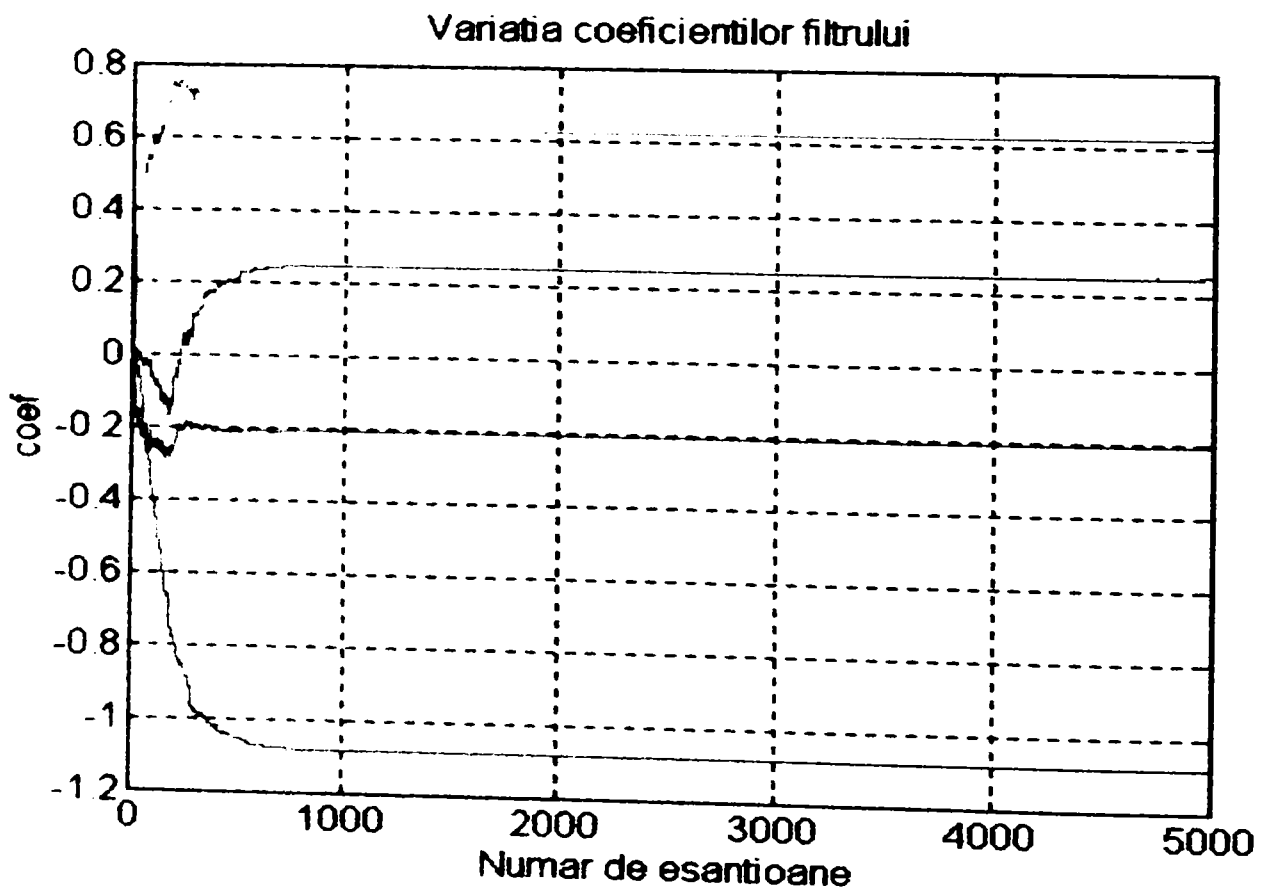


Fig. 5.11. Variația coeficienților filtrului adaptiv utilizând aproximația Padé (1+1), algoritmul SHARF implementat în forma lattice ($\alpha = 0.2$; $T_e = 2s$).

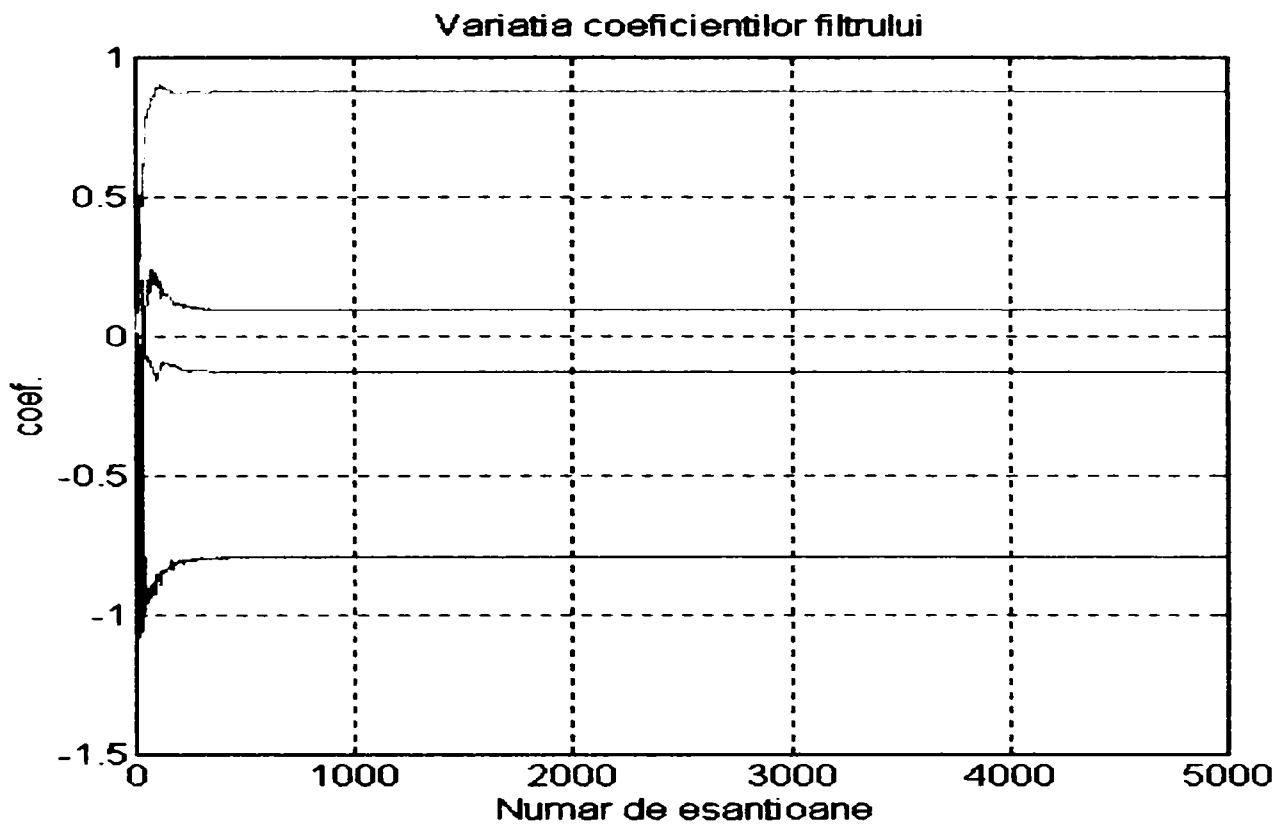


Fig. 5.12. Variația coeficienților filtrului adaptiv utilizând aproximația Padé (1+1), algoritmul Steiglitz Mc-Bride implementat în forma directă ($\alpha = 0.2$; $T_c = 3s$).

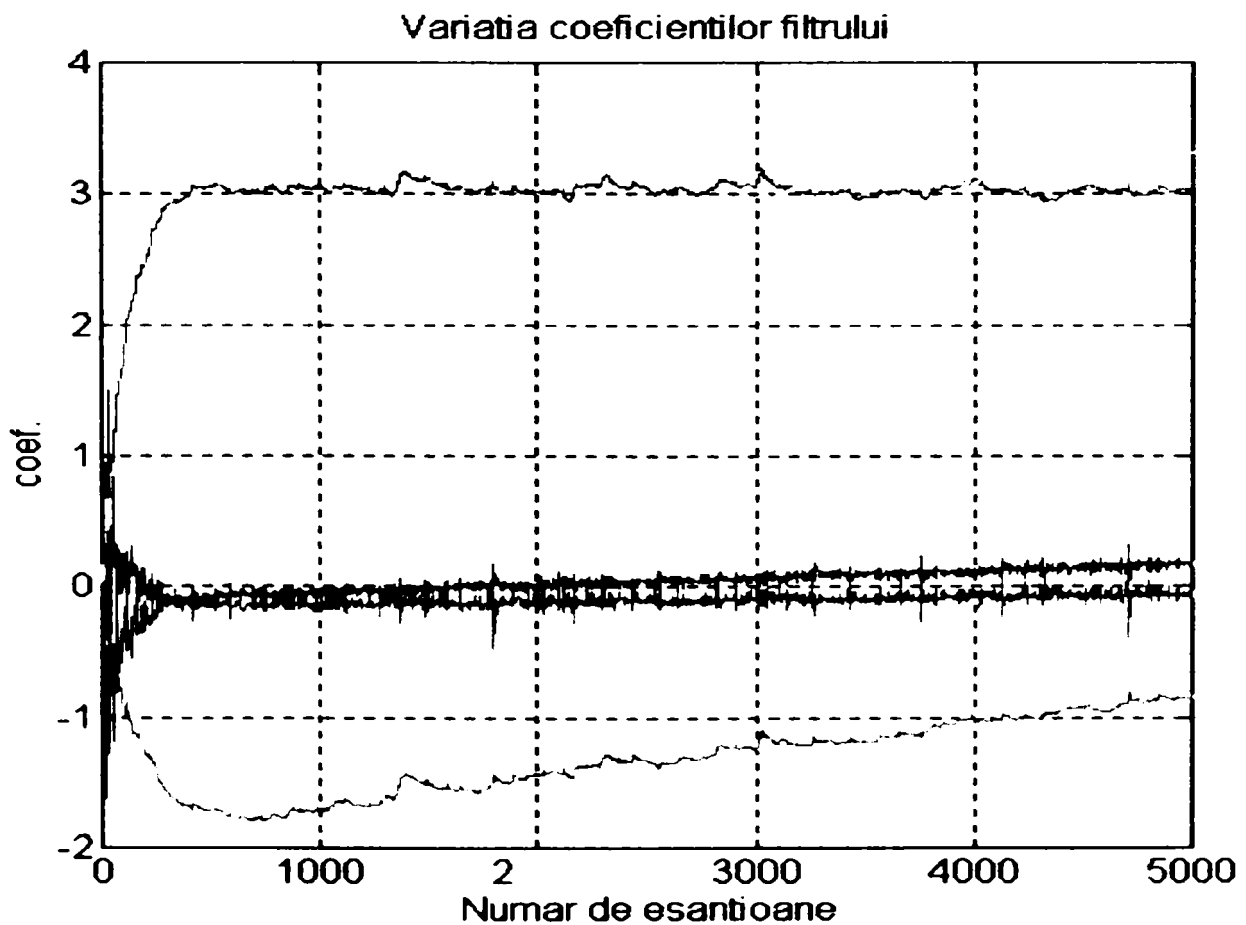


Fig. 5.13. Variația coeficienților filtrului adaptiv utilizând aproximația Padé (1+1), algoritmul Steiglitz Mc-Bride implementat în forma lattice ($\alpha = 0.2$; $T_c = 40s$).

5.2.4. Validarea alegerii funcției sistem aproximativă

În paragraful precedent, autorul a verificat că, utilizând unul din cei trei algoritmi de identificare implementați în ambele forme, (directă și lattice) și una dintre cele două metode de echivalare a filtrului analogic cu un filtru numeric, se obține un filtru cu coeficienți care depind de valorile parametrilor procesului și de perioada de eșantionare. Coeficienții filtrului adaptiv tind spre aceleași valori, convergența acestora depinzând de algoritmul de identificare utilizat și de forma de implementare.

Autorul și-a pus problema dacă răspunsul filtrului adaptiv obținut pe baza utilizării unei aproximații Padé este același cu cel obținut la ieșirea unui sistem cu funcția de transfer identică cu funcția de transfer a modelului ales, dată de relația (5.1). În acest scop a comparat semnalul de la ieșirea sistemului cu funcția de transfer dată de relația (5.1) cu semnalul de la ieșirea filtrului adaptiv în situația utilizării algoritmului gradientului implementat în forma directă, utilizând metoda aproximării integralei prin metoda dreptunghiurilor. Valorile alese pentru parametrii sistemului au fost $\tau = 4$ secunde, $T = 7$ secunde, $k = 5$, iar perioada de eșantionare aleasă a fost $T_e = 2$ secunde. Utilizând aproximarea Padé (1+1), valorile coeficienților filtrului rezultat sunt: $a_1 = -1,2778$; $a_2 = 0,3889$; $b_0 = 0$ și $b_1 = 0,5556$.

Semnalul aplicat atât la intrarea sistemului necunoscut cu funcția de transfer dată de relația (5.1), cât și la intrarea filtrului adaptiv a fost un semnal dreptunghiular cu amplitudine unitară, semnal pentru care se poate deduce ușor răspunsul sistemului necunoscut. În figura 5.14 este reprezentată forma de variație a coeficienților filtrului adaptiv în cazul în care valoarea parametrului de adaptare este $\mu = 0,001$.

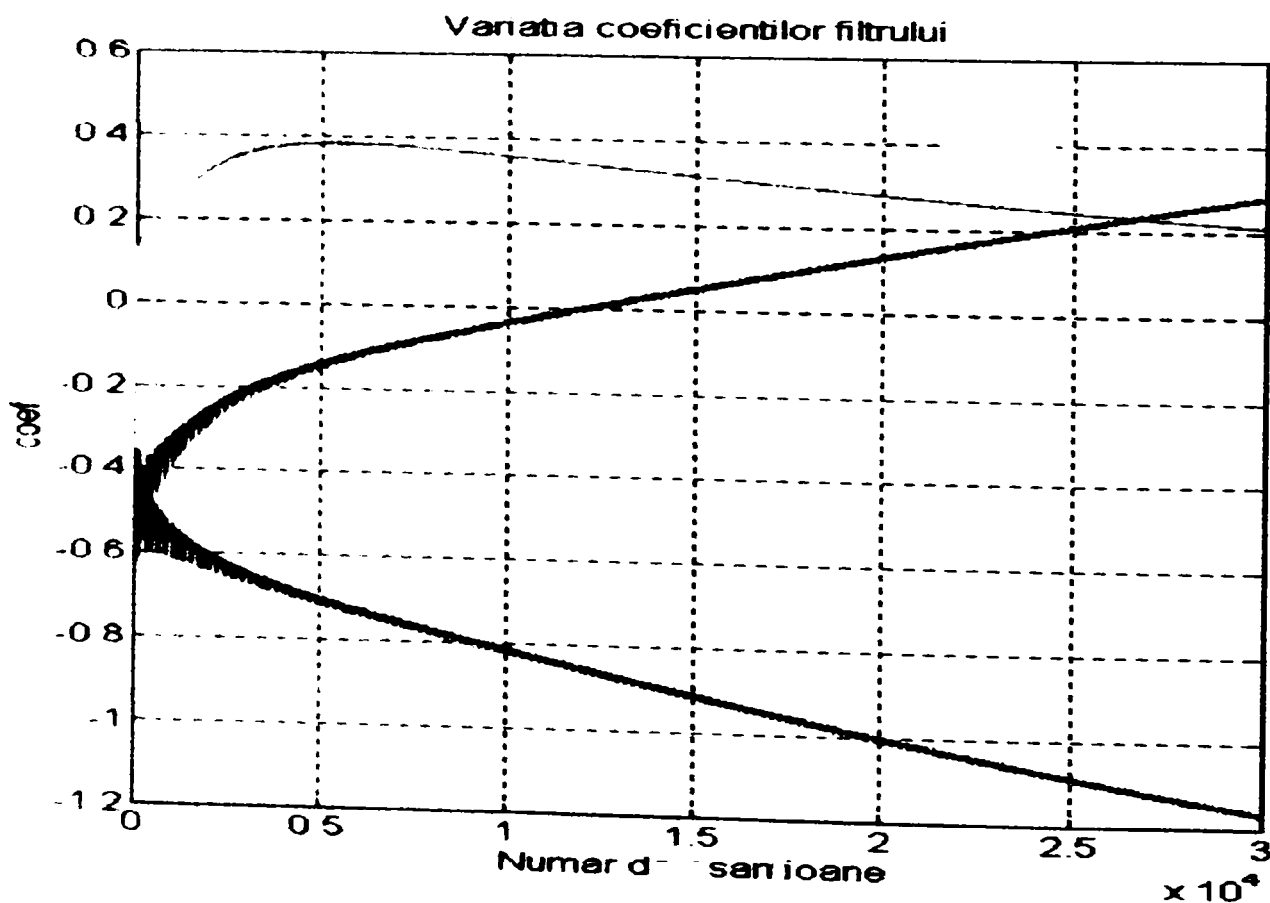


Fig. 5.14. Forma de variație a coeficienților filtrului adaptiv pentru $\mu = 0,001$.

Odată cu creșterea valorii parametrului de adaptare, deși crește viteza de convergență, coeficienții filtrului adaptiv prezintă oscilații mai mari, fapt care face ca filtrul adaptiv să devină instabil sau ca erorile obținute în determinarea parametrilor sistemului pe baza valorilor coeficienților filtrului adaptiv să devină inadmisibil de mari.

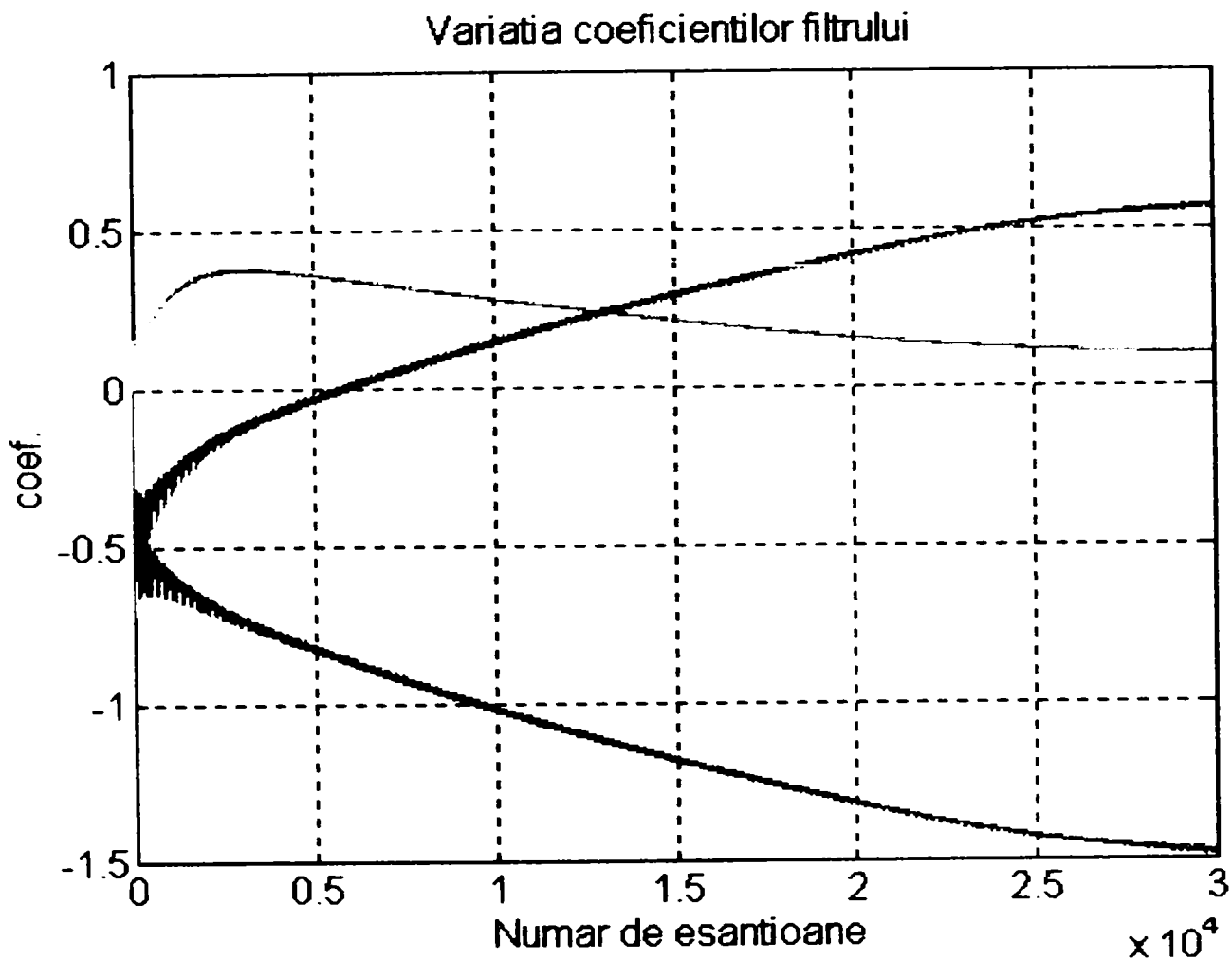


Fig. 5.15. Forma de variație a coeficienților filtrului adaptiv pentru $\mu = 0,0015$.

Se observă că, în ipoteza că valorile inițiale ale coeficienților filtrului adaptiv sunt nule, numărul de eșantioane după care se realizează adaptarea filtrului este de ordinul a zeci de mii, fapt care face ca procesul de măsurare să dureze foarte mult. **Este necesar ca, în situația utilizării filtrului adaptiv la identificarea parametrilor proceselor lente, pe baza unei măsurători anterioare utilizând altă metodă, să se determine cu aproximație valorile parametrilor sistemului și utilizând aceste valori să se determine setul de valori inițiale ale coeficienților filtrului adaptiv.**

Autorul a verificat experimental faptul că, dacă se aleg valorile inițiale ale coeficienților filtrului adaptiv egale cu cele rezultate pe baza măsurării valorilor parametrilor sistemului necunoscut, filtrul adaptiv rezultat este stabil. În figura 5.16 este reprezentată forma de variație a semnalului de la ieșirea sistemului necunoscut, forma de variație a semnalului de la ieșirea filtrului adaptiv și forma de variație a coeficienților filtrului adaptiv în situația aplicării la intrarea sistemului a semnalului dreptunghiular de amplitudine unitară. Se observă că semnalul de la ieșirea filtrului adaptiv este similar cu cel de la ieșirea sistemului necunoscut, iar coeficienții filtrului adaptiv sunt aproximativ constanți, fapt ce arată că filtrul adaptiv reprezintă o bună aproximație a sistemului necunoscut.

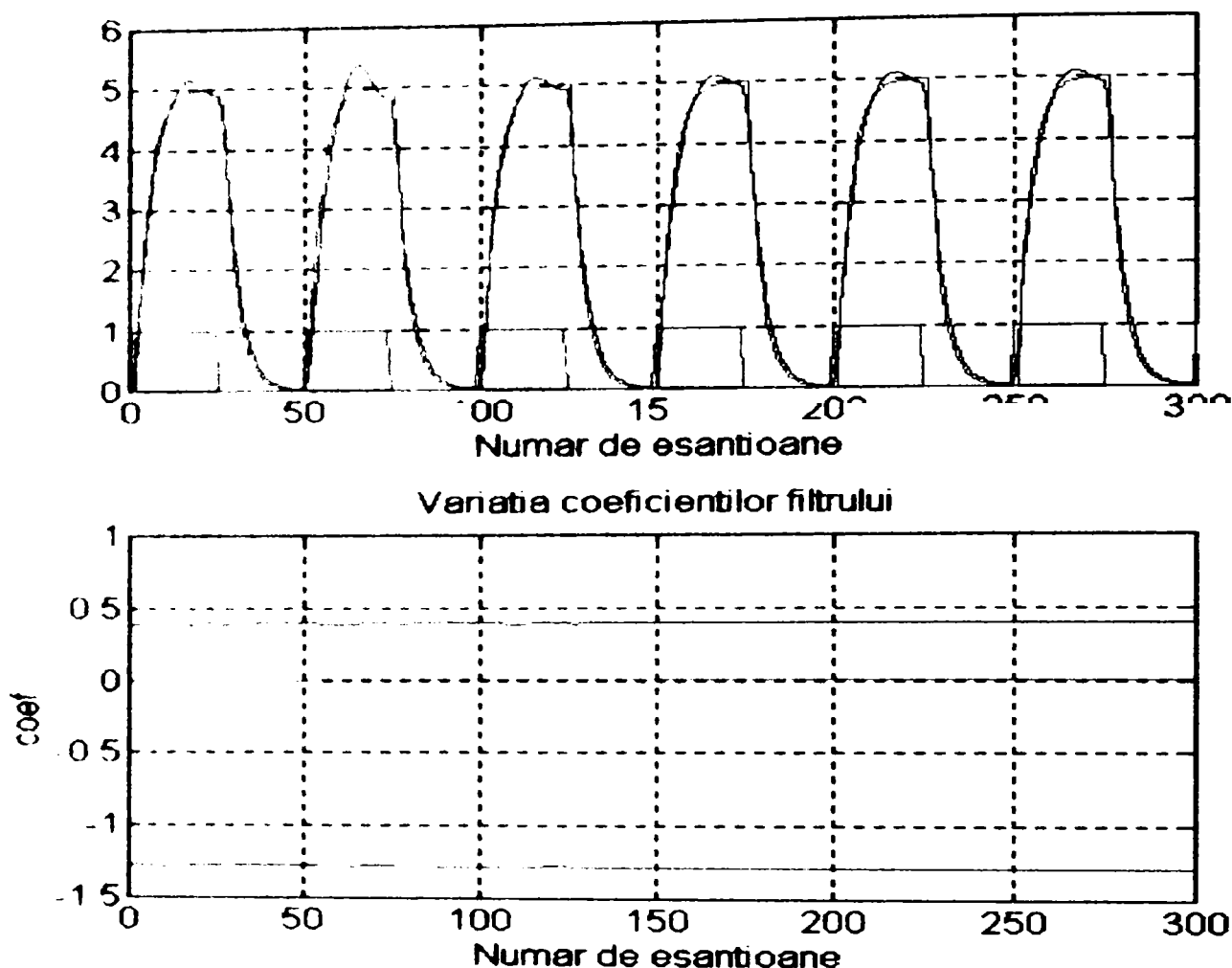


Fig. 5.16. Variația semnalului de test, de ieșire a sistemului necunoscut și a filtrului adaptiv; forma de variație a coeficienților filtrului adaptiv.

De asemenea autorul a verificat experimental faptul că, odată cu creșterea numărului de eșantioane, valorile coeficienților filtrului adaptiv rămân constante. Aceasta rezultă din figura 5.17 în care s-au reprezentat aceleași semnale ca și în figura 5.16, dar reprezentarea s-a făcut pentru un număr mai mare de eșantioane.

5.3. Reglatoare adaptive

5.3.1. Alegerea și acordarea reglatoarelor pentru procese lente

Procesele lente sunt caracterizate prin modele aproximative având constante de timp mai mari de 10 s. cel mai adesea conținând și timp mort. Alegerea tipului de regulator se face pe baza unor criterii care țin seama de caracteristicile procesului și de performanțele impuse [46].

Pentru procese cu o constantă de timp medie și un timp mort redus, dacă amplitudinea perturbațiilor este mică sau medie și frecvența acestora este redusă, se recomandă utilizarea unui regulator de tip bipozițional sau de tip P. Dacă frecvența perturbațiilor este mare și amplitudinea având diverse valori, se recomandă utilizarea unui regulator PI.

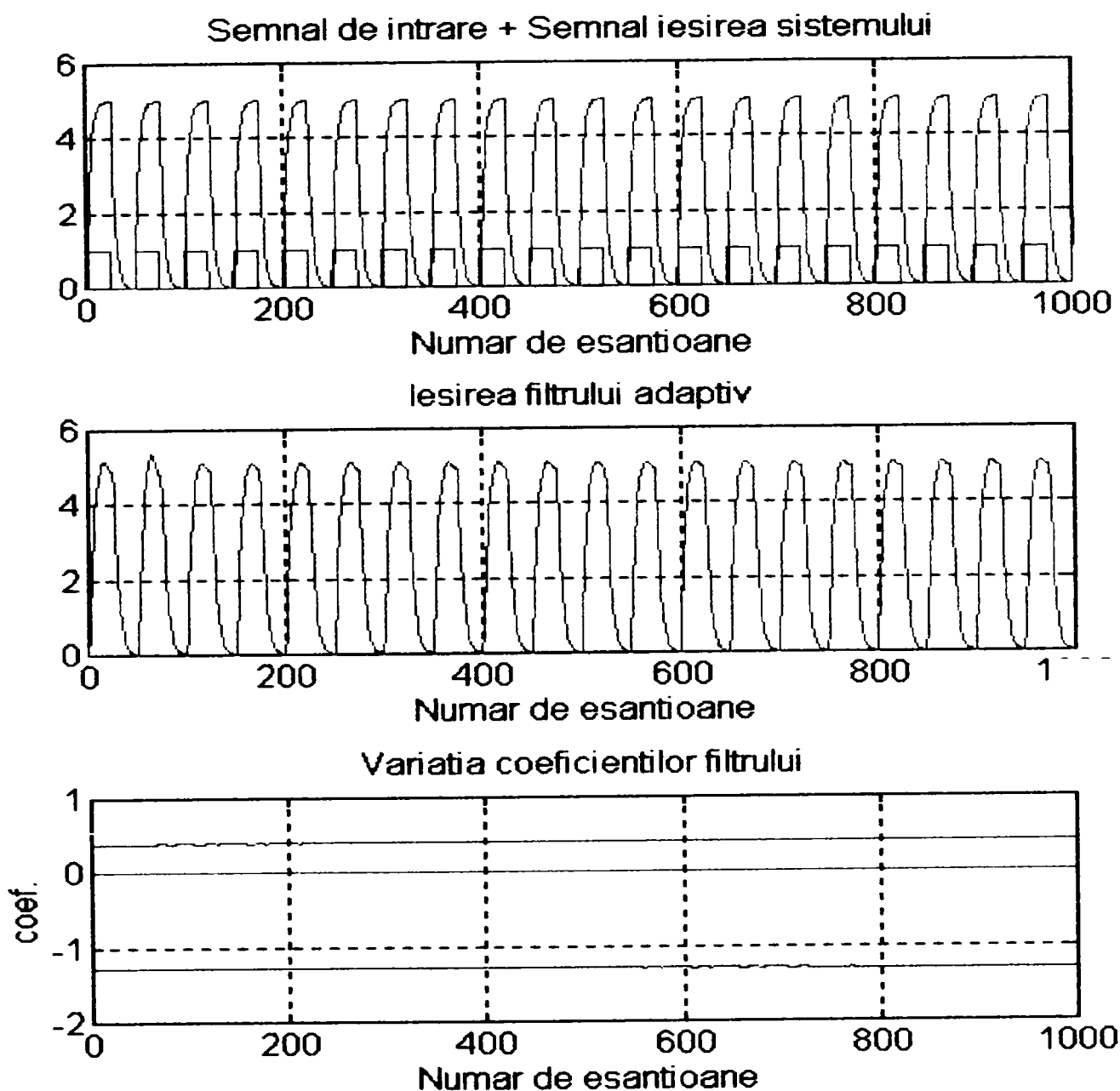


Fig. 5.17. Variația semnalului de test, de ieșire a sistemului necunoscut și a filtrului adaptiv; forma de variație a coeficienților filtrului adaptiv.

Alegerea și acordarea reguletoarelor pentru procese cu timp mort reprezintă una din problemele cele mai dificile în practica reglării automate datorită atât dificultăților la determinarea cu precizie a timpului mort ce caracterizează procesul cât și influenței nefavorabile a timpului mort asupra comportării tranzitorii a unui sistem de reglare automată. Criteriile care pot fi utilizate pentru acordarea reguletoarelor cu timp mort sunt:

1. Criterii bazate pe metoda limitei de stabilitate;
2. Criterii bazate pe rezultate ale identificării;
3. Criterii experimentale, având procesul în funcțiune.

Deoarece metoda de reglare adaptivă a temperaturii propusă de autor are la bază identificarea parametrilor procesului, se vor prezenta doar criteriile bazate pe rezultatele identificării parametrilor procesului.

În [46] s-au prezentat criteriile de alegere a parametrilor reguletoarelor bazate pe rezultatele identificării.

1. Relațiile Ziegler - Nichols :

a. Pentru regulator P

$$K_{R \text{ opt}} = \frac{T}{K \cdot \tau} ; \quad (5.13.a)$$

b. Pentru regulator PI

$$K_{R \text{ opt}} = \frac{0,9T}{K \cdot \tau} , \quad (5.13.b)$$

$$T_{i \text{ opt}} = 3,3\tau$$

2. Relațiile Oppelt :

a. Pentru reguloare P

$$K_{R \text{ opt}} = \frac{T}{K \cdot \tau} ; \quad (5.14.a)$$

b. Pentru reguloare PI

$$K_{R \text{ opt}} = \frac{0,8T}{K \cdot \tau} , \quad (5.14.b)$$

$$T_{i \text{ opt}} = 3\tau .$$

3. Relațiile Kopelovici :

Pe baza unor cercetări experimentale, au fost stabilite relații pentru calculul parametrilor optimi de acord, asigurând un răspuns tranzitoriu aperiodic cu durata minimă și un răspuns cu suprareglaj maxim $\sigma = 20\%$. În tabelul 5.9 sunt prezentate aceste relații de calcul.

Tabelul 5.9. Valorile parametrilor de acord după Kopelovici.

Tipul Regulatorului	Răspuns aperiodic cu durată minimă	Răspuns oscilant cu $\sigma = 20\%$
P	$K_{R \text{ opt}} = \frac{0,3 \cdot T}{K \cdot \tau}$	$K_{R \text{ opt}} = \frac{0,7 \cdot T}{K \cdot \tau}$
PI	$K_{R \text{ opt}} = \frac{0,6 \cdot T}{K \cdot \tau}$ $T_{i \text{ opt}} = 0,8\tau + 0,5 T$	$K_{R \text{ opt}} = \frac{0,7 \cdot T}{K \cdot \tau}$ $T_{i \text{ opt}} = \tau + 0,3 T$

4. Relațiile Chien, Hrones, Reswicz:

Pe baza unor încercări pe modele sunt prezentate în tabelele 5.10 și 5.11 relațiile de calcul ale parametrilor optimi de acord pentru o comportare optimă la variații treaptă a intrării și la variația perturbației. S-au prezentat valorile parametrilor reguletoarelor pentru răspunsuri tranzitorii aperiodice cu durată minimă și răspuns cu $\sigma = 20\%$ și durată minimă, atât la variația treaptă a mărimii de intrare, cât și pentru comportarea optimă la perturbații.

Tabelul 5.10. Valorile optime ale parametrilor de acord la variația treaptă a intrării.

Tipul Regulatorului	Răspuns aperiodic și durată a răspunsului minimă	Răspuns oscilant cu $\sigma = 20\%$ și durată minimă
P	$K_{R \text{ opt}} = \frac{0,3 \cdot T}{K \cdot \tau}$	$K_{R \text{ opt}} = \frac{0,7 \cdot T}{K \cdot \tau}$
PI	$K_{R \text{ opt}} = \frac{0,35 \cdot T}{K \cdot \tau}$ $T_{i \text{ opt}} = 1,2 \tau$	$K_{R \text{ opt}} = \frac{0,6 \cdot T}{K \cdot \tau}$ $T_{i \text{ opt}} = \tau$

Tabelul 5.11. Valorile optime ale parametrilor de acord la variația perturbațiilor.

Tipul Regulatorului	Răspuns aperiodic și durată a răspunsului minimă	Răspuns oscilant amortizat cu $\sigma = 20\%$ și durată minimă
P	$K_{R \text{ opt}} = \frac{0,3 \cdot T}{K \cdot \tau}$	$K_{R \text{ opt}} = \frac{0,7 \cdot T}{K \cdot \tau}$
PI	$K_{R \text{ opt}} = \frac{0,6 \cdot T}{K \cdot \tau}$ $T_{i \text{ opt}} = 4 \tau$	$K_{R \text{ opt}} = \frac{0,7 \cdot T}{K \cdot \tau}$ $T_{i \text{ opt}} = 2,3 \tau$

5.3.2. Implementarea reguletoarelor adaptive

Așa cum s-a prezentat în paragraful anterior, pentru procese cu o constantă de timp și timp mort, este recomandabilă alegerea reguletoarelor de tip P sau PI.

Funcțiile de transfer a sistemului analogic se deduc pornind de la relația dintre mărimea de la ieșirea reguletorului și cea de la intrarea sa.

1. În cazul reguletoarelor de tip P, ecuația ieșirii este dată de relația

$$y_R(t) = K_R \cdot x(t),$$

iar funcția de transfer a sistemului analogic de relația

$$H_R(s) = K_R.$$

2. În cazul reguletoarelor de tip PI, ecuația ieșirii este dată de relația

$$y_R(t) = K_R \cdot x(t) + \frac{K_R}{T_i} \int_0^t x(\tau) d\tau,$$

iar funcția de transfer a sistemului analogic de relația

$$H_R(s) = \frac{K_R}{T_i} \cdot \frac{1 + sT_i}{s}.$$

Cunoscând setul de coeficienți ai filtrului adaptiv la iterația n, se pot calcula parametrii procesului, T, τ și K. În funcție de tipul de reguletor adaptiv ales și de criteriul de alegere al acestuia și cunoscând valorile parametrilor procesului, se pot determina valorile parametrilor reguletorului, determinând astfel funcția de transfer analogică a reguletorului.

Funcția sistem a sistemului discret, așa cum s-a prezentat anterior, se poate obține utilizând una dintre metodele de echivalare a sistemului analogic cu un sistem discret, metoda dreptunghiurilor sau metoda trapezelor. Indiferent de metoda de echivalare utilizată și de tipul de regulator, relația generală de determinare a mărimii de ieșire a sistemului discret este

$$y(n) = b_0 \cdot x(n) + b_1 \cdot x(n-1) + a_1 \cdot y(n-1).$$

În funcție de tipul de regulator și de metoda de echivalare utilizată, în tabelul 5.12, autorul a prezentat valorile coeficienților relației (5.15) în funcție de parametrii regulatorului și de perioada de eșantionare.

Tabelul 5.12. Valorile coeficienților regulatorului numeric.

Metoda de echivalare	Tip regulator	b_0	b_1	a_1
Metoda dreptunghiurilor	P	K_R	0	0
	PI	$K_R \left(1 + \frac{T_c}{T_i} \right)$	$-K_R$	1
Metoda trapezelor	P	K_R	0	0
	PI	$\frac{K_R}{2T_i} (T_c + 2T_i)$	$\frac{K_R}{2T_i} (T_c - 2T_i)$	1

5.4. Sistemul de reglare adaptivă al temperaturii la cuptorul cu rezistență electrică

5.4.1. Structura sistemului de reglare a temperaturii

În figura 5.18 este prezentată structura sistemului de reglare a temperaturii la cuptorul cu rezistență electrică, sistem de reglare ce are în componență următoarele elemente:

1. Cuptorul cu rezistență electrică, acesta reprezentând sistemul a cărui mărime de ieșire, temperatura, se dorește a fi controlată prin modificarea puterii disipate pe rezistența electrică a acestuia. Mărimea de intrare a acestui sistem este puterea disipată pe rezistența electrică.

2. Redresorul comandat în tensiune (RCT), cu rolul de a permite reglarea valorii efective a tensiunii de la ieșirea sa în funcție de nivelul tensiunii continue aplicate la intrare. Mărimea de ieșire a redresorului comandat în tensiune este valoarea efectivă a tensiunii aplicată la bornele rezistenței electrice și deci, în funcție de valoarea rezistenței, puterea debitată pe aceasta.

3. Traductorul de temperatură (T), utilizat pentru obținerea la ieșire a unei tensiuni proporțională cu temperatura. S-a utilizat un traductor de temperatură cromel-alumel cu temperatura maximă măsurabilă de 1200°C și o tensiune de ieșire maximă de 48mV. Acest nivel mic de tensiune furnizat de traductorul de temperatură a necesitat utilizarea unui amplificator de tensiune cu amplificare mare și care să funcționeze în curent continuu.

4. **Amplificatorul de tensiune (A)**, utilizat pentru amplificarea tensiunii furnizate la ieșirea traductorului de temperatură în scopul obținerii la ieșire a unei tensiuni cu amplitudinea de 10V, tensiune care să corespundă domeniului de măsurare al sistemului de achiziție de date.

5. **Calculatorul IBM-PC** având în componență o placă de achiziție de date de tip ADA-3100, utilizat în scopul implementării algoritmului de reglare numerică a temperaturii.

6. **Placa de achiziție de date ADA-3100** utilizată atât în scopul conversiei analog-numerică a tensiunii de la ieșirea amplificatorului, cât și în scopul conversiei numeric-analogice pentru furnizarea la ieșire a unei tensiuni care să comande redresorul comandat în tensiune.

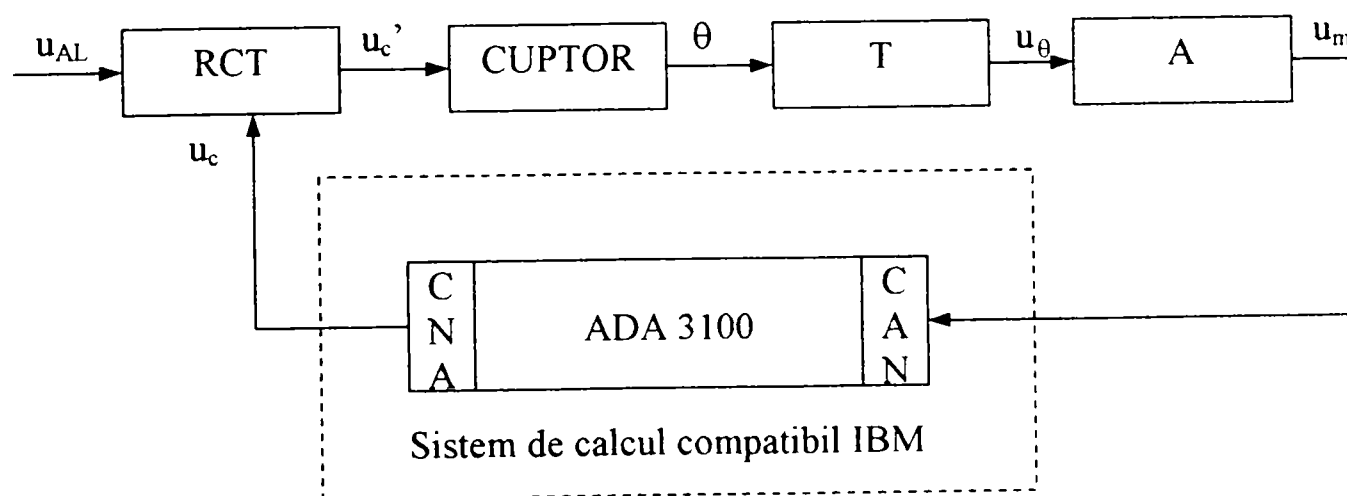


Fig. 5.18. Structura sistemului de reglare adaptivă a temperaturii pentru cuptorul cu rezistență electrică.

5.4.2. Caracteristicile plăcii de achiziție de date ADA-3100

Realizarea conversiei analog-numerică și numeric-analogice necesare, așa cum s-a prezentat anterior, în vederea măsurării temperaturii din interiorul cuptorului, respectiv pentru reglarea valorii efective a tensiunii pe rezistența de sarcină a fost făcută cu ajutorul plăcii de achiziție de date ADA-3100 ale cărei caracteristici principale sunt:

- existența a 8 canale analogice de intrare care pot fi utilizate în mod diferențial sau cu tensiunea de intrare măsurată față de masă;
- posibilitatea realizării conversiei analog-numerică pe 12 biți, la o frecvență de eșantionare de 12 KHz pe un canal;
- existența a 2 canale analogice de ieșire, semnalele de ieșire provenind de la ieșirile a două convertoare numeric-analogice pe 12 biți fiecare;
- existența a 8 linii de intrare numerice și a 8 linii de ieșire numerice ;
- domeniul tensiunilor de intrare $\pm 5, \pm 10$, sau $0 - 10$ V , selectabil prin programare;
- domeniul tensiunilor de ieșire $\pm 5, \pm 10, 0 - 5$ sau $0 - 10$ V , selectabil prin programare.
- posibilitatea alegerii unei amplificări a semnalului de la intrare de 1, 2, 4, 8, 16 sau reglabilă, înaintea realizării operației de conversie analog-numerică;
- existența unei memorii FIFO cu capacitate de 8 Kocteți.

5.4.3. Măsurarea parametrilor cuptorului cu rezistență electrică prin metoda de identificare on-line

Așa cum s-a prezentat anterior, în situația utilizării unui filtru adaptiv la identificarea parametrilor proceselor lente este necesar ca utilizând o altă metodă să se determine cu aproximație valorile parametrilor sistemului și utilizând aceste valori să se determine setul de valori inițiale ale coeficienților filtrului.

Autorul a determinat utilizând metoda de identificare on-line prezentată în paragraful 5.1 parametrii modelului încălzirii cuptorului cu rezistență electrică. Impulsul de test a constat în aplicarea pe durata impulsului de test a tensiunii redresate de valoare maximă la bornele rezistenței care produce încălzirea cuptorului, interval de timp în care temperatura în interiorul cuptorului a crescut. După acest interval de timp tensiunea de alimentare a fost deconectată, temperatura în interiorul cuptorului scăzând. Durata impulsului de test aplicată a fost de 20 minute, durata de integrare a fost de 40 minute, iar perioada de eșantionare de 0,2 secunde.

Pentru determinarea parametrilor modelului încălzirii cuptorului autorul a efectuat un număr de 10 măsurători în condiții inițiale nule, rezultatele măsurătorilor fiind prezentate în Tabelul 5.13.

Forma de variație a temperaturii pentru una dintre măsurători este prezentată în figura 5.20.

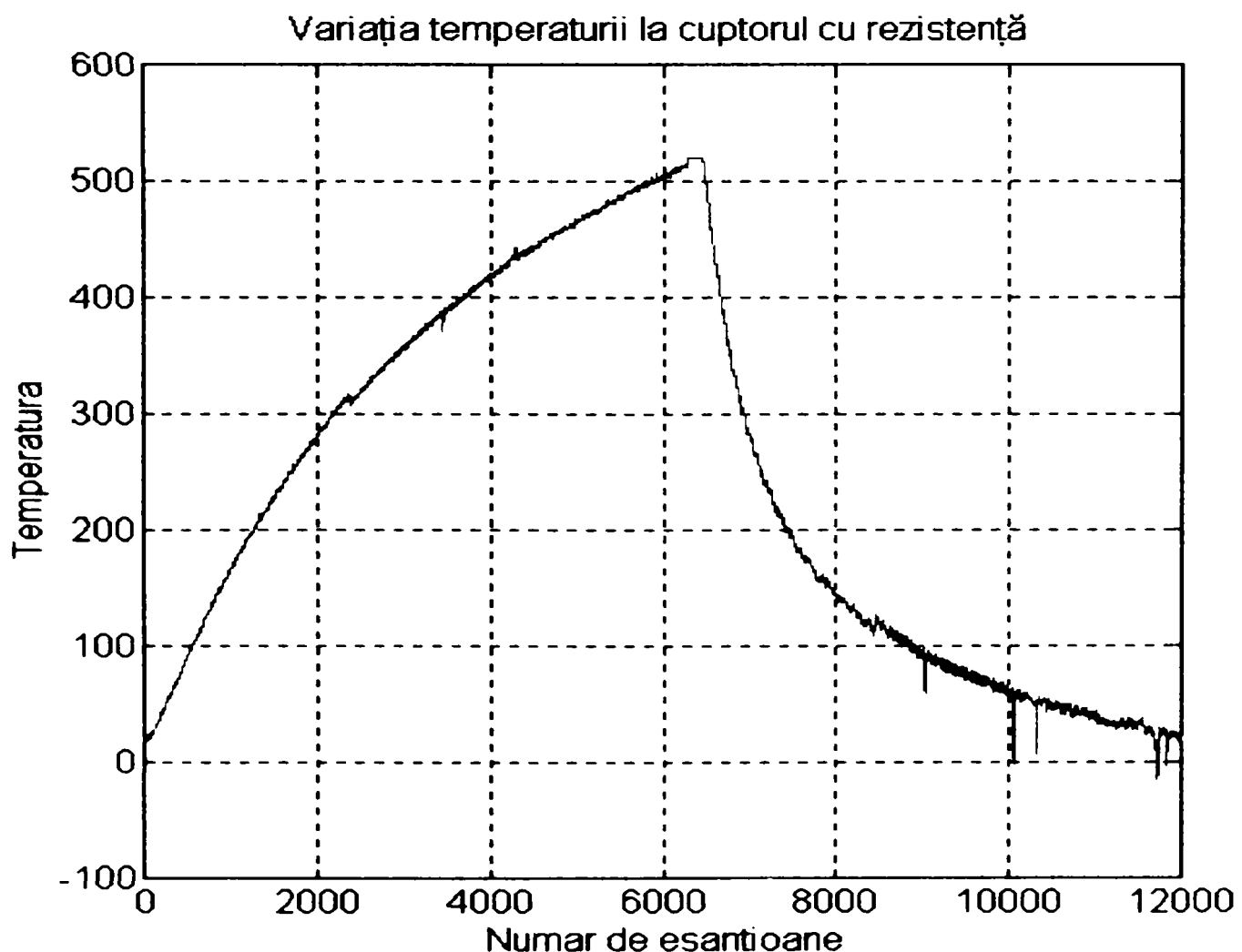


Fig. 5.20. Forma de variație a temperaturii utilizând metoda de identificare on-line.

Tabelul 5.13. Valorile parametrilor modelului încălzirii cuptorului măsurate prin metoda de identificare on-line.

Nr.crt.	$K_{m\grave{a}s}$	$T_{m\grave{a}s}$ (secunde)	$\tau_{m\grave{a}s}$ (secunde)
1	442,60	274,22	66,21
2	438,56	272,21	64,87
3	450,88	280,34	65,38
4	443,24	275,33	66,86
5	436,22	271,00	67,12
6	445,78	278,22	66,43
7	436,50	270,29	63,26
8	447,87	279,24	65,96
9	440,26	275,34	67,28
10	438,22	273,37	66,32

Pe baza rezultatelor prezentate în Tabelul 5.13, autorul a ales ca valori inițiale ale parametrilor modelului încălzirii cuptorului $K = 440$, $T = 275$ secunde și $\tau = 66$ secunde.

Așa cum s-a prezentat în paragraful 5.2.3., datorită faptului că algoritmul SHARF prezintă oscilații ale valorilor coeficienților filtrului adaptiv mai mici, indiferent de forma de implementare sau de metoda de echivalare a filtrului analogic cu un filtru numeric, autorul a utilizat acest algoritm în procesul de identificare adaptivă a parametrilor modelului încălzirii cuptorului.

De asemenea, deoarece forma de implementare lattice prezintă o viteză de convergență mai mare decât forma de implementare directă, autorul a utilizat forma de implementare lattice a filtrului adaptiv.

Deoarece numărul de coeficienți ai filtrului adaptiv în varianta utilizării aproximației Padé (1+1) este mai mic în situația utilizării metodei dreptunghiurilor decât în situația utilizării metodei trapezelor, autorul a utilizat metoda aproximării integralei prin metoda dreptunghiurilor.

Cu valorile inițiale alese pentru parametrii modelului încălzirii cuptorului s-a testat pentru mai multe valori ale perioadei de eșantionare convergența coeficienților filtrului adaptiv, autorul ajungând la concluzia că perioada de eșantionare optimă este de 30 secunde înțelegând prin aceasta intervalul de timp necesar între două reglări consecutive ale puterii debitate de rezistența electrică. În figura 5.21 este prezentată forma de variație a coeficienților filtrului adaptiv în situația utilizării algoritmului SHARF, implementând filtrul adaptiv în forma lattice și utilizând metoda aproximării integralei prin metoda dreptunghiurilor.

5.4.4. Prezentarea metodei propuse de reglare adaptivă a temperaturii

Metoda de reglare adaptivă a temperaturii utilizează un filtru adaptiv în scopul determinării valorilor parametrilor modelului procesului de încălzire a cuptorului cu rezistență electrică, valori care se pot deduce, așa cum s-a prezentat anterior cunoscând valorile coeficienților filtrului adaptiv.

Cunoscând valorile parametrilor modelului procesului de încălzire a cuptorului cu rezistență electrică și utilizând criteriile de alegere a parametrilor reglatoarelor bazate pe rezultatele identificării prezentate în paragraful 5.3.1 se pot determina parametrii regulatorului.

Implementarea regulatorului adaptiv s-a făcut utilizând relațiile prezentate în paragraful 5.3.2, valorile coeficienților regulatorului numeric fiind prezentate în Tabelul 5.12.

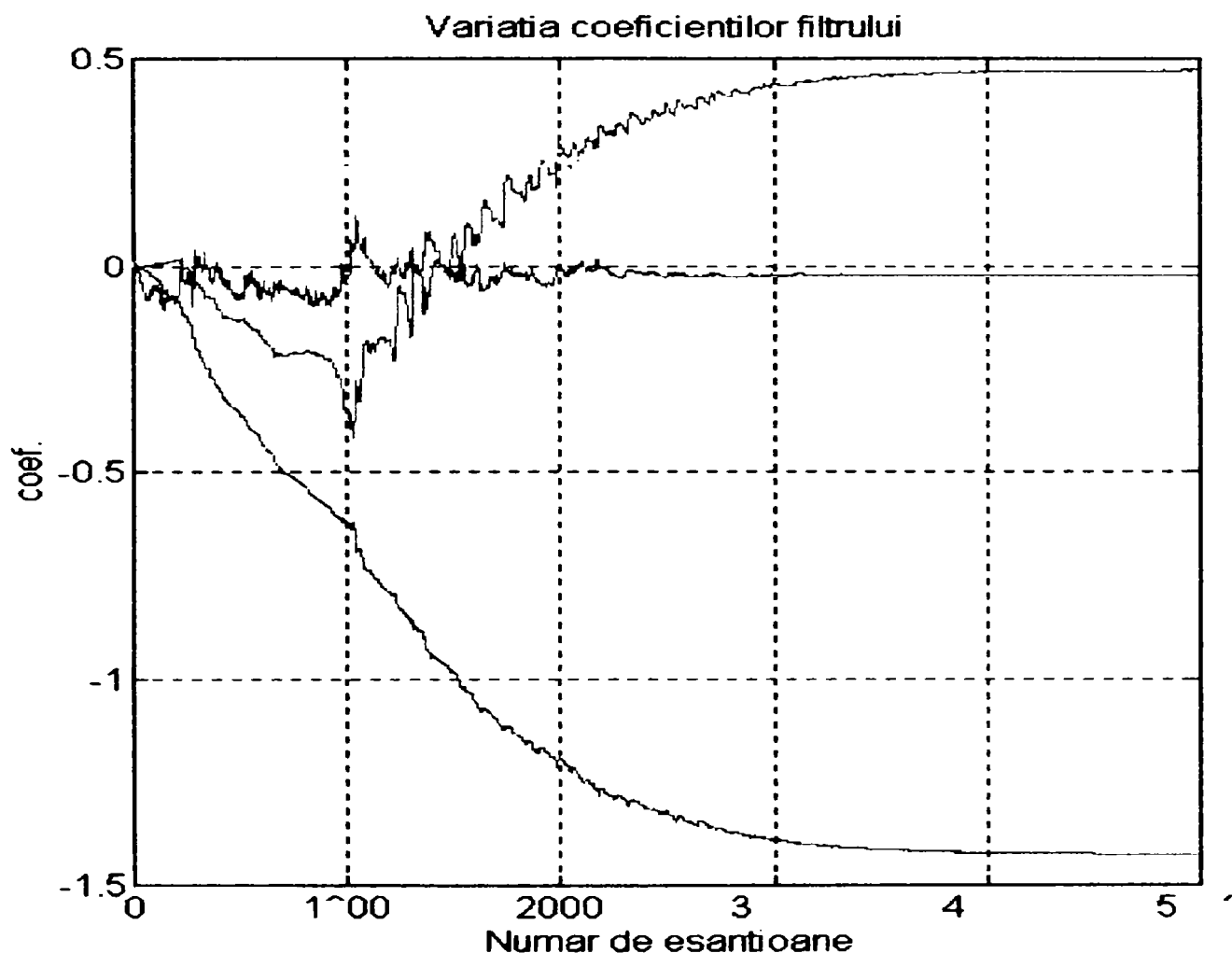


Fig. 5.21. Variația coeficienților filtrului adaptiv utilizând aproximația Padé (1+1), algoritmul SHARF implementat în formă lattice ($\alpha = 0,2$; $T_c = 30$ s).

5.5. Eliminarea zgomotelor cu caracter negaussian

Experimental s-a constatat că în procesul de măsurare al temperaturii cuptorului apar zgomote cu caracter de impuls cu amplitudine mare, datorate funcționării redresorului comandat în tensiune.

Procesul de reglare a temperaturii în cuptorul cu rezistență electrică presupune cunoașterea exactă a valorii momentane a temperaturii, adică necesitatea eliminării influenței impulsurilor de amplitudine mare și durată scurtă.

Din acest motiv, în procesul de măsurare a temperaturii s-a utilizat un filtru MLMS cu lungimea ferestrei de 75 eșantioane pentru care s-a constatat experimental că se poate realiza atât o reglare în timp real a temperaturii cât și o bună rejecție a zgomotelor.

În figura 5.22 este prezentată forma de variație a temperaturii în cuptor utilizând filtrul MLMS.

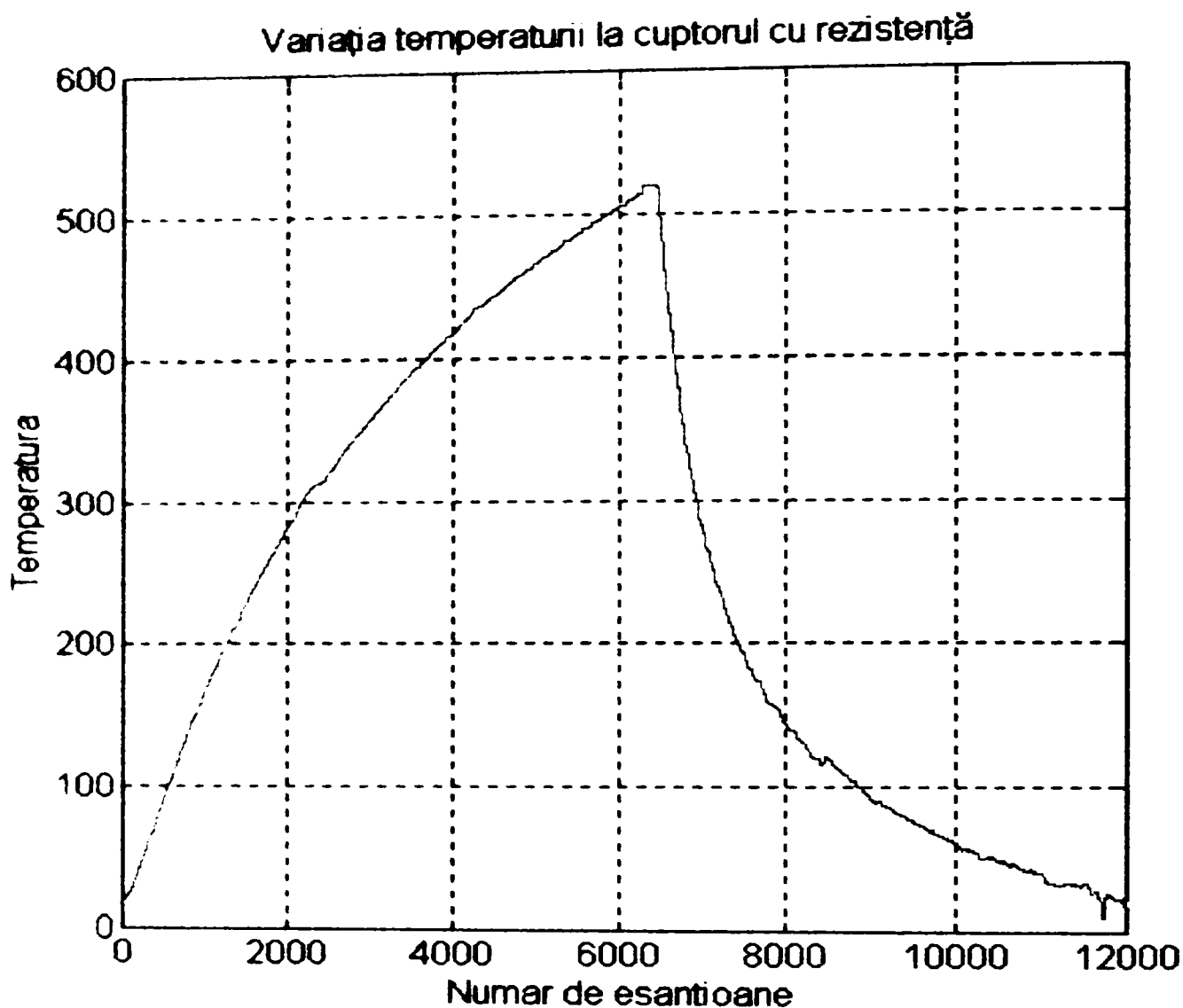


Fig. 5.22. Forma de variație temperaturii în cuptorul cu rezistență utilizând un filtru MLMS cu lungimea ferestrei de 75 eşantioane.

5.6. Rezultate experimentale

În figura 5.23 este prezentată structura sistemului de reglare adaptivă a temperaturii realizat de autor. În această figură s-au folosit notațiile din figura 5.18. Autorul a verificat experimental funcționarea sistemului de reglare adaptivă a temperaturii utilizând atât un regulator adaptiv numeric de tip P cât și de tip PI, parametrii regulatorului fiind aleși pe baza criteriilor de identificare a parametrilor procesului.

În figurile 5.24 – 5.33 sunt prezentate formele de variație a temperaturii în interiorul cuptorului în situația utilizării relațiilor de alegere a parametrilor reglatoarelor prezentate în paragraful 5.3.1.

Reglarea puterii debitate de rezistența electrică s-a făcut la intervale de 30 secunde iar citirea valorii temperaturii din interiorul cuptorului s-a făcut la intervale de 0,5 secunde.

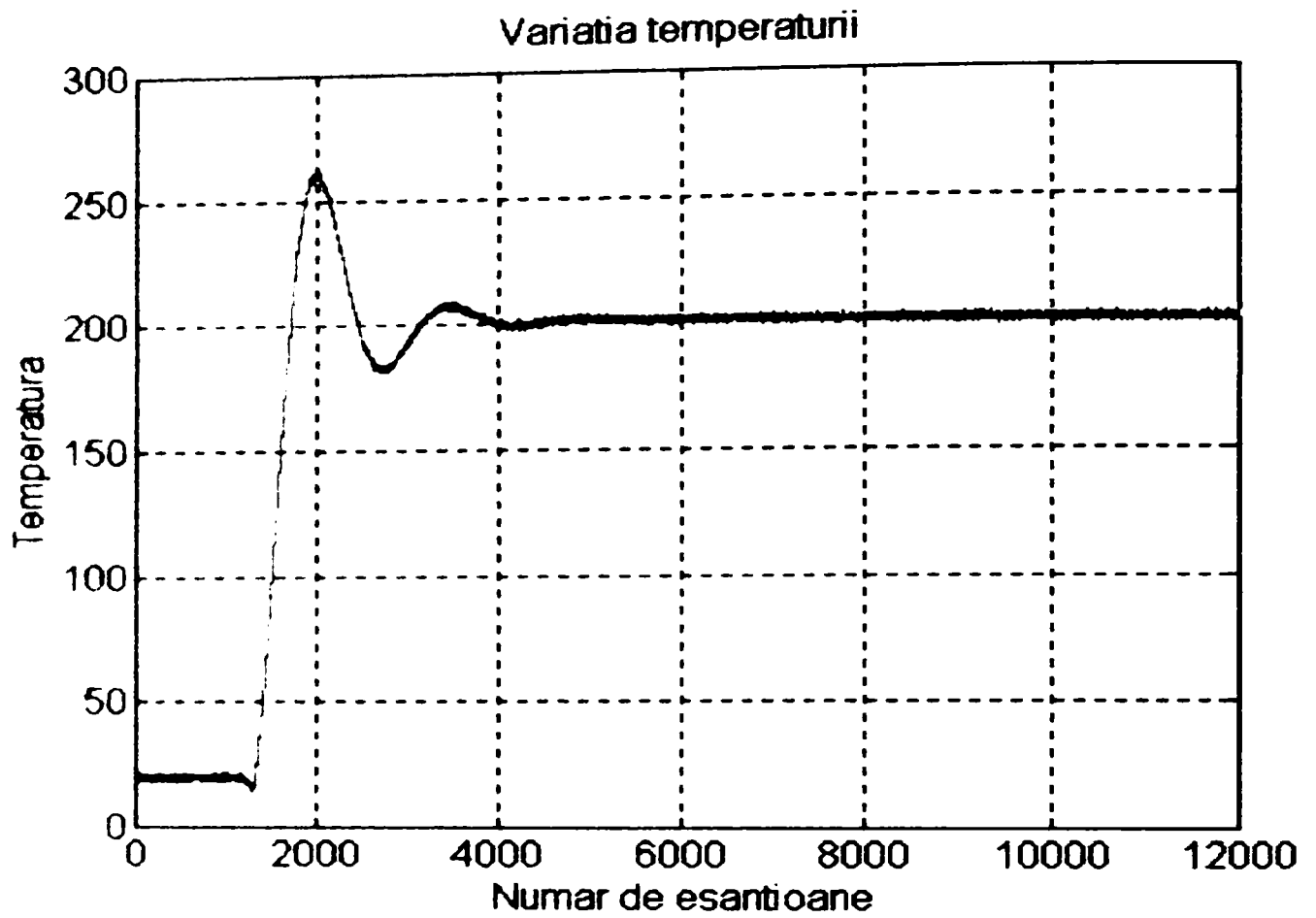


Fig. 5.24. Variația temperaturii în cazul utilizării relațiilor Ziegler-Nichols pentru regulator P.

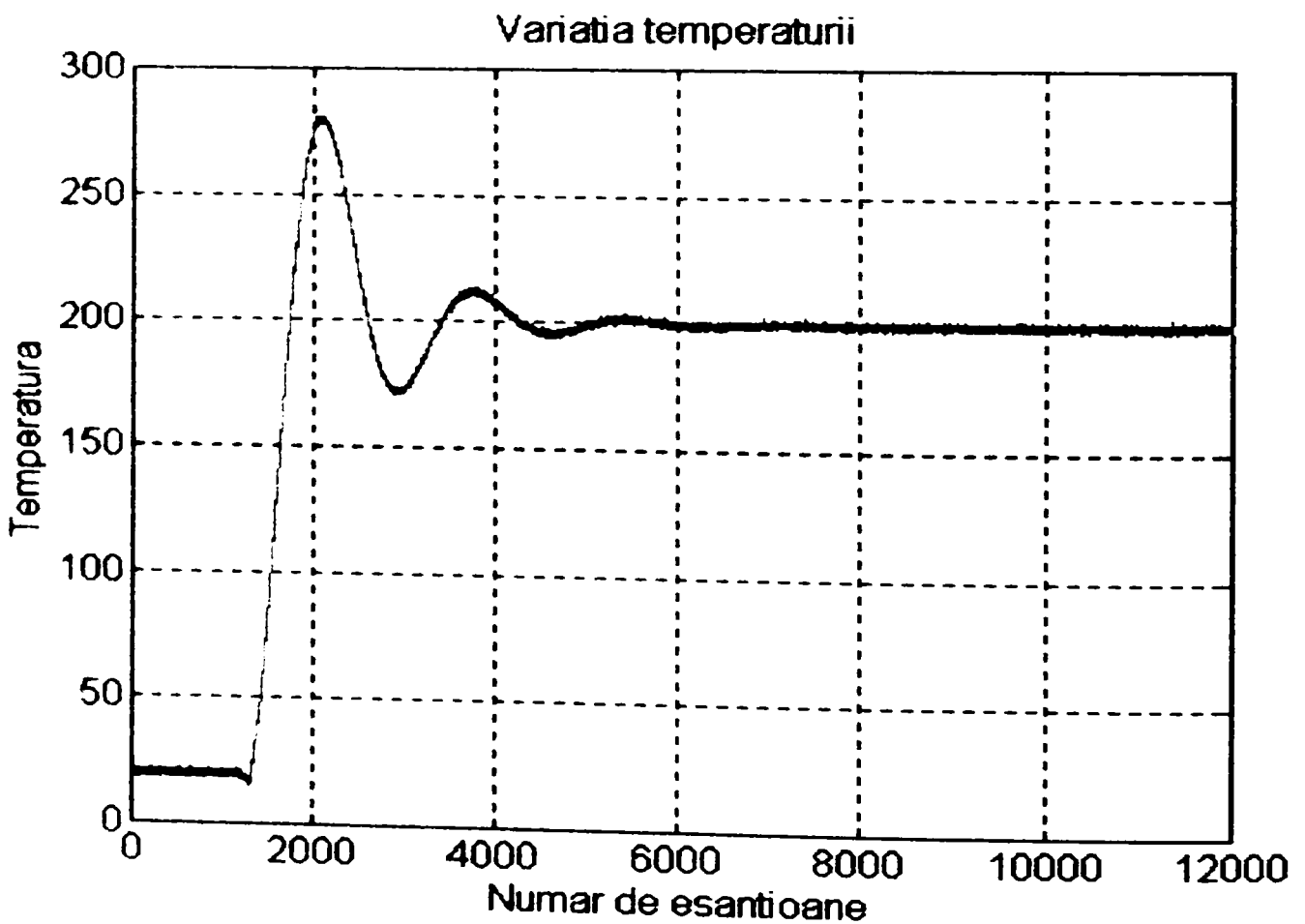


Fig. 5.25. Variația temperaturii în cazul utilizării relațiilor Ziegler-Nichols pentru regulator PI.

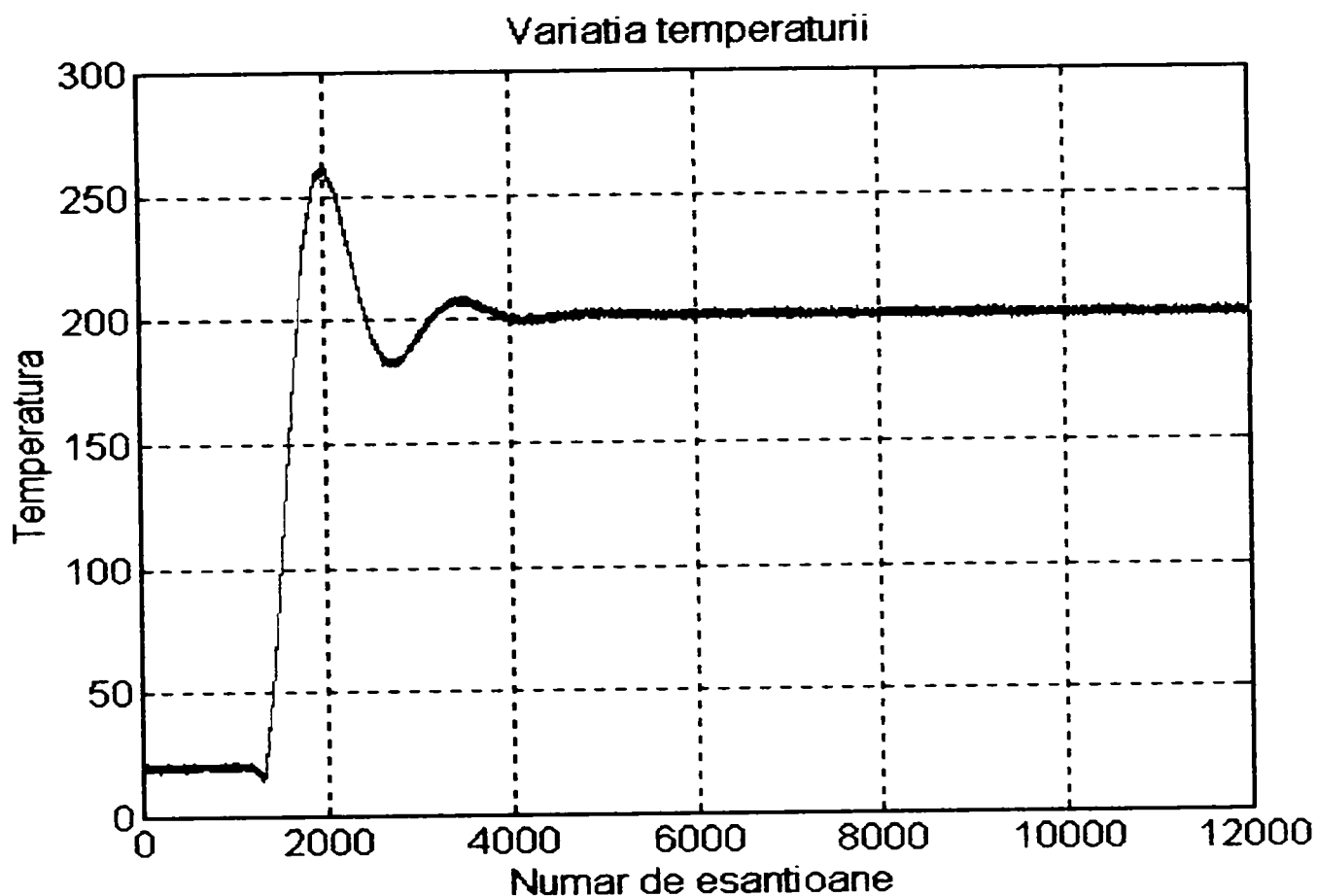


Fig. 5.26. Variația temperaturii în cazul utilizării relațiilor Oppelt pentru regulator P.

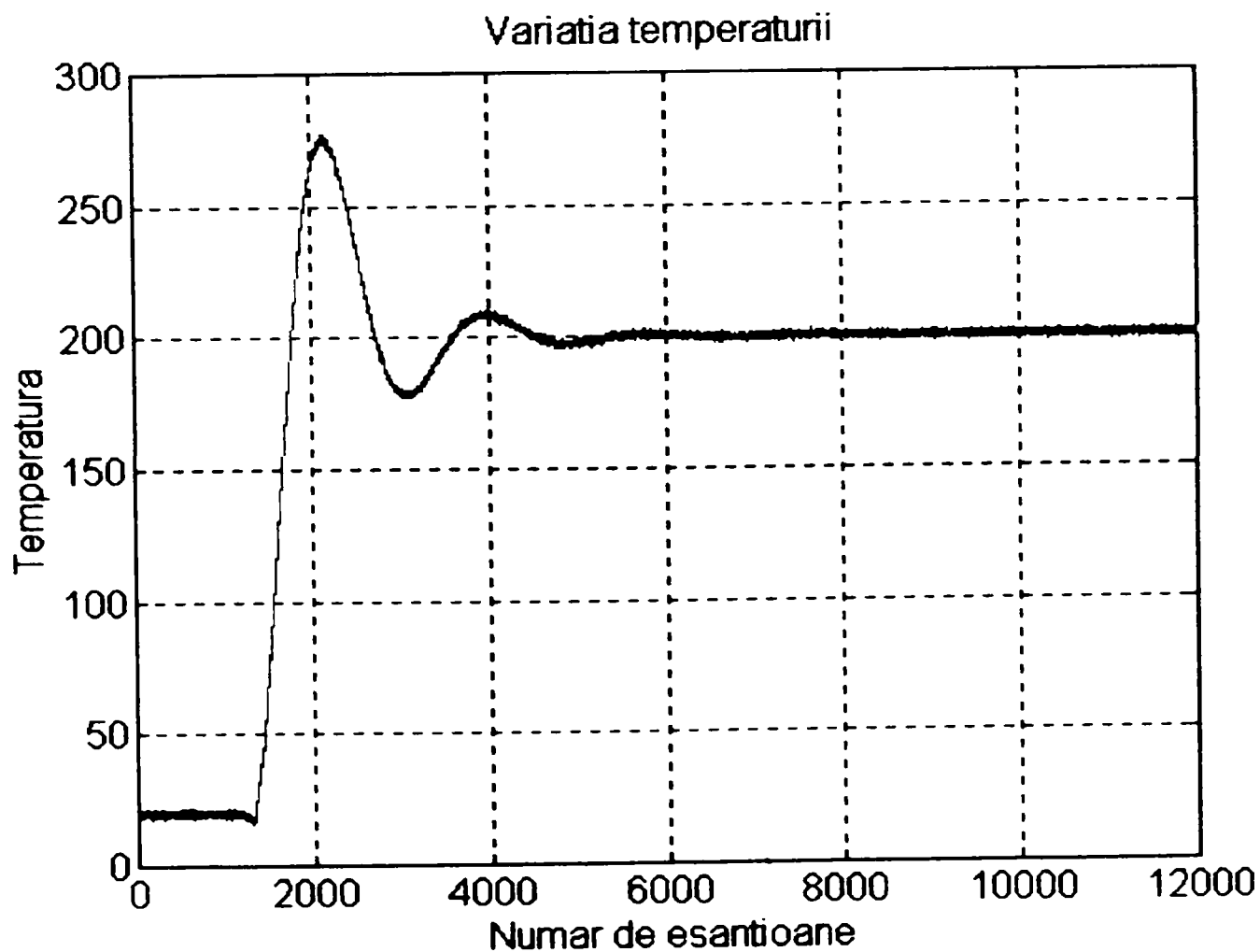


Fig. 5.27. Variația temperaturii în cazul utilizării relațiilor Oppelt pentru regulator PI.

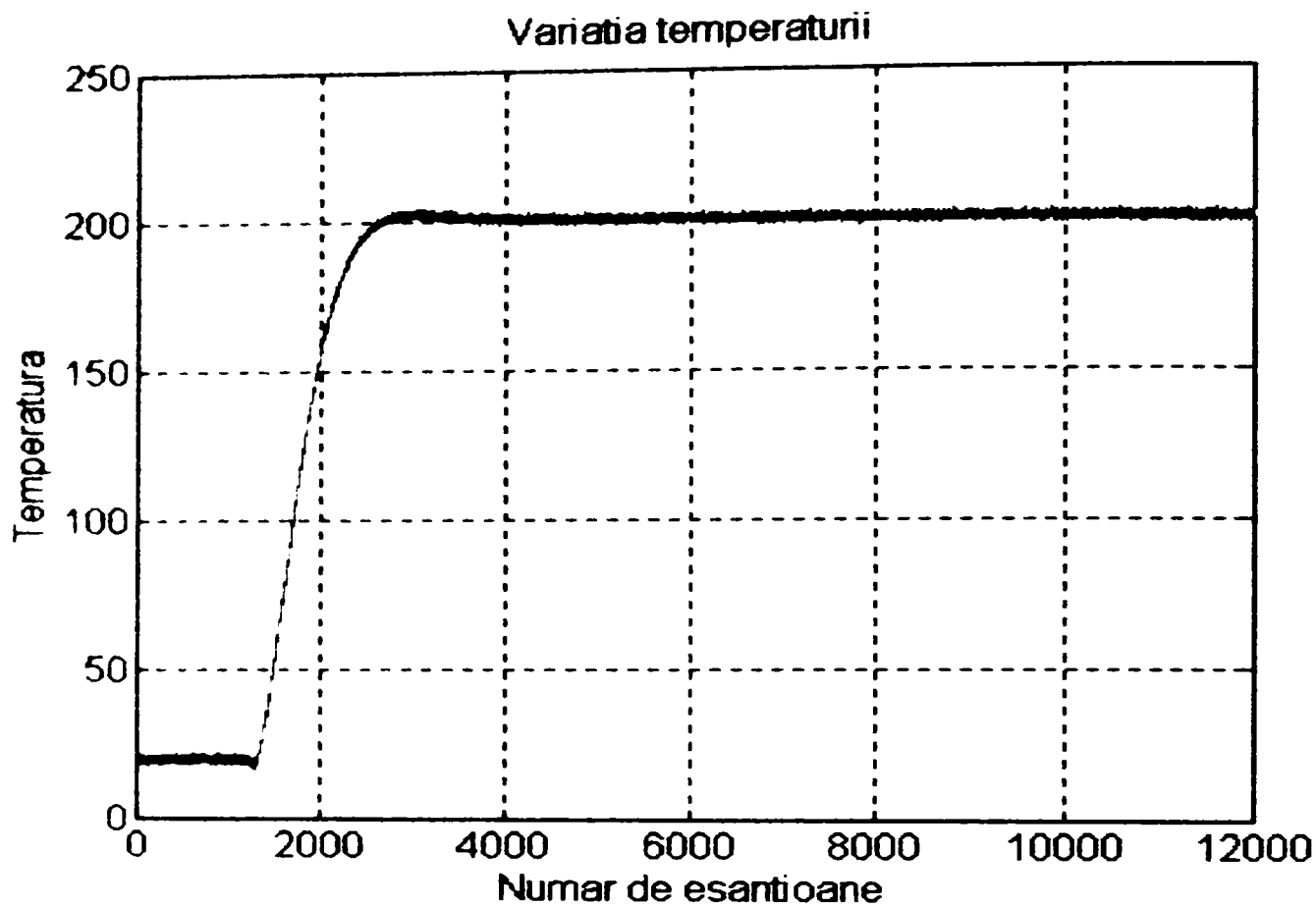


Fig. 5.28. Variația temperaturii în cazul utilizării relațiilor Kopelovici pentru regulator P, răspuns aperiodic cu durata minimă.

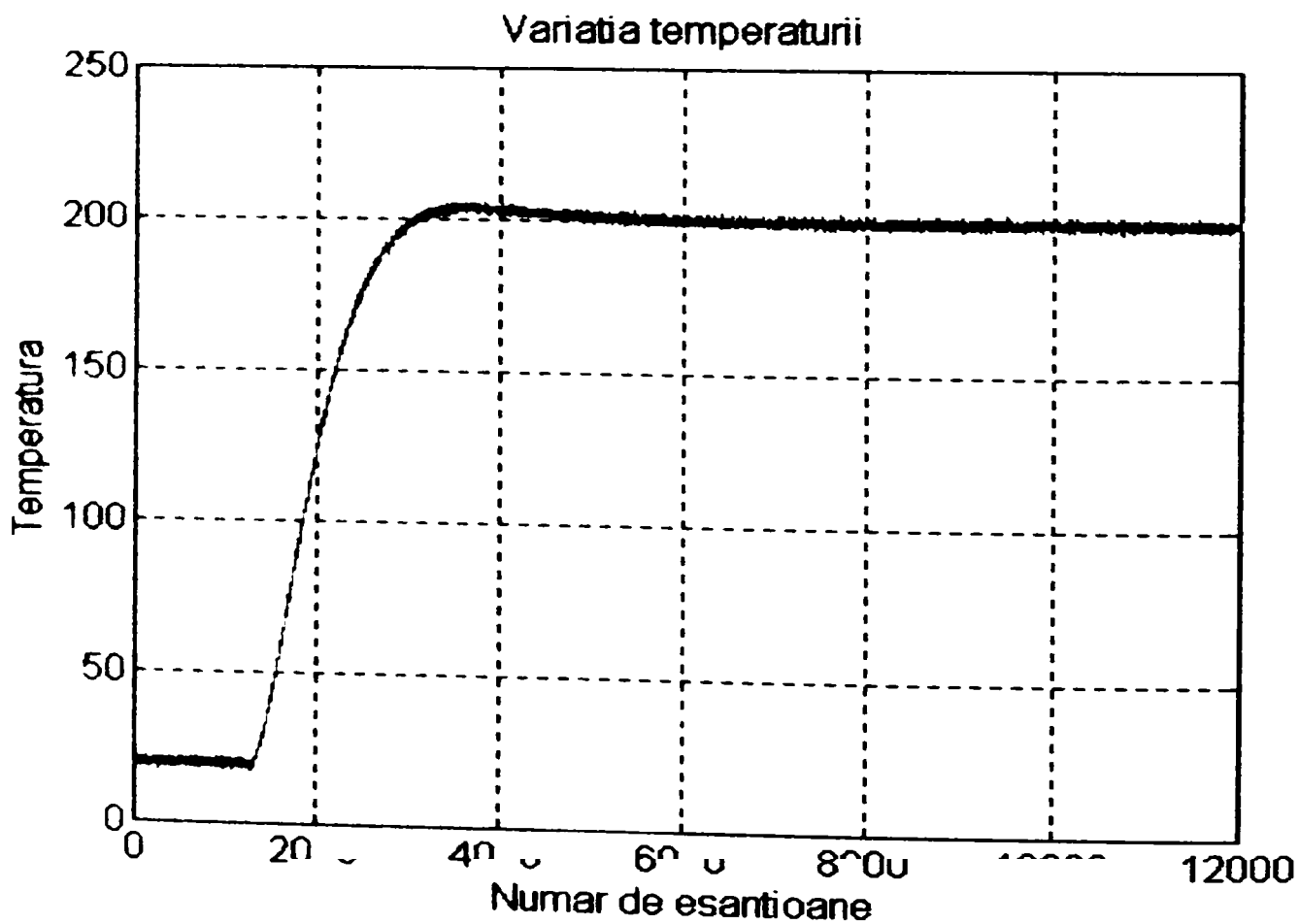


Fig. 5.29. Variația temperaturii în cazul utilizării relațiilor Kopelovici pentru regulator PI, răspuns aperiodic cu durata minimă.

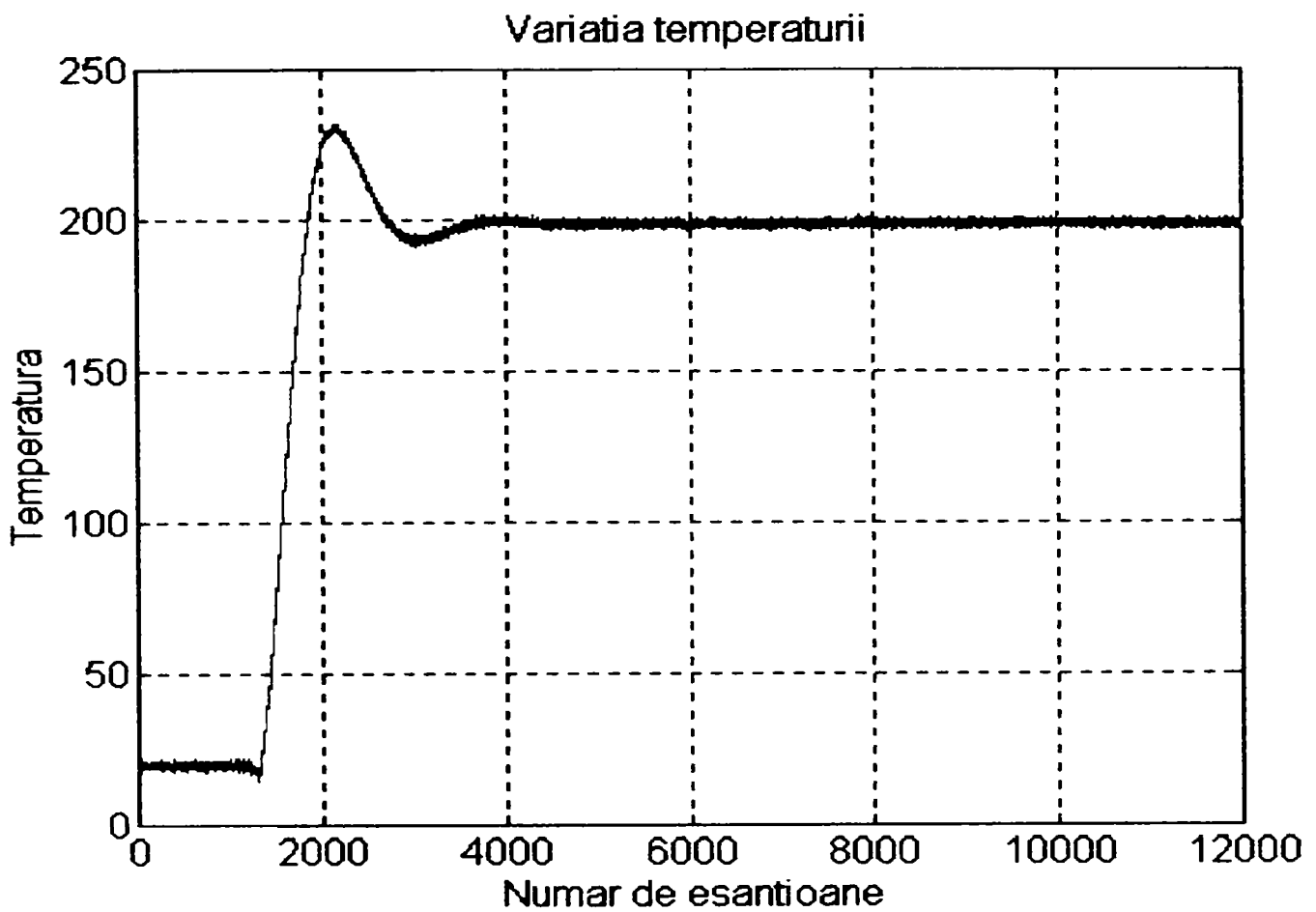


Fig. 5.30. Variația temperaturii în cazul utilizării relațiilor Kopelovici pentru regulator P, răspuns oscilant cu $\sigma = 20\%$.

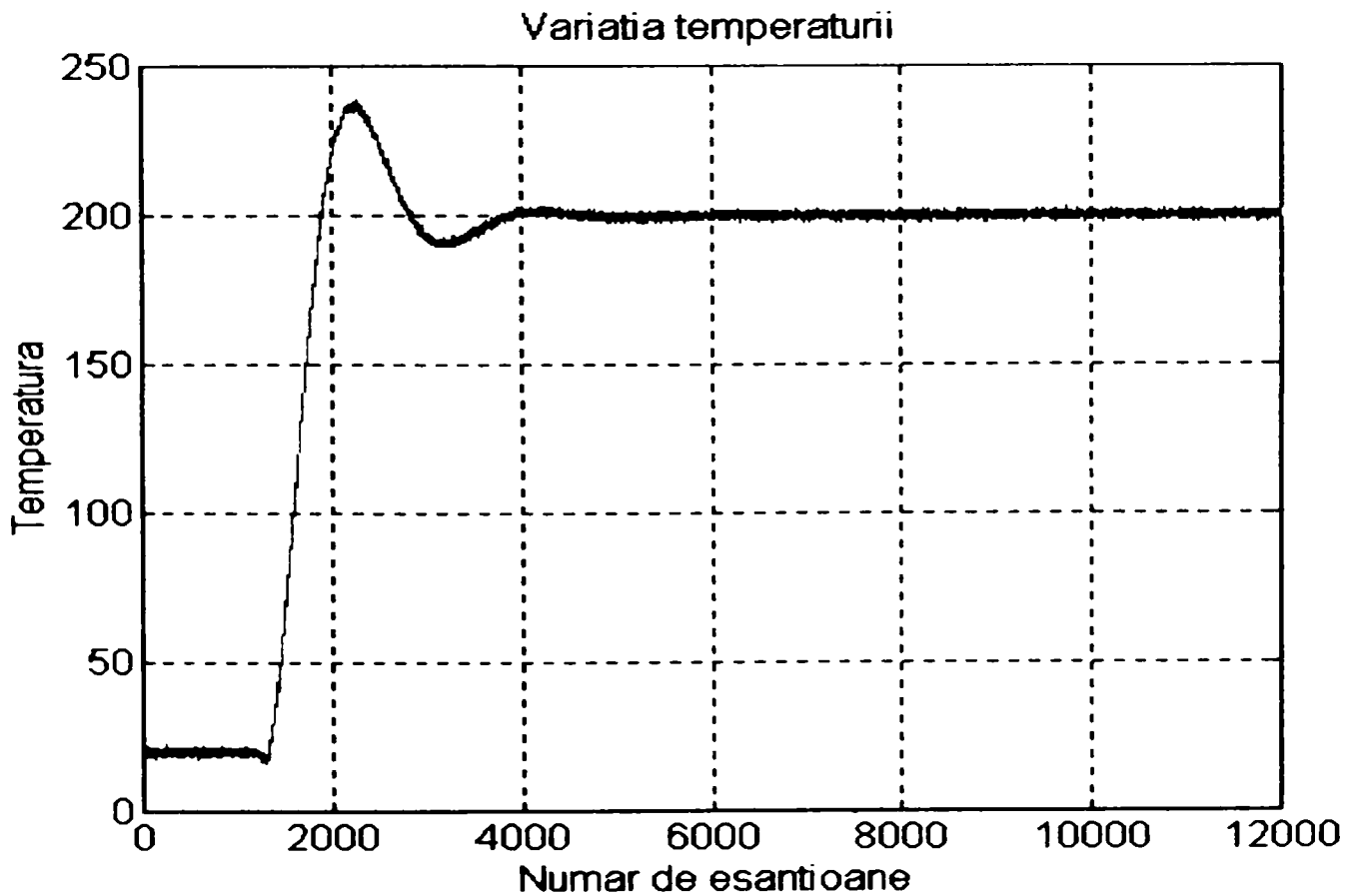


Fig. 5.31. Variația temperaturii în cazul utilizării relațiilor Kopelovici pentru regulator PI, răspuns oscilant cu $\sigma = 20\%$.

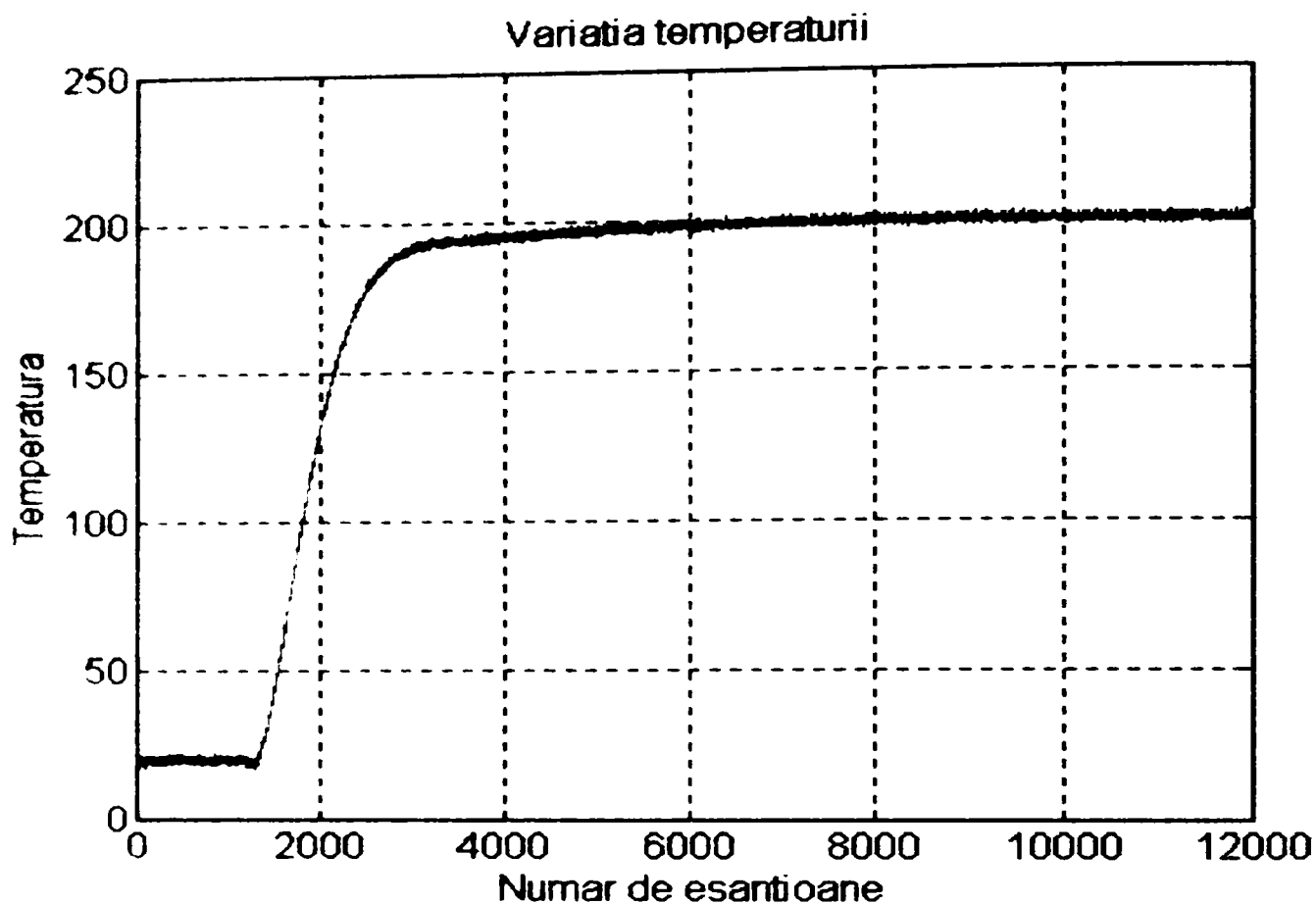


Fig. 5.32. Variația temperaturii în cazul utilizării relațiilor Chien, Hrones, Reswich pentru regulator PI, răspuns aperiodic cu durată minimă.

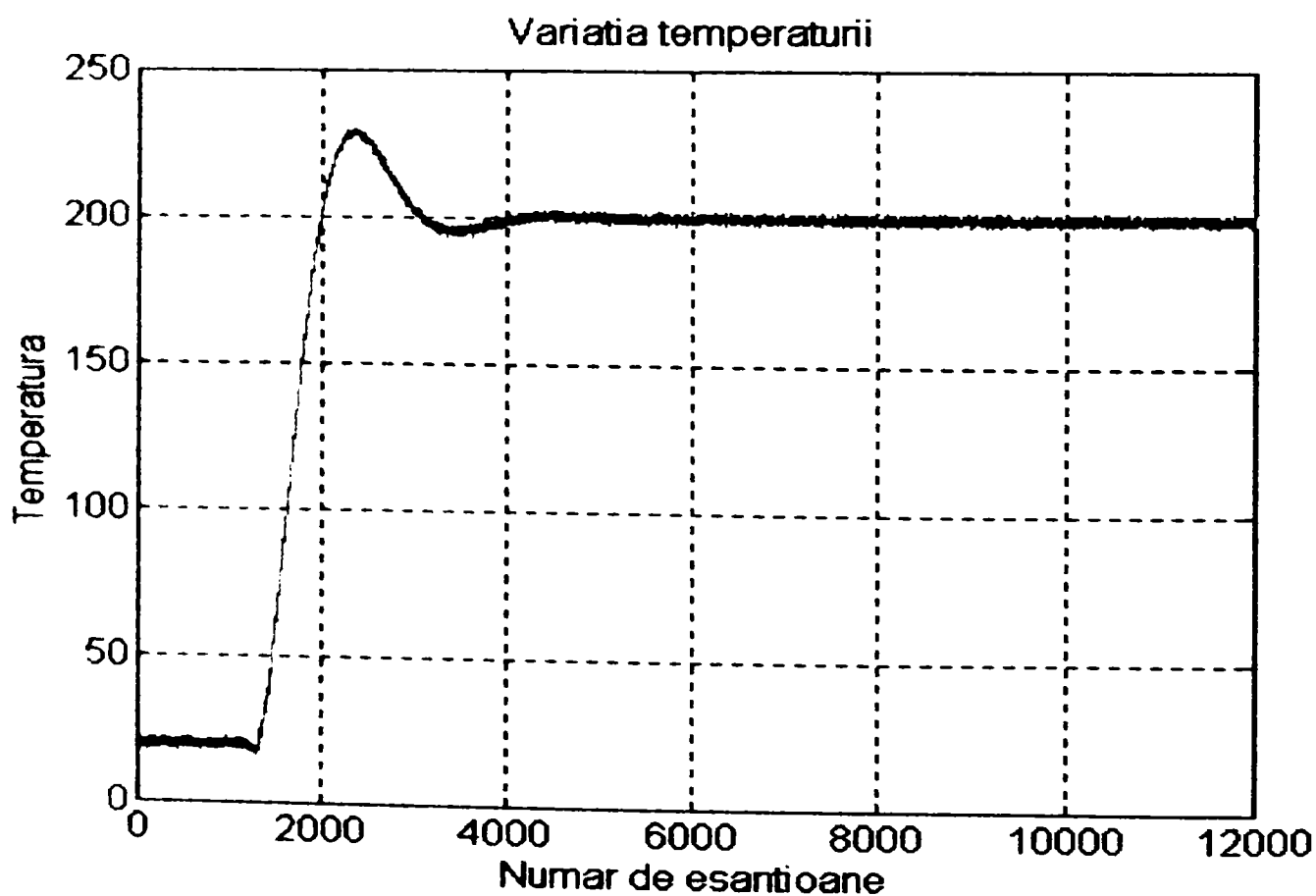


Fig. 5.33. Variația temperaturii în cazul utilizării relațiilor Chien, Hrones, Reswich pentru regulator PI, răspuns oscilant cu $\sigma = 20\%$.

5.7. Concluzii

În acest capitol autorul prezintă o metodă originală de identificare în timp real a parametrilor procesului de încălzire a cuptorului cu rezistență electrică. Verificarea metodei propuse s-a făcut prin compararea rezultatelor cu cele obținute prin metoda de identificare on-line. În continuare s-a prezentat un sistem de reglare adaptivă a temperaturii din interiorul cuptorului utilizând un regulator adaptiv numeric ai cărui parametrii se modifică în funcție de valorile parametrilor procesului obținute în urma identificării. Din cele prezentate se desprind următoarele concluzii:

1. Metoda de identificare on-line prezentată în primul paragraf prezintă două dezavantaje. Primul dezavantaj constă în faptul că, dacă durata impulsului de test și a duratei de integrare nu sunt alese în corelație cu valorile reale ale parametrilor procesului, valorile măsurate ale parametrilor procesului prezintă erori mari. Al doilea dezavantaj constă în faptul că, pentru măsurarea valorilor parametrilor procesului, sistemul trebuie scos din starea de echilibru, iar valorile măsurate sunt considerate constante până la o nouă măsurare.

2. Funcția de transfer a sistemului se poate obține prin utilizarea unei aproximații Pade a funcției de transfer asociată unui proces cu timp mort, iar funcția sistem a sistemului discret se obține prin utilizarea unei metode de echivalare a filtrului analogic cu un filtru numeric. Autorul a analizat două metode de echivalare a filtrului analogic cu un filtru numeric, metoda aproximării integralei prin metoda dreptunghiurilor și metoda aproximării integralei prin metoda trapezelor.

3. Autorul a determinat relațiile de calcul a coeficienților filtrului cunoscând valorile parametrilor procesului, precum și relațiile reciproce, de determinare a valorilor parametrilor procesului cunoscând valorile coeficienților filtrului adaptiv, pentru cele 4 aproximații Pade prezente în literatura de specialitate.

4. Autorul a studiat cei 3 algoritmi de identificare prezentați în capitolul 3, implementați atât în forma directă cât și în forma lattice ajungând la concluzia că indiferent de algoritm, forma de implementare lattice prezintă o viteză de convergență mai mare față de forma de implementare directă. De asemenea, algoritmul SHARF prezintă cele mai mici oscilații ale valorilor coeficienților, indiferent de forma de implementare.

5. Pe baza studiului efectuat, autorul a utilizat în procesul de identificare al parametrilor procesului de încălzire al cuptorului algoritmul SHARF implementat în forma lattice și utilizând metoda aproximației integralei prin metoda dreptunghiurilor.

6. În continuare s-a demonstrat experimental că, pentru ca filtrul adaptiv să poată fi utilizat la identificarea parametrilor proceselor lente este necesar ca, pe baza unei măsurări anterioare utilizând altă metodă, să se determine cu aproximație valorile parametrilor sistemului și utilizând aceste valori să se determine setul de valori inițiale ale coeficienților filtrului adaptiv.

7. Autorul a calculat coeficienții regulatorului adaptiv pentru reglatoare de tip P și PI, utilizând ambele metode de echivalare a filtrului analogic cu un filtru numeric.

8. În încheiere, sunt prezentate rezultatele experimentale obținute în procesul de reglare a temperaturii utilizând criteriile bazate pe rezultatele identificării în alegerea parametrilor reglatoarelor.

În concluzie, metoda de reglare a temperaturii propusă și experimentată de autor, prezintă următoarele avantaje:

- măsurarea parametrilor procesului de încălzire a cuptorului cu rezistență electrică constă în calculul acestora după fiecare interval de reglare a puterii debitate de rezistența electrică, cunoscând valorile reactualizate ale coeficienților filtrului adaptiv. În acest mod nu mai este necesară scoaterea sistemului din starea de echilibru ca în cazul metodei de identificare on-line.

- cunoscând practic în fiecare moment valoarea parametrilor procesului de încălzire a cuptorului, valorile parametrilor regulatorului numeric sunt alese în fiecare interval de reglare a puterii în funcție de valorile ale procesului

- performanțele obținute prin această metodă permit o mai bună reglare a temperaturii, în sensul reducerii timpului în care se atinge valoarea prestabilită a temperaturii în situația aplicării la intrarea de referință a regulatorului a unui semnal treaptă. De asemenea, după atingerea valorii impuse a temperaturii, oscilațiile în jurul acestei valori sunt mai mici în situația utilizării metodei propuse de autor decât în situația utilizării unui regulator cu parametrii constanți.

6. CONCLUZII ȘI CONTRIBUȚII ORIGINALE

Teza abordează problematica prelucrării adaptive a semnalelor, cuprinzând o prezentare sistematizată a principalelor metode și algoritmi adaptivi. Autorul a realizat un studiu referitor la performanțele acestora prin efectuarea de teste în condiții similare (același număr de evenimente conținând același număr de eșantioane) obținând concluzii importante. De asemenea, în teză sunt prezentate contribuțiile autorului referitoare la definirea unor noi tipuri de filtre adaptive precum și la studiul performanțelor acestora. În încheiere autorul prezintă o metodă originală de reglare a temperaturii în interiorul cuptorului cu rezistență electrică utilizând tehnicile adaptive de prelucrare a semnalelor.

În primul capitol sunt definite filtrele adaptive liniare și neliniare și sunt prezentați principalii factori care influențează alegerea unui algoritm adaptiv. De asemenea sunt prezentate principalele categorii de aplicații ale filtrelor adaptive.

În capitolul 2 autorul prezintă concluziile unui studiu comparativ în care sunt analizate performanțele filtrelor adaptive liniare. Astfel s-au obținut concluzii importante prin efectuarea de teste în aceleași condiții (număr de evenimente și număr de eșantioane / eveniment).

2.1. Autorul a simulat funcționarea unui canal de comunicații modelat de o funcție de transfer cu un pol, canal perturbat de zgomot cu caracter gaussian. Semnalul de la intrarea canalului a fost ales un proces autoregresiv de ordin I. În continuare s-au calculat coeficienții filtrului Wiener precum și eroarea medie pătratică minimă în ipoteza că filtrul este utilizat în scopul obținerii celei mai bune aproximări a semnalului de la intrarea canalului.

2.2. Deși valorile coeficienților filtrului Wiener depind de valorile coeficienților funcției de transfer a sistemului echivalent și de valoarea dispersiei zgomotului introdus de canalul de comunicație, autorul a verificat experimental faptul că performanțele comparative ale algoritmilor studiați rămân aceleași și coeficienții filtrelor adaptive converg spre valorile coeficienților filtrului Wiener.

2.3. Determinarea coeficienților filtrului adaptiv transversal pornind de la rezolvarea ecuației normale prezintă dezavantajul unui volum de calcule mare precum și a erorilor introduse în calculul matricei de autocorelație și a vectorului de intercorelație. Aceste erori sunt cauzate de estimatorul folosit.

Autorul a arătat experimental că algoritmul LMS permite reducerea volumului de calcule dar viteza de convergență este mai mică. De asemenea, nu s-a constatat experimental o creștere a erorii medii pătratice.

În scopul creșterii vitezei de convergență comparativ cu algoritmul LMS sunt prezentați 4 algoritmi adaptivi de reactualizare a coeficienților filtrului utilizând un pas de adaptare

variabil. Acești algoritmi permit creșterea vitezei de convergență, dar prezintă o creștere a volumului de calcule necesar implementării, reflectat în creșterea timpului de rulare a programelor.

În capitolul 3 sunt prezentate posibilitățile de implementare a filtrelor adaptive de tip IIR utilizând structuri lattice. Au fost analizate 5 structuri de filtre adaptive lattice, prezentându-se un studiu experimental referitor la performanțele filtrelor adaptive care utilizează aceste structuri. În urma realizării studiului autorul a ajuns la următoarele concluzii:

3.1. Analizând structura bazată pe algoritmul lui Durbin autorul a verificat experimental faptul că numărul de celule ale filtrului lattice utilizat într-o aplicație de identificare este determinat de structura sistemului de identificat. Creșterea numărului de celule ale filtrului lattice nu duce la reducerea erorii medii pătratice ci doar la creșterea volumului de calcule necesar implementării filtrului.

3.2. În cazul structurii bazate pe metoda gradientului adaptiv, s-a arătat faptul că stabilitatea filtrului depinde de valoarea parametrului de adaptare precum și de valoarea zgomotului care influențează canalul. Implementarea acestei structuri prezintă cel mai mic volum de calcule, reflectat în timpul de rulare al programului.

3.3. Rezolvarea problemei instabilității filtrelor se poate face utilizând structura bazată pe metoda gradientului adaptiv normalizat. Utilizarea acestei structuri conduce la o creștere a volumului de calcule necesar implementării filtrului adaptiv datorită necesității efectuării operației de normalizare a coeficientului de adaptare.

3.4. În cazul structurii bazate pe utilizarea conceptului de spații vectoriale autorul a determinat experimental forma de variație a coeficienților de reflexie “înainte” și “înapoi” precum și a erorii medii pătratice. S-a verificat experimental faptul că valoarea inițială a erorii reziduale de predicție influențează viteza de convergență dar valorile finale ale coeficienților de reflexie rămân aceleași. Datorită creșterii complexității algoritmului de implementare a filtrului adaptiv realizat cu această structură, volumul de calcule este mai mare în comparație cu situația utilizării structurilor precedente.

3.5. În cazul structurii bazate pe reactualizarea coeficientului unghiular autorul a determinat forma de variație a coeficienților de reflexie precum și influența parametrului de adaptare asupra convergenței valorilor coeficienților filtrului. Deși timpul de calcul necesar implementării soft a structurii bazată pe reactualizarea coeficientului unghiular este cel mai mare, această structură este des utilizată datorită existenței procesoarelor de semnal CORDIC care permit reducerea semnificativă a timpului de calcul.

Capitolul 4 prezintă un studiu realizat de autor referitor la posibilitățile de utilizare a filtrelor adaptive neliniare atât în scopul filtrării perturbațiilor care constau în impulsuri de amplitudine mare și durată scurtă, cât și în scopul prelucrării semnalelor cu nivelul mediei nestaționar. Autorul definește noi tipuri de filtre adaptive neliniare studiind proprietățile acestora și subliniind principalele lor avantaje.

4.1. Studiul experimental asupra proprietăților filtrelor OSLMS, în special a filtrelor MLMS și ALMS, a dus la obținerea concluziei că filtrul ALMS este mai rapid convergent decât filtrul MLMS. O altă concluzie la care a ajuns autorul a fost aceea că variația coeficienților de predicție în situația apariției unui impuls este mai mică în cazul filtrului MLMS decât în cazul filtrului ALMS, rezultând că filtrarea MLMS este recomandabilă în situația în care se dorește eliminarea impulsurilor cu amplitudine mare și durată scurtă. De

asemenea, datorită volumului suplimentar de calcule necesar implementării filtrului MLMS timpul de rulare pentru acest tip de filtru este mai mare decât în cazul filtrării ALMS pentru aceeași lungime a ferestrei.

4.2. Implementarea filtrului adaptiv bazat pe avantajele acestor două tipuri de filtre s-a făcut utilizând un parametru γ , de detecție a unui impuls în interiorul ferestrei, parametru definit în două moduri (relațiile 4.11, respectiv 4.12). S-a verificat experimental faptul că ambele moduri de definire duc la performanțe ale filtrului adaptiv similare, însă timpul de calcul necesar implementării filtrului adaptiv este mai mare în situația utilizării relației (4.11).

4.3. În paragraful 4.1.6 autorul a definit un nou tip de filtru adaptiv denumit MALMS, filtru care combină avantajele filtrelor MLMS și ALMS. Definirea acestui tip de filtru s-a făcut prin introducerea de către autor a unui coeficient de ponderare a celor două tipuri de filtre în funcție de amplitudinea parametrului de detecție a prezenței unui impuls în fereastra de date. Avantajul oferit de acest tip de filtru constă în faptul că pentru valori ale parametrului γ cuprinse între γ_{\min} și γ_{\max} (interval în care funcționează filtrul MALMS) este realizată atât o viteză de convergență mai mare decât în cazul filtrului ALMS, cât și o influență mai mică a impulsurilor de amplitudine mare și durată scurtă asupra variației coeficienților de predicție decât în cazul filtrului MLMS.

4.4. În paragraful 4.2. autorul a definit o nouă categorie de filtre adaptive IIR, denumite de autor IIR-OSLMS, care reactualizează coeficienții pe baza unuia din cei 3 algoritmi de identificare prezentați în capitolul precedent, în combinație cu o operație de ordonare. Autorul a definit relațiile de reactualizare ale coeficienților filtrelor adaptive și a studiat experimental caracteristicile filtrelor adaptive în procesul de identificare a parametrilor unui sistem necunoscut. În urma studiului efectuat autorul a desprins două concluzii importante.

Prima concluzie se referă la volumul de calcule necesar implementării acestor tipuri de filtre. S-a demonstrat experimental faptul că implementarea filtrului IIR necesită cel mai mic volum de calcule, în timp ce, datorită operațiilor de comparare necesare implementării filtrului Median-IIR, acesta necesită cel mai mare volum de calcule.

A doua concluzie la care s-a ajuns este aceea că cele mai bune performanțe în ceea ce privește reducerea influenței impulsurilor de amplitudine mare și durată scurtă asupra formei de variație a coeficienților le are filtrul Median-IIR, indiferent de de algoritmul utilizat și de forma de implementare. Rezultă că, deși timpul de calcul necesar implementării acestui filtru este cel mai mare, există aplicații în care utilizarea acestui tip de filtru prezintă avantaje în eliminarea influenței impulsurilor de amplitudine mare și durată scurtă. De asemenea, acest tip de filtru poate fi utilizat la identificarea în timp real a parametrilor proceselor lente.

4.5. O altă contribuție a autorului o constituie prezentarea sistematizată a principalelor tipuri de filtre neliniare întâlnite în literatura de specialitate (filtrele combinaționale, filtre WOS, FMH și FWH) care pot fi utilizate ca filtre adaptive, precum și a modului de implementare a acestora.

În capitolul 5 autorul prezintă o metodă originală de identificare în timp real a parametrilor procesului de încălzire a cuptorului cu rezistență electrică. Pe baza valorilor obținute pentru parametrii procesului este realizată reglarea temperaturii în interiorul cuptorului utilizând un regulator numeric, fiind prezentate rezultatele experimentale obținute, precum și

concluziile și contribuțiile autorului referitoare la avantajele oferite de metoda de reglare a temperaturii propusă.

5.1. Funcția de transfer aproximativă a sistemului care modelează procesul de încălzire al cuptorului cu rezistență se poate obține prin utilizarea unei aproximații Pade a funcției de transfer asociată unui proces cu o constantă de timp și timp mort, iar funcția sistem a sistemului discret se poate obține prin utilizarea unei metode de echivalare a filtrului analogic cu un filtru numeric. Autorul a analizat două metode de echivalare a filtrului analogic cu un filtru numeric, metoda aproximării integralei prin metoda dreptunghiurilor și metoda aproximării integralei prin metoda trapezelor. În urma analizei efectuate s-a ajuns la concluzia că utilizarea primei metode este mai eficientă din punct de vedere al numărului de coeficienți rezultați ai filtrului adaptiv precum și al relațiilor dintre coeficienții filtrului și parametrii sistemului.

5.2. Autorul a determinat relațiile de calcul ale coeficienților filtrului cunoscând valorile parametrilor procesului, precum și relațiile reciproce, de determinare a valorilor parametrilor procesului cunoscând valorile coeficienților filtrului adaptiv, pentru cele 4 aproximații Pade prezente în literatura de specialitate.

5.3. Studiul celor 3 algoritmi de identificare prezentați în capitolul 3, implementați atât în forma directă cât și în forma lattice a dus la concluzia că, indiferent de algoritm, forma de implementare lattice prezintă o viteză de convergență mai mare față de forma de implementare directă. De asemenea, algoritmul SHARF prezintă cele mai mici oscilații ale valorilor coeficienților, indiferent de forma de implementare. Pe baza studiului efectuat, autorul a utilizat în procesul de identificare al parametrilor procesului de încălzire a cuptorului algoritmul SHARF implementat în forma lattice, funcția sistem a sistemului discret obținându-se prin utilizarea metodei aproximației integralei prin metoda dreptunghiurilor.

5.4. Pentru ca filtrul adaptiv să poată fi utilizat la identificarea parametrilor proceselor lente autorul a demonstrat experimental că este necesar ca, pe baza unei măsurări anterioare utilizând altă metodă, să se determine cu aproximație valorile parametrilor sistemului. Setul de valori inițiale ale coeficienților filtrului adaptiv se determină utilizând valorile aproximative ale parametrilor sistemului.

5.5. Rezultatele experimentale obținute în procesul de reglare a temperaturii utilizând criteriile bazate pe rezultatele identificării în alegerea parametrilor reguletoarelor confirmă faptul că utilizarea metodei adaptive de reglare a temperaturii cuptorului propusă de autor prezintă avantaj față de metoda de identificare on-line.

Măsurarea parametrilor procesului de încălzire a cuptorului cu rezistență electrică constă în calculul acestora după fiecare interval de reglare a puterii debitate de rezistența electrică, cunoscând valorile reactualizate ale coeficienților filtrului adaptiv. În acest mod nu mai este necesară scoaterea sistemului din starea de echilibru ca în cazul metodei de identificare on-line.

Cunoscând practic în fiecare moment parametrii procesului de încălzire a cuptorului, valorile parametrilor regulatorului numeric sunt alese în fiecare interval de reglare a puterii în funcție de valorile parametrilor procesului.

Performanțele obținute prin această metodă permit o mai bună reglare a temperaturii, în sensul reducerii timpului în care se atinge valoarea prestabilită a temperaturii în situația aplicării la intrarea de referință a regulatorului a unui semnal treaptă. De asemenea, după atingerea valorii impuse a temperaturii, oscilațiile în jurul acestei valori sunt mai mici în situația utilizării metodei propuse de autor decât în situația utilizării unui regulator cu parametrii constanți.

7. BIBLIOGRAFIE

1. S.Thomas Alexander, "Adaptive Signal Processing", Springer Verlag New York Inc., 1986.
2. Simon Haykin, "Adaptive Filter Theory", Prentice Hall, Englewood Cliffs, New Jersey, 1991.
3. Allan V. Oppenheim, Ronald W. Schaffer, "Digital Signal Processing", Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1986.
4. Ioan Nafoaniță, Andrei Câmpeanu, Alexandru Isar, "Semnale Circuite și Sisteme", U.P.Timișoara, 1995.
5. A. Mateescu, S. Ciochină, N. Dumitriu, A. Șerbănescu, L. Stanciu, "Prelucrarea numerică a semnalelor", Editura Tehnică, 1997.
6. A. Isar, I. Nafoaniță, "Reprezentări timp-frecvență", Editura Politehnica, 1998.
7. Jacob Benesty, Pierre Duhamel, "A Fast Exact Least Mean Square Adaptive Algorithm", IEEE Transactions on Signal Processing, vol 40, No.12 December 1992, pag. 2905-2920.
8. Peter J. Voltz, Frank Kozin, "Almost-Sure Convergence of the Continuous-Time LMS Algorithm", IEEE Transactions on Signal Processing, vol 40, No.2 February 1992, pag. 395-401.
9. Victor Solo, "The Error Variance of LMS with Time-Varying Weights". IEEE Transactions on Signal Processing, vol 40, No.4 April 1992, pag. 803-813.
10. Hanan Herzberg, Raziël Haimi-Cohen, Yair Be'ery; "A Systolic Array Realization of an LMS Adaptive Filter and the Effect of Delayed Adaption", IEEE Transactions on Signal Processing, vol 40, No.11 November 1992, pag. 2799-2803.
11. George V. Moustakides, Sergios Theodoridis, "Fast Newton Transversal Filters - A New Class of Adaptive Estimation Algorithms", IEEE Transactions on Signal Processing, vol 39, No.10 October 1991, pag. 2184-2183.
12. K.C. Ho, P.C. Ching, "Performance Analysis of Split-Path LMS Adaptive Filter for AR Modeling", IEEE Transactions on Signal Processing, vol. 40, No.6 June 1992, pag. 1375-1382.

13. Joseph B. Evans, Ping Xue, Bede Liu, "Analysis and Implementation of Variable Step Size Adaptive Algorithms". IEEE Transactions on Signal Processing, vol 41, No.8 August 1993, pag. 2517-2535.
14. Raymond H. Kwong, Edward W. Johnston, "A Variable Step Size LMS Algorithm", IEEE Transactions on Signal Processing, vol 40, No.7 July 1992, pag. 1633-1642.
15. Paulo S.R. Diniz, Luiz W.P. Biscainho; "Optimal Variable Step Size for the LMS/Newton Algorithm with Application to Subband Adaptive Filtering", IEEE Transactions on Signal Processing, vol 40, No.11 November 1992, pag. 2825-2829.
16. C Pănoiu, M. Pănoiu, "Consideration about Implementation Of Variable Step Size Adaptive Algorithms", ", International Conference on Automation and Quality Control, Cluj-Napoca, 28-29 May, 1998.
17. C Pănoiu, M.Pănoiu, D.Negruț, "Implementation of Adaptive Transversals Filters using Variable Step Size Adaptive Algorithm", Automatic Control and Testing Conference A96-Theta 10-Cluj 1996.
18. Eweda Eweda, "Comparison of RLS, LMS and Sign Algorithms for Tracking Randomly Time-Varying Channels", IEEE Transactions on Signal Processing, vol 42, No.11 November 1994, pag. 2937-2944.
19. Allen D. Hillary, Roland T. Chin, "Iterative Wiener Filters for Image Restoration", IEEE Transactions on Signal Processing, vol 39, No.8 August 1991, pag. 1892-1899.
20. Kai X. Miao, H. (Howard) Fan, Milos Doroslovacki, "Cascade Lattice IIR Adaptive Filters". IEEE Transactions on Signal Processing, vol 42, No.4 April 1994, pag. 721-742.
21. Shiao-Wu Chen, James S. Gibson, "A New Unwindowed Lattice Filter for RLS Estimation. IEEE Transactions on Signal Processing", vol 40, No.9 September 1992, pag. 2158-2165.
22. Y.H. Hu, H.E. Liao, "CALF: A CORDIC Adaptive Lattice Filter", IEEE Transactions on Signal Processing, vol 40, No.4 April 1992, pag. 990-993.
23. Phillip A. Regalia, "Stable and Efficient Lattice Algorithms for Adaptive IIR Filtering", IEEE Transactions on Signal Processing, vol 40, No.2 February 1992, pag. 375-388.
24. Elias Koukoutsis, George Carayamis, Cristos C. Halkias, "Superlattice/Superladder Computational Organization for Linear Prediction and Optimal FIR Filtering", IEEE Transactions on Signal Processing, vol 39, No.10 October 1991, pag. 2199-2215.
25. Phillip A. Regalia, "An Improved Lattice - Based Adaptive IIR Notch Filter", IEEE Transactions on Signal Processing, vol 39, No.9 September 1991, pag. 2124-2128.
26. Jose A. Rodriguez - Fonollosa, Enrique Masgran, "Simplified Gradient Calculation in Adaptive IIR Lattice Filters", IEEE Transactions on Signal Processing, vol 39, No.7 July 1991, pag. 1702-1705.

27. C.Pănoiu, M.Pănoiu, A.Iagăr, "The Temperature Control System using Cascade Lattice IIR Adaptive Filters", Automatic Control and Testing Conference A96-Theta 10-Cluj 1996.
28. C.Pănoiu, D.Negruț, "Analiza metodelor de implementare a filtrelor numerice recursive", Sesiunea de Comunicări Științifice Națională - Reșița 1993.
29. Prashant P. Ganfhi, Saleem A. Kassam, "Design and Performance of Combination Filters for Signal Restoration", IEEE Transactions on Signal Processing, vol 39, No.7 July 1991, pag. 1524-1540.
30. G.R. Arce, N.C. Gallagher, T. A. Nodes, "Advances in Computer Vision and Image Processing, vol 2, cap. 3 - Median Filters: Theory for One and Two Dimensional Filters", 1986 by JAI Press Inc., pag 89-166.
31. Yifeng Fu, Geoffrey A. Williamson, Peter M. Clarkson, "Adaptive Algorithms for Non-Gaussian Noise Environments: The Order Statistic Least Mean Square Algorithms, IEEE Transactions on Signal Processing", vol 42, No.11 November 1994, pag. 2945-2954.
32. Long-Wen Chang Jin-Her Lin, "A Bit-Level Syntonic Array for Median Filter", IEEE Transactions on Signal Processing, vol 40, No.8 August 1992, pag. 2079-2083.
33. Liu Yin, Yrjo Neuvo, "Fast Adaption and Performance Characteristics of FIR-WOS Hybrid Filters", IEEE Transaction on Signal Processing, vol 42, No.7. July 1994, pag. 1610-1628.
34. Long-Wen Chang, Shang-Shung Yu, "A New Implementation of Generalised Order Statistic Filter by Threshold Decomposition", IEEE Transactions on Signal Processing, vol 40, No.12 December 1992, pag. 3063-3069.
35. Francesco Palmieri, Charles C. Boncelet, "LI-Filters-A New Class of Order Statistics Filters", IEEE Transactions on Acoustics Speech and Signal Processing, vol 37, No.5. May 1989, pag. 691-701.
36. Kenneth E. Barner, Gonzalo R. Arce, "Permutation Filters: A Class of Nonlinear Filters Based on Set Permutations", IEEE Transactions on Signal Processing, vol 42, No.4 April 1994, pag. 782-798.
37. J. Patrick Fitch, Edward J. Coyle, Neal C. Gallagher, "Median Filtering by Threshold Decomposition", IEEE Transactions on Acoustics, Speech and Signal Processing, vol ASSP 32, No.6, December 1984, pag. 1183-1188.
38. Pao-Ta Yu, "Some Representation Properties of Stack Filters", IEEE Transactions on Signal Processing, vol 40, No.9 September 1992, pag. 2261-2266.
39. Peter D. Wendt, Edward J. Coyle, Neal C. Gallagher, "Stack Filters", IEEE Transactions on Acoustics, Speech and Signal Processing, vol ASSP 34, No.4, August 1986, pag. 898-911.
40. Harold Loughbotham, David Eberby, "The WMMR Filters: A Class of Robust Edge Enhancers", IEEE Transactions on Signal Processing, vol 41, No.4 April 1993, pag. 1680-1685.

41. Tarek I. Haweel, Peter M. Claekson, "A Class of Order Statistic LMS Algorithms", IEEE Transactions on Signal Processing, vol. 40. No. 1. January 1992, pag.44-53.
42. C.Pănoiu, M. Pănoiu, "An Adaptive OSLMS Filter", International Conference on Automation and Quality Control, Cluj-Napoca, 28-29 May, 1998.
43. William A. Sethares, "Adaptive Algorithm with Nonlinear Data and Error Functions", IEEE Transactions on Signal Processing, vol 40, No.9 September 1992, pag. 2199-2206.
44. George-Othon A. Glentis, Nicholas Kalouptsidis, "Efficiente Order Recursive Algorithms for Multichannel Least Squares Filtering", IEEE Transactions on Signal Processing, vol 40, No 6 June 1992, pag. 1354-1374.
45. Jechang Jeong, William J. Williams, "A Unified Fast Recursive Algorithm for Data Shuffling in Varians Orders", IEEE Transactions on Signal Processing, vol 40, No.6 May 1992, pag. 1091-1095.
46. I. Dumitrache, S. Dumitru, I. MiHu, F. Munteanu, Gh. Muscă, C. Calcev, "Automatizări Electronice", E.D.P.București 1993.
47. N.Rusu, C.Pănoiu, M.Pănoiu, S.Anghel, "The Temperature Adaptive Control System of an Furnace with Electric Resistance using an IBM-PC", Buletinul Științific Timișoara, Tom 40 (54), Fascicula 1.2, pag. 123-132.
48. C.Pănoiu, D.Negruț. "Program de implementare a filtrelor numerice utilizînd transformata biliniară". Sesiunea de Comunicări Științifice Națională - Reșița 1993.
49. H.(Howard) Fan, Wiaqi Liu, "Delta Levinson and Schur-Type RLS Algorithms for Adaptive Signal Processing", IEEE Transactions on Signal Processing, vol 42, No.7 June 1994, pag. 1629-1639.
50. Henry Leng, Simon Han Kin. "Detection and Estimation Using an Adaptive Rational Function Filter", IEEE Transactions on Signal Processing, vol 42, No.12 December 1994, pag. 3366-3376.
- 51 John J. Shynk. "Adaptive IIR Filtering Using Parallel-Form Realizations", IEEE Transactions on Acoustics, Speech and Signal Processing, vol 37, No.4, April 1989, pag. 519-533.
52. Hong Fan. "A Structural Wiew of Asymptotic Convergence Speed of Adaptive IIR Filtering Algorithms: Part I - Infinite Precision Implementation", IEEE Transactions on Signal Processing, vol 41, No 4. April 1993, pag. 1493-1517.
53. Ji-Nan, Roft Unbehanen, "Bias-Remedy Least Mean Square Equation Error Algorithm for IIR Parameter Recursive Estimation", IEEE Transactions on Signal Processing, vol 40, No.1 January 1992, pag. 62-69.

54. Neng-Chung Hu, Hong-I Chang, Okan K. Ersoy, "Generalised Discrete Hartley Transforms", IEEE Transactions on Signal Processing, vol 40, No.12 December 1992, pag. 2931-2940.
55. K.C. Ho, P.C. Ching, Y.T. Chan, "A New Configuration for Convergence Speedup in Adaptive Time-Delay Estimation", IEEE Transactions on Signal Processing, vol 40, No.11 November 1992, pag. 2683-2691.
56. Surendra Prasad, Shiv Dutt Joshi, "A New Recursive Pseudo Least Squares Algorithm for ARMA Filtering and Modeling: Part I+II", IEEE Transactions on Signal Processing, vol 40, No.11 November 1992, pag. 2776-2783.
57. George-Othon A Glentis, Nicholas Kalouptsidis, "Fast Adaptive Algorithms for Multichannel Filtering and System Identifications", IEEE Transactions on Signal Processing, vol 40, No.10 October 1993, pag. 2433-2458.
58. J. Thomas Cilke, Delores M. Etter, "A New Adaptive Algorithm to Reduce Weight Fluctuations Caused by High Variance Data", IEEE Transactions on Signal Processing, vol 40, No.9 September 1992, pag. 2324-2327.
59. Dennis R. Morgan, Charles Sauford, "A Control Theory Approach to the Stability and Transient Analysis of the Filtered-X LMS Adaptive Noise Filter", IEEE Transactions on Signal Processing, vol 40, No.9 September 1992, pag. 2341-2346.
60. Philip A. Nelson, Haneo Hamada, Stephen J. Elliott, "Adaptive Inverse Filters for Stereophonic Sound Reproduction", IEEE Transactions on Signal Processing, vol 40, No.7 July 1992, pag. 1621-1632.
61. Daniel F. Marshall, W. Kenneth Jenkins, "A Fast Quasi-Newton Adaptive Filtering Algorithm", IEEE Transactions on Signal Processing, vol 40, No.7 July 1992, pag. 1652-1662.
62. Tonz F. Chan, Per C. Hansen, "A Look-Ahead Levinson Algorithm for General Toeplitz Systems", IEEE Transactions on Signal Processing, vol. 40, No.5 May 1992, pag. 1079-1090.
63. George Scheets, Donald R. Hummels, "An Adaptive Bit Time Estimation", IEEE Transactions on Signal Processing, vol 40, No.5 May 1992, pag. 1134-1142.
64. Bin Yang, Johann F. Bohme, "Rotation-Based RLS Algorithms : Unified Derivations, Numerical Properties, and Parallel Implementations", IEEE Transactions on Signal Processing, vol 40, No.5 May 1992, pag. 1151-1167.
65. Amrane Houacine, "Regularized Fast Recursive Least Squares Algorithms for Finite Memory Filtering", IEEE Transactions on Signal Processing, vol 40, No.4 April 1992, pag. 758-769.

- 66 Long-Wen Chang, Shen-Wen Lee, "Systolic Arrays for the Discrete Hartley Transform", IEEE Transactions on Signal Processing, vol 39, No.11 November 1991, pag. 2411-2499
- 67 Nikolas P. Galatsanos, Aggelos K. Katsaggelos, Roland T. Chin, Allen D. Hillery, "Least Squares Restoration of Multichannel Images", IEEE Transactions on Signal Processing, vol 39, No.10 October 1991, pag. 2222-2236.
68. Tulay Adali, Sasan H. Ardalan, "Analysis of a Stabilization Technique for the Fixed - Point Prewindowed RLS Algorithm", IEEE Transactions on Signal Processing, vol 39, No.9 September 1991, pag. 2079-2082.
69. Hector Perez, Shigeo Tsugii, "A Fast Parallel Form IIR Adaptive Filter Algorithm", IEEE Transactions on Signal Processing, vol 39, No.9 September 1991, pag. 2118-2122.
- 70 Gregory E. Bottomley, S.T. Alexander, "A Novel Approach for Stabilizing Recursive Least Squares Filters", IEEE Transactions on Signal Processing, vol 39, No.8 August 1991, pag. 1770-1779.
- 71 N. E. Hubbing, S. T. Alexander, "Statistical Analysis of Initialization Methods for RLS Adaptive Filters", IEEE Transactions on Signal Processing, vol 39, No.8 August 1991, pag. 1793-1804.
- 72 T W. Wong, C.P. Kwong, "Adaptive Filtering Using Hartley Transform and Overlap - Save Method", IEEE Transactions on Signal Processing, vol 39, No.7 July 1991, pag. 1708-1711.
73. *** ADA 3100/ ADA 3100A, User's Manual, Real Time Devices Inc., USA, 1991.
74. *** Signal View, User's Manual and Reference Guide, Real Time Devices Inc., USA, 1991.
75. *** Signal Math, User's Manual, Real Time Devices Inc., USA, 1991.

8. ANEXE

Anexa II.1

```
function eroare=eroare
    C=ones(21,21);
    [X,Y]=meshgrid(0:1:2,-2:1:0);
    Z=0.7586-X.*0.8668+Y.*0.6886+X.*Y.*1.2+X.*X.*1.1+Y.*Y.*1.1;
    subplot(2,2,1);
    mesh(X,Y,Z,C);
    view(+37.5,30);
    title('eroarea medie patratice');
    xlabel(' w0 ');
    ylabel(' w1 ');
    zlabel(' eroarea');
    grid;
    subplot(2,2,3);
    mesh(X,Y,Z,C);
    view(0,0);
    title('eroarea medie patratice');
    xlabel(' w0 ');
    ylabel(' w1 ');
    zlabel(' eroarea');
    grid;
    subplot(2,2,4);
    mesh(X,Y,Z,C);
    view(90,0);
    title('eroarea medie patratice');
    xlabel(' w0 ');
    ylabel(' w1 ');
    zlabel(' eroarea');
    grid;
    [X,Y]=meshdom(0:1:2,-2:1:0);
    Z=0.7586-X.*0.8668+Y.*0.6886+X.*Y.*1.2+X.*X.*1.1+Y.*Y.*1.1;
    subplot(2,2,2);
    c=contour(0:1:2,-2:1:0,Z,10);
    clabel(c,'manual');
    title('eroarea medie patratice');
    xlabel(' w0 ');
    ylabel(' w1 ');
    grid;
```



```
function filtru_pdm=filtru1

m=0.1;           % parametru de adaptare
R=[1 1 0.6,0.6 1.1]; % matricea de autocorelatie
P=[0.4334,-0.3443]; % matricea de intercorelatie
I=[1 0,0 1];    % matricea unitate
W0=[0,0];       % initializare matrice coeficienti
t=1:1:50;       % numar esantioane
w1=zeros(size(t)); % evolutia coeficient w1
w2=zeros(size(t)); % evolutie coeficient w2

    MM=I-m*R;           %
    S=I,                %
    for k=1:50          % determinarea
    W0=MM*W0+m*S*P;    % evolutiei
    S=S+MM;            % coeficientilor
    MM=MM*(I-m*R);    % de predictie
    w1(k)=W0(1,1),     %
    w2(k)=W0(2,1),    %
    end,               %

    save w11.dat w1 -ascii % salvare w1
    save w21.dat w2 -ascii % salvare w2

subplot(2,1,1);
plot(t,w1);
title('variatia coeficientului w1');
xlabel('numar esantioane');
ylabel('w1');
grid;

subplot(2,1,2);
plot(t,w2);
title('variatia coeficientului w2');
xlabel('numar esantioane');
ylabel('w2');
grid;
```

```
function filtru_pdm=filtru1
```

```

m=0.5;           % parametru de adaptare
R=[1.1 0.6;0.6 1.1]; % matricea de autocorelatie
P=[0.4334;-0.3443]; % matricea de intercorelatie
I=[1 0;0 1];    % matricea unitate
W0=[0;0];       % initializare matrice coeficienti
t=1:1:100;      % numar esantioane
w1=zeros(size(t)); % evolutia coeficient w1
w2=zeros(size(t)); % evolutie coeficient w2
wf1=zeros(size(t)); % evolutia coeficient w1
wf2=zeros(size(t)); % evolutie coeficient w2

[M,V]=eig(R);
MT=M.';
MTI=inv(MT);
PP=MT*P;
l1=V(1,1);
l2=V(2,2);
t1=PP(1,1)/l1;
t2=PP(2,1)/l2;
k1=1-m*t1;
k2=1-m*t2;
s1=1;
s2=1;

for k=1:100
    w1(k)=k1*W0(1,1)+m*PP(1,1)*s1; % determinarea
    w2(k)=k2*W0(2,1)+m*PP(2,1)*s2; % evolutiei
    s1=s1+k1; % coeficientilor
    s2=s2+k2; % de predictie
    k1=k1*(1-m*t1); %
    k2=k2*(1-m*t2); %
end

for k=1:100
    wf1(k)=MTI(1,1)*w1(k)+MTI(1,2)*w2(k);
    wf2(k)=MTI(2,1)*w1(k)+MTI(2,2)*w2(k);
end

save w13.dat wf1 -ascii % salvare w1
save w23.dat wf2 -ascii % salvare w2

```

```

function filtru_lms=filtru1

t=1:1:1000,    % numar de esantioane
m=0.05,       % parametru de ajustare
media=0,      % media semnalelor zgomot
var1=0.3,     % varianta zgomotului v1
var2=0.1;     % varianta zgomotului v2
n=200,        % numar de evenimente

er=zeros(size(t)),    % initializare matrice eroare medie patratica
wf1=zeros(size(t)),  % initializare matrice coef. w1 mediu
wf2=zeros(size(t)),  % initializare matrice coef. w2 mediu

t0=clock;
for k=1:n

    v1=sqrt(var1)*randn(size(t))+media;    % generare semnal v1
    v2=sqrt(var2)*randn(size(t))+media;    % generare semnal v2

    d=zeros(size(t)),    % initializare matrice semnal dorit
    x=zeros(size(t)),    % initializare matrice semnal iesire canal
    u=zeros(size(t)),    % initializare matrice semnal iesire canal + zgomot
    w1=zeros(size(t));   % initializare matrice coeficient 1 predictie
    w2=zeros(size(t));   % initializare matrice coeficient 2 predictie
    y=zeros(size(t));    % initializare matrice iesire filtru predictie

        d(1)=v1(1);      %
        for i=2:1000    % generare semnal
            d(i)=v1(i)-0.7943*d(i-1); % dorit
        end            %
        x(1)=d(1);      %
        for i=2:1000    % generare semnal
            x(i)=d(i)+0.9443*x(i-1); % iesire canal
        end            %
        for i=1:1000    % generare semnal
            u(i)=x(i)+v2(i);    % iesire canal +
        end            % zgomot

        w1(1)=0;        %
        wf1(1)=wf1(1)+w1(1); %
        w2(1)=0;        %
        wf2(1)=wf1(1)+w1(1); %
        y(1)=w1(1)*u(1); %
        e(1)=d(1)-y(1); %
        er(1)=e(1)*e(1)+er(1); %
        w1(2)=w1(1)+m*e(1)*u(1); %
        wf1(2)=wf1(1)+w1(2); %
        w2(2)=w2(1);    % calcul coeficienti
        wf2(2)=wf2(1)+w2(2); % de predictie
    end
end

```

```

y(2)=w1(2)*u(2)+w2(2)*u(1);      %   si eroare
for i=3:1000                       %   medie patratica
    e(i-1)=d(i-1)-y(i-1);          %
    er(i-1)=e(i-1)*e(i-1)+er(i-1); %
    w1(i)=w1(i-1)+m*e(i-1)*u(i-1); %
    w2(i)=w2(i-1)+m*e(i-1)*u(i-2); %
    wf1(i)=wf1(i)+w1(i);           %
    wf2(i)=wf2(i)+w2(i);           %
    y(i)=w1(i)*u(i)+w2(i)*u(i-1);  %
end                                  %
end
for i=1:1000                        %
    er(i)=er(i)/n;                  % calcul medie coeficienti
    wf1(i)=wf1(i)/n;               % si medie eroare
    wf2(i)=wf2(i)/n;               %
end                                  %

```

```

durata=etime(clock,t0)
save er1.dat er -ascii
save wf11.dat wf1 -ascii
save wf21.dat wf2 -ascii
axis('equal');
subplot(3,1,1);
plot(t,wf1);
title('Variatia coeficientului de predictie w1');
xlabel('Numar de esantioane');
ylabel('Coeficient w1');
grid;
subplot(3,1,2);
plot(t,wf2);
title('Variatia coeficientului de predictie w2');
xlabel('Numar de esantioane');
ylabel('Coeficient w2');
grid;

subplot(3,1,3);
plot(t,er);
title('Eroarea medie patratica');
xlabel('Numar de esantioane');
ylabel('Eroarea');
grid;

```

```

function filtru_vsa=filtru3

t=1:1:1000;           % numar de esantioane
p=0.005;             % parametru de ajustare
m0=2,               % parametru de schimbare de semn
m1=3,               % parametru de pastrare de semn
alfa=0.0001;        % coeficient de modificare a parametrului m
media=0;            % media semnalelor zgomot
var1=0.3,           % varianta zgomotului v1
var2=0.1,           % varianta zgomotului v2
n=200,              % numar de evenimente
mmin=0.0005;        % valoare minima a parametrului de ajustare
mmax=0.07,          % valoare maxima a parametrului de ajustare

er=zeros(size(t));  % initializare matrice eroare medie patratica
wf1=zeros(size(t)); % initializare matrice coef. w1 mediu
wf2=zeros(size(t)); % initializare matrice coef. w2 mediu
t0=clock,
for k=1:n

    v1=sqrt(var1)*randn(size(t))+media;    % generare semnal v1
    v2=sqrt(var2)*randn(size(t))+media;    % generare semnal v2

    d=zeros(size(t)); % initializare matrice semnal dorit
    x=zeros(size(t)); % initializare matrice semnal iesire canal
    u=zeros(size(t)); % initializare matrice semnal iesire canal + zgomot
    w1=zeros(size(t)); % initializare matrice coeficient 1 predictie
    w2=zeros(size(t)); % initializare matrice coeficient 2 predictie
    y=zeros(size(t)); % initializare matrice iesire filtru predictie
    m=zeros(size(t)); % initializare matrice parametru de ajustare
    s=zeros(size(t));

        d(1)=v1(1), %
        for i=2:1000 % generare semnal
            d(i)=v1(i)-0.7943*d(i-1); % dorit
        end %

        x(1)=d(1); %
        for i=2:1000 % generare semnal
            x(i)=d(i)+0.9443*x(i-1); % iesire canal
        end %

        for i=1:1000 % generare semnal
            u(i)=x(i)+v2(i); % iesire canal +
        end % zgomot

    m(1)=p,
    ro=0,
    r1=0,

```

```

w1(1)=0; %
wf1(1)=wf1(1)+w1(1); %
w2(1)=0; %
wf2(1)=wf1(1)+w1(1); %
y(1)=w1(1)*u(1); %
e(1)=d(1)-y(1); %
s(1)=e(1)*u(1); %
er(1)=e(1)*e(1)+er(1); %
w1(2)=w1(1)+m(1)*e(1)*u(1); %
wf1(2)=wf1(2)+w1(2); %
w2(2)=w2(1); % calcul coeficienti
wf2(2)=wf2(2)+w2(2); % de predictie
y(2)=w1(2)*u(2); % si eroare
for i=3:1000 % medie patratice
    e(i-1)=d(i-1)-y(i-1); %
    s(i-1)=e(i-1)*u(i-1); %
    if s(i-2)*s(i-1) < 0
        r0=r0+1;
        r1=0;
    else
        r1=r1+1;
        r0=0;
    end
    if r0 == m0
        r0=0;
        if mmin <= m(i-2)-alfa
            m(i-1)=m(i-2)-alfa;
        else
            m(i-1)=mmin;
        end
    else
        m(i-1)=m(i-2);
    end
    if r1 == m1
        r1=0;
        if mmax >= m(i-2)+alfa
            m(i-1)=m(i-2)+alfa;
        else
            m(i-1)=mmax;
        end
    else
        m(i-1)=m(i-2);
    end
    er(i-1)=e(i-1)*e(i-1)+er(i-1); %
    w1(i)=w1(i-1)+m(i-1)*e(i-1)*u(i-1); %
    w2(i)=w2(i-1)+m(i-1)*e(i-1)*u(i-2); %
    wf1(i)=wf1(i)+w1(i); %
    wf2(i)=wf2(i)+w2(i); %
    y(i)=w1(i)*u(i)+w2(i)*u(i-1); %
end %

```

```

end
    for i=1:1000
        er(i)=er(i)/n;
        wf1(i)=wf1(i)/n;
        wf2(i)=wf2(i)/n;
    end
durata=etime(clock,t0)
save er1.dat er -ascii
save wf11.dat wf1 -ascii
save wf21.dat wf2 -ascii

axis('equal');
subplot(3,1,1);
plot(t,wf1);
title('Variatia coeficientului de predictie w1');
xlabel('Numar de esantioane');
ylabel('Coeficient w1');
grid;

subplot(3,1,2);
plot(t,wf2);
title('Variatia coeficientului de predictie w2');
xlabel('Numar de esantioane');
ylabel('Coeficient w2');
grid;

subplot(3,1,3);
plot(t,er);
title('Eroarea medie patratica');
xlabel('Numar de esantioane');
ylabel('Eroarea');
grid;

```

```
function filtru_lattice=filtru1
```

```
t=1:1:1000; % numar de esantioane
media=0; % media semnalelor zgomot
var1=0.3; % varianta zgomotului v1
var2=0.1; % varianta zgomotului v2
n=200; % numar de evenimente

er1=zeros(size(t)); % initializare matrice eroare medie patratica pas 1
er2=zeros(size(t)); % initializare matrice eroare medie patratica pas 2
kf1=zeros(size(t)); % initializare matrice coeficient 1
kf2=zeros(size(t)); % initializare matrice coeficient 2

t0=clock;
for k=1:n

    v1=sqrt(var1)*randn(size(t))+media; % generare semnal v1
    v2=sqrt(var2)*randn(size(t))+media; % generare semnal v2

    d=zeros(size(t)); % initializare matrice semnal dorit
    x=zeros(size(t)); % initializare matrice semnal iesire canal
    u=zeros(size(t)); % initializare matrice semnal iesire canal + zgomot
    y=zeros(size(t)); % initializare matrice iesire filtru predictie
    k1=zeros(size(t)); % initializare matrice coeficient k1
    k2=zeros(size(t)); % initializare matrice coeficient k2
    eps0=zeros(size(t)); % initializare matrice eroare medie patratică pas 0
    eps1=zeros(size(t)); % initializare matrice eroare medie patratică pas 1
    eps2=zeros(size(t)); % initializare matrice eroare medie patratica pas 2
    ef1=zeros(size(t)); % initializare matrice eroare "inainte" pas 1
    ef2=zeros(size(t)); % initializare matrice eroare "inainte" pas 2
    eb1=zeros(size(t)); % initializare matrice eroare "inapoi" pas 1
    eb2=zeros(size(t)); % initializare matrice eroare "inapoi" pas 2

        d(1)=v1(1); %
        for i=2:1000 % generare semnal
            d(i)=v1(i)-0.7943*d(i-1); % dorit
        end %
        x(1)=d(1); %
        for i=2:1000 % generare semnal
            x(i)=d(i)+0.9443*x(i-1); % iesire canal
        end %
        for i=1:1000 % generare semnal
            u(i)=x(i)+v2(i); % iesire canal +
        end % zgomot

s=0;
for i=1:1000
    s=s+u(i)*u(i)/1000;
end
```



```

k1(1)=0,
k2(1)=0,
eps0(1)=s;
eps1(1)=(1-k1(1)*k1(1))*eps0(1);
eps2(1)=(1-k2(1)*k2(1))*eps1(1);
eps0(2)=s;
k1(2)=0,
k2(2)=0,
eps1(2)=(1-k1(2)*k1(2))*eps0(2);
eps2(2)=(1-k2(2)*k2(2))*eps1(2);
eps0(3)=s;
k1(3)=0,
k2(3)=0,
eps1(3)=(1-k1(3)*k1(3))*eps0(3);
eps2(3)=(1-k2(3)*k2(3))*eps1(3);
r0=u(1)*u(1)+u(2)*u(2)+u(3)*u(3);
r1=u(1)*u(2)+u(2)*u(3);
r2=u(1)*u(3);
r3=0;
    for p=4:1000
        r0=r0+x(p)*x(p);
        r1=r1+x(p)*x(p-1);
        r2=r2+x(p)*x(p-2);
        r3=r3+x(p)*x(p-3);
        eps0(p)=r0/p;
        k1(p)=r1/(p-1)/eps0(p);
        w1=k1(p);
        eps1(p)=(1-k1(p)*k1(p))*eps0(p);
        k2(p)=(r2/(p-2)-w1*r1/(p-1))/eps1(p);
        w2=k2(p);
        w1=w1*(1-k2(p));
        eps2(p)=(1-k2(p)*k2(p))*eps1(p);
        ef1(p)=x(p)-k1(p)*x(p-1);
        eb1(p)=x(p-1)-k1(p)*x(p);
        ef2(p)=ef1(p)-k2(p)*eb1(p-1);
        eb2(p)=eb1(p-1)-k2(p)*ef1(p);
        kf1(p)=kf1(p)+k1(p);
        kf2(p)=kf2(p)+k2(p);
    end
for i=1:1000
    er2(i)=er2(i)+eps2(i);
    er1(i)=er1(i)+eps1(i);
end
end
for i=1:1000
    er1(i)=er1(i)/n;           %
    er2(i)=er2(i)/n;         % calcul medie eroare
    kf1(i)=kf1(i)/n;         % pas 1 + 2
    kf2(i)=kf2(i)/n;         % si coeficienti
end                           % filtru

```

```
durata=etime(clock,t0)
```

```
subplot(3,1,1);  
plot(t,kf1,'c');  
title('Variatia coeficientilor filtrului ');  
xlabel('Numar de esantioane');  
ylabel('Coeficient k1 ');  
grid;
```

```
subplot(3,1,2);  
plot(t,kf2,'m');  
title('Variatia coeficientilor filtrului ');  
xlabel('Numar de esantioane');  
ylabel('Coeficient k2 ');  
grid;
```

```
subplot(3,1,3);  
plot(t,er1,'c',t,er2,'m');  
title('Variatia erorii medii patratice pentru 2 pasi ');  
xlabel('Numar de esantioane');  
ylabel('Eroarea medie patratice ');  
grid;
```

```
function filtru_lattice=filtru2
```

```
t=1:1:1000, % numar de esantioane
media=0, % media semnalelor zgomot
var1=0.3, % varianta zgomotului v1
var2=0.1, % varianta zgomotului v2
n=20, % numar de evenimente
```

```
er1=zeros(size(t)), % initializare matrice eroare medie patratica pas 1
er2=zeros(size(t)), % initializare matrice eroare medie patratica pas 2
er3=zeros(size(t)), % initializare matrice eroare medie patratica pas 3
kf1=zeros(size(t)), % initializare matrice coeficient 1
kf2=zeros(size(t)), % initializare matrice coeficient 2
kf3=zeros(size(t)), % initializare matrice coeficient 3
```

```
t0=clock;
```

```
for k=1:n
```

```
v1=sqrt(var1)*randn(size(t))+media; % generare semnal v1
v2=sqrt(var2)*randn(size(t))+media; % generare semnal v2
```

```
d=zeros(size(t)); % initializare matrice semnal dorit
x=zeros(size(t)); % initializare matrice semnal iesire canal
u=zeros(size(t)); % initializare matrice semnal iesire canal + zgomot
y=zeros(size(t)); % initializare matrice iesire filtru predictie
k1=zeros(size(t)); % initializare matrice coeficient k1
k2=zeros(size(t)); % initializare matrice coeficient k2
k3=zeros(size(t)); % initializare matrice coeficient k3
eps0=zeros(size(t)); % initializare matrice eroare medie patratica pas 0
eps1=zeros(size(t)); % initializare matrice eroare medie patratica pas 1
eps2=zeros(size(t)); % initializare matrice eroare medie patratica pas 2
eps3=zeros(size(t)); % initializare matrice eroare medie patratica pas 3
ef1=zeros(size(t)); % initializare matrice eroare "inainte" pas 1
ef2=zeros(size(t)); % initializare matrice eroare "inainte" pas 2
ef3=zeros(size(t)); % initializare matrice eroare "inainte" pas 3
eb1=zeros(size(t)); % initializare matrice eroare "inapoi" pas 1
eb2=zeros(size(t)); % initializare matrice eroare "inapoi" pas 2
eb3=zeros(size(t)); % initializare matrice eroare "inapoi" pas 3
    d(1)=v1(1); %
    for i=2:1000 % generare semnal
        d(i)=v1(i)-0.7943*d(i-1); % dorit
    end %
    x(1)=d(1); %
    for i=2:1000 % generare semnal
        x(i)=d(i)+0.9443*x(i-1); % iesire canal
    end %
    for i=1:1000 % generare semnal
        u(i)=x(i)+v2(i); % iesire canal +
    end % zgomot
```

```

s=0;
for i=1:1000
    s=s+u(i)*u(i)/1000;
end
k1(1)=0;
k2(1)=0;
k3(1)=0;
eps0(1)=s;
eps1(1)=(1-k1(1)*k1(1))*eps0(1);
eps2(1)=(1-k2(1)*k2(1))*eps1(1);
eps3(1)=(1-k3(1)*k3(1))*eps2(1);
eps0(2)=s;
k1(2)=0;
k2(2)=0;
k3(2)=0;
eps1(2)=(1-k1(2)*k1(2))*eps0(2);
eps2(2)=(1-k2(2)*k2(2))*eps1(2);
eps3(2)=(1-k3(2)*k3(2))*eps2(2);
eps0(3)=s;
k1(3)=0;
k2(3)=0;
k3(3)=0;
eps1(3)=(1-k1(3)*k1(3))*eps0(3);
eps2(3)=(1-k2(3)*k2(3))*eps1(3);
eps3(3)=(1-k3(3)*k3(3))*eps2(3);
r0=u(1)*u(1)+u(2)*u(2)+u(3)*u(3);
r1=u(1)*u(2)+u(2)*u(3);
r2=u(1)*u(3);
r3=0;
for p=4:1000
    r0=r0+x(p)*x(p);
    r1=r1+x(p)*x(p-1);
    r2=r2+x(p)*x(p-2);
    r3=r3+x(p)*x(p-3);
    eps0(p)=r0/p;
    k1(p)=r1/(p-1)/eps0(p);
    w1=k1(p);
    eps1(p)=(1-k1(p)*k1(p))*eps0(p);
    k2(p)=(r2/(p-2)-w1*r1/(p-1))/eps1(p);
    w2=k2(p);
    w1=w1*(1-k2(p));
    eps2(p)=(1-k2(p)*k2(p))*eps1(p);
    k3(p)=(r3/(p-3)-w1*r2/(p-2)-w2*r1/(p-1))/eps2(p);
    w3=k3(p);
    w11=w1-k3(p)*w2;
    w21=w2-k3(p)*w1;
    w1=w11;
    w2=w21;
    eps3(p)=(1-k3(p)*k3(p))*eps2(p);
    efl(p)=x(p)-k1(p)*x(p-1);

```

```

        eb1(p)=x(p-1)-k1(p)*x(p);
        ef2(p)=ef1(p)-k2(p)*eb1(p-1);
        eb2(p)=eb1(p-1)-k2(p)*ef1(p);
        ef3(p)=ef2(p)-k3(p)*eb2(p-1);
        eb3(p)=eb2(p-1)-k3(p)*ef2(p);
        kf1(p)=kf1(p)+k1(p);
        kf2(p)=kf2(p)+k2(p);
        kf3(p)=kf3(p)+k3(p);
    end

for i=1:1000
    er3(i)=er3(i)+eps3(i);
    er2(i)=er2(i)+eps2(i);
    er1(i)=er1(i)+eps1(i);
end
end

        for i=1:1000                %
            er1(i)=er1(i)/n;        % calcul medie eroare
            er2(i)=er2(i)/n;        % pas 1 + 2
            er3(i)=er3(i)/n;        %
            kf1(i)=kf1(i)/n;        % si coeficienti
            kf2(i)=kf2(i)/n;        %
            kf3(i)=kf3(i)/n;        % filtru
        end                        %

durata=etime(clock,t0)
subplot(4,1,1);
plot(t,kf1,'c');
title('Variatia coeficientilor filtrului ');
xlabel('Numar de esantioane');
ylabel('Coeficient k1 ');
grid;
subplot(4,1,2);
plot(t,kf2,'m');
title('Variatia coeficientilor filtrului ');
xlabel('Numar de esantioane');
ylabel('Coeficient k2 ');
grid;
subplot(4,1,3);
plot(t,kf3,'y');
title('Variatia coeficientilor filtrului ');
xlabel('Numar de esantioane');
ylabel('Coeficient k3 ');
grid;
subplot(4,1,4);
plot(t,er1,'c',t,er2,'m',t,er3,'y');
title('Variatia erorii medii patratice pentru 3 pasi ');
xlabel('Numar de esantioane');
ylabel('Eroarea medie patratice ');
grid;

```

```
function filtru_lattice=filtru13
```

```

t=1:1:1000;    % numar de esantioane
m=0.05;       % parametru de ajustare
media=0;      % media semnalelor zgomot
var1=0.3;     % varianta zgomotului v1
var2=0.1;     % varianta zgomotului v2
beta=0.075;   % coeficient adaptiv
n=200;        % numar de evenimente

er1=zeros(size(t)); % initializare matrice eroare medie patratica pas 1
er2=zeros(size(t)); % initializare matrice eroare medie patratica pas 2
kf1=zeros(size(t)); % initializare matrice coeficient 1
kf2=zeros(size(t)); % initializare matrice coeficient 2

t0=clock;
for k=1:n

    v1=sqrt(var1)*randn(size(t))+media;    % generare semnal v1
    v2=sqrt(var2)*randn(size(t))+media;    % generare semnal v2

    d=zeros(size(t));    % initializare matrice semnal dorit
    x=zeros(size(t));    % initializare matrice semnal iesire canal
    u=zeros(size(t));    % initializare matrice semnal iesire canal + zgomot
    y=zeros(size(t));    % initializare matrice iesire filtru predictie
    k1=zeros(size(t));   % initializare matrice coeficient k1
    k2=zeros(size(t));   % initializare matrice coeficient k2
    ef1=zeros(size(t));  % initializare matrice eroare "inainte" pas 1
    ef2=zeros(size(t));  % initializare matrice eroare "inainte" pas 2
    eb1=zeros(size(t));  % initializare matrice eroare "inapoi" pas 1
    eb2=zeros(size(t));  % initializare matrice eroare "inapoi" pas 2

        d(1)=v1(1);    %
        for i=2:1000    % generare semnal
            d(i)=v1(i)-0.7943*d(i-1); % dorit
        end            %
        x(1)=d(1);    %
        for i=2:1000    % generare semnal
            x(i)=d(i)+0.9443*x(i-1); % iesire canal
        end            %
        for i=1:1000    % generare semnal
            u(i)=x(i)+v2(i);    % iesire canal +
        end            % zgomot

    k1(1)=0;
    k2(1)=0;
    ef1(1)=u(1);
    eb1(1)=k1(1)*u(1);
    ef2(1)=0;
    eb2(1)=0;

```

```

k1(2)=k1(1)+beta*eb1(1)*u(1);
k2(2)=k2(1)+beta*eb1(1)*u(1);
    for p=2:999
        ef1(p)=u(p)-k1(p)*u(p-1);
        eb1(p)=u(p-1)-k1(p)*u(p);
        ef2(p)=ef1(p)-k2(p)*eb1(p-1);
        eb2(p)=eb1(p-1)-k2(p)*ef1(p);
        k1(p+1)=k1(p)+beta*(ef1(p)*u(p-1)+eb1(p)*u(p));
        k2(p+1)=k2(p)+beta*(ef2(p)*eb1(p-1)+eb2(p)*ef1(p));
        kf1(p)=kf1(p)+k1(p);
        kf2(p)=kf2(p)+k2(p);
    end
    end
for i=1:1000
    er2(i)=er2(i)+ef2(i)*ef2(i);
    er1(i)=er1(i)+ef1(i)*ef1(i);
end
end
    for i=1:1000
        er1(i)=er1(i)/n;
        er2(i)=er2(i)/n;
        kf1(i)=kf1(i)/n;
        kf2(i)=kf2(i)/n;
    end
m1=mean(er1)
m2=mean(er2)
k1=mean(kf1)
k2=mean(kf2)
durata=etime(clock,t0)
save kf11.dat kf1 -ascii
save kf21.dat kf2 -ascii
save er11.dat er1 -ascii
save er21.dat er2 -ascii
subplot(3,1,1),
plot(t,kf1,'c');
title('Variatia coeficientilor filtrului ');
xlabel('Numar de esantioane');
ylabel('Coeficient k1 ');
grid;
subplot(3,1,2);
plot(t,kf2,'m');
title('Variatia coeficientilor filtrului ');
xlabel('Numar de esantioane');
ylabel('Coeficient k2 ');
grid;
subplot(3,1,3);
plot(t,er1,'c',t,er2,'m');
title('Variatia erorii medii patratice pentru 2 pasi ');
xlabel('Numar de esantioane');
ylabel('Eroarea medie patratice ');
grid;

```

```

function filtru_lattice=filtru14
t=1:1:1000; % numar de esantioane
m=0.05; % parametru de ajustare
media=0; % media semnalelor zgomot
var1=0.3; % varianta zgomotului v1
var2=0.75; % varianta zgomotului v2
alfa=0.01; % coeficient adaptiv
n=200; % numar de evenimente
er1=zeros(size(t)); % initializare matrice eroare medie patratica pas 1
er2=zeros(size(t)); % initializare matrice eroare medie patratica pas 2
kf1=zeros(size(t)); % initializare matrice coeficient 1
kf2=zeros(size(t)); % initializare matrice coeficient 2
t0=clock;
for k=1:n
v1=sqrt(var1)*randn(size(t))+media; % generare semnal v1
v2=sqrt(var2)*randn(size(t))+media; % generare semnal v2
d=zeros(size(t)); % initializare matrice semnal dorit
x=zeros(size(t)); % initializare matrice semnal iesire canal
u=zeros(size(t)); % initializare matrice semnal iesire canal + zgomot
y=zeros(size(t)); % initializare matrice iesire filtru predictie
k1=zeros(size(t)); % initializare matrice coeficient k1
k2=zeros(size(t)); % initializare matrice coeficient k2
ef1=zeros(size(t)); % initializare matrice eroare "inainte" pas 1
ef2=zeros(size(t)); % initializare matrice eroare "inainte" pas 2
eb1=zeros(size(t)); % initializare matrice eroare "inapoi" pas 1
eb2=zeros(size(t)); % initializare matrice eroare "inapoi" pas 2
d1=zeros(size(t)); % initializare matrice coeficient de adaptare 1
d2=zeros(size(t)); % initializare matrice coeficient de adaptare 2
    d(1)=v1(1); %
    for i=2:1000 % generare semnal
        d(i)=v1(i)-0.7943*d(i-1); dorit
    end %
    x(1)=d(1); %
    for i=2:1000 % generare semnal
        x(i)=d(i)+0.9443*x(i-1); % iesire canal
    end %
    for i=1:1000 % generare semnal
        u(i)=x(i)+v2(i); % iesire canal +
    end % zgomot
k1(1)=0;
k2(1)=0;
ef1(1)=u(1);
eb1(1)=k1(1)*u(1);
ef2(1)=0;
eb2(1)=0;
d1(1)=ef1(1)*ef1(1)+eb1(1)*eb1(1);
d2(1)=1-alfa+ef2(1)*ef2(1)+eb2(1)*eb2(1);
k1(2)=k1(1)+1/d1(1)*eb1(1)*u(1);
k2(2)=k2(1)+1/d2(1)*eb1(1)*u(1);

```



```

for p=2:999
    ef1(p)=u(p)-k1(p)*u(p-1);
    eb1(p)=u(p-1)-k1(p)*u(p);
    ef2(p)=ef1(p)-k2(p)*eb1(p-1);
    eb2(p)=eb1(p-1)-k2(p)*ef1(p);
    d1(p)=(1-alfa)*d1(p-1)+ef1(p)*ef1(p)+eb1(p)*eb1(p);
    d2(p)=(1-alfa)*d2(p-1)+ef2(p)*ef2(p)+eb2(p)*eb2(p);
    k1(p+1)=k1(p)+1/d1(p)*(ef1(p)*u(p-1)+eb1(p)*u(p));
    k2(p+1)=k2(p)+1/d2(p)*(ef2(p)*eb1(p-1)+eb2(p)*ef1(p));
    kf1(p)=kf1(p)+k1(p);
    kf2(p)=kf2(p)+k2(p);
end
for i=1:1000
    er2(i)=er2(i)+ef2(i)*ef2(i);
    er1(i)=er1(i)+ef1(i)*ef1(i);
end
end
for i=1:1000
    er1(i)=er1(i)/n;
    er2(i)=er2(i)/n;
    kf1(i)=kf1(i)/n;
    kf2(i)=kf2(i)/n;
end
m1=mean(er1)
m2=mean(er2)
k1=mean(kf1)
k2=mean(kf2)
durata=etime(clock,t0)
save kf12.dat kf1 -ascii
save kf22.dat kf2 -ascii
save er12.dat er1 -ascii
save er22.dat er2 -ascii
subplot(3,1,1);
plot(t,kf1,'c');
title('Variatia coeficientilor filtrului ');
xlabel('Numar de esantioane');
ylabel('Coeficient k1 ');
grid;
subplot(3,1,2);
plot(t,kf2,'m');
title('Variatia coeficientilor filtrului ');
xlabel('Numar de esantioane');
ylabel('Coeficient k2 ');
grid;
subplot(3,1,3);
plot(t,er1,'c',t,er2,'m');
title('Variatia erorii medii patratice pentru 2 pasi ');
xlabel('Numar de esantioane');
ylabel('Eroarea medie patratice ');
grid;

```

```
function filtru_lattice=filtru15
```

```
t=1:1:1000; % numar de esantioane
media=0; % media semnalelor zgomot
var1=0.3; % varianta zgomotului v1
var2=0.1; % varianta zgomotului v2
alfa=0.01; % coeficient adaptiv
n=200; % numar de evenimente
delta=0.1; % parametru de adaptare
```

```
er1=zeros(size(t)); % initializare matrice eroare medie patratica pas 1
er2=zeros(size(t)); % initializare matrice eroare medie patratica pas 2
k1ff=zeros(size(t)); % initializare matrice coeficient 1 "inainte"
k1bf=zeros(size(t)); % initializare matrice coeficient 1 "inapoi"
k2ff=zeros(size(t)); % initializare matrice coeficient 2 "inainte"
k2bf=zeros(size(t)); % initializare matrice coeficient 2 "inapoi"
```

```
t0=clock;
for k=1:n
```

```
v1=sqrt(var1)*randn(size(t))+media; % generare semnal v1
v2=sqrt(var2)*randn(size(t))+media; % generare semnal v2
```

```
d=zeros(size(t)); % initializare matrice semnal dorit
x=zeros(size(t)); % initializare matrice semnal iesire canal
u=zeros(size(t)); % initializare matrice semnal iesire canal + zgomot
y=zeros(size(t)); % initializare matrice iesire filtru predictie
k1f=zeros(size(t)); % initializare matrice coeficient k1 "inainte"
k1b=zeros(size(t)); % initializare matrice coeficient k1 "inapoi"
k2f=zeros(size(t)); % initializare matrice coeficient k2 "inainte"
k2b=zeros(size(t)); % initializare matrice coeficient k2 "inapoi"
e1f=zeros(size(t)); % initializare matrice eroare "inainte" pas 1
e2f=zeros(size(t)); % initializare matrice eroare "inainte" pas 2
e1b=zeros(size(t)); % initializare matrice eroare "inapoi" pas 1
e2b=zeros(size(t)); % initializare matrice eroare "inapoi" pas 2
delta1=zeros(size(t)); % initializare matrice coeficient de adaptare 1
delta2=zeros(size(t)); % initializare matrice coeficient de adaptare 2
g1=zeros(size(t)); % initializare matrice amplificare pas 1
g2=zeros(size(t)); % initializare matrice amplificare pas 2
eps0f=zeros(size(t)); % initializare matrice eroare patratica
eps0b=zeros(size(t)); % initializare matrice eroare patratica
eps1f=zeros(size(t)); % initializare matrice eroare patratica
eps1b=zeros(size(t)); % initializare matrice eroare patratica
eps2f=zeros(size(t)); % initializare matrice eroare patratica
eps2b=zeros(size(t)); % initializare matrice eroare patratica
    d(1)=v1(1); %
    for i=2:1000 % generare semnal
        d(i)=v1(i)-0.7943*d(i-1); % dorit
    end %
```

```

x(1)=d(1);           %
for i=2:1000        % generare semnal
    x(i)=d(i)+0.9443*x(i-1); % iesire canal
end                %

for i=1:1000        % generare semnal
    u(i)=x(i)+v2(i);  % iesire canal +
end                % zgomot

e0b(1)=0;
e1b(1)=0;
e2b(1)=0;
delta1(1)=0;
delta2(1)=0;
g0(1)=1;
g1(1)=1;
g2(1)=1;
eps0b(1)=delta;
eps0f(1)=delta;
eps1f(1)=delta;
eps2f(1)=delta;
eps1b(1)=delta;
eps2b(1)=delta;
for i=2:1000
    e0b(i)=u(i);
    e0f(i)=u(i);
    eps0b(i)=eps0b(i-1)+u(i)*u(i);
    eps0f(i)=eps0f(i-1)+u(i)*u(i);
    g0(i)=1;
    delta1(i)=delta1(i-1)+e0b(i-1)*e0f(i)/g0(i-1);
    e1f(i)=e0f(i)-delta1(i)*e0b(i-1)/eps0b(i-1);
    e1b(i)=e0b(i-1)-delta1(i)*e0f(i)/eps0f(i);
    eps1f(i)=eps0f(i)-delta1(i)*delta1(i)/eps0b(i-1);
    eps1b(i)=eps0b(i-1)-delta1(i)*delta1(i)/eps0f(i);
    g1(i-1)=g0(i-1)-e0b(i-1)*e0b(i-1)/eps0b(i-1);
    k1f(i)=delta1(i)/eps0f(i);
    k1b(i)=delta1(i)/eps0b(i-1);
    delta2(i)=delta2(i-1)+e1b(i-1)*e1f(i)/g1(i-1);
    e2f(i)=e1f(i)-delta2(i)*e1b(i-1)/eps1b(i-1);
    e2b(i)=e1b(i-1)-delta2(i)*e1f(i)/eps1f(i);
    eps2f(i)=eps1f(i)-delta2(i)*delta2(i)/eps1b(i-1);
    eps2b(i)=eps1b(i-1)-delta2(i)*delta2(i)/eps1f(i);
    g2(i-1)=g1(i-1)-e1b(i-1)*e1b(i-1)/eps1b(i-1);
    k2f(i)=delta2(i)/eps1f(i);
    k2b(i)=delta2(i)/eps1b(i-1);
    k1ff(i)=k1ff(i)+k1f(i);
    k1bf(i)=k1bf(i)+k1b(i);
    k2ff(i)=k2ff(i)+k2f(i);
    k2bf(i)=k2bf(i)+k2b(i);
end

```

```

for i=1:1000
    er2(i)=er2(i)+e2f(i)*e2f(i);
    er1(i)=er1(i)+e1f(i)*e1f(i);
end
end
    for i=1:1000                %
        er1(i)=er1(i)/n;        % calcul medie eroare
        er2(i)=er2(i)/n;        % pas 1 + 2
        k1ff(i)=k1ff(i)/n;      % si coeficienti
        k2ff(i)=k2ff(i)/n;      % filtru
        k1bf(i)=k1bf(i)/n;      %
        k2bf(i)=k2bf(i)/n;      %
    end                          %
durata=etime(clock,t0)
save k1ff3.dat k1ff -ascii
save k2ff3.dat k2ff -ascii
save k1bf3.dat k1bf -ascii
save k2bf3.dat k2bf -ascii
save er13.dat er1 -ascii
save er23.dat er2 -ascii
subplot(3,1,1);
plot(t,k1ff,'c',t,k1bf,'y');
title('Variatia coeficientilor filtrului ');
xlabel('Numar de esantioane');
ylabel('Coeficienti k1 ');
grid;
subplot(3,1,2);
plot(t,k2ff,'c',t,k2bf,'y');
title('Variatia coeficientilor filtrului ');
xlabel('Numar de esantioane');
ylabel('Coeficienti k2 ');
grid;
subplot(3,1,3);
plot(t,er1,'c',t,er2,'m');
title('Variatia erorii medii patratice pentru 2 pasi ');
xlabel('Numar de esantioane');
ylabel('Eroarea medie patratice ');
grid;

```

```

function filtru_lattice=filtru6

t=1:1:1000,           % numar de esantioane
media=0,              % media semnalelor zgomot
var1=0.3,             % varianta zgomotului v1
var2=0.1,             % varianta zgomotului v2
m=0.01,              % coeficient adaptiv
n=200,               % numar de evenimente

er1=zeros(size(t)),  % initializare matrice eroare medie patratica pas 1
er2=zeros(size(t)),  % initializare matrice eroare medie patratica pas 2
kf1=zeros(size(t)),  % initializare matrice coeficient 1
kf2=zeros(size(t)),  % initializare matrice coeficient 2

t0=clock;
for k=1:n

    v1=sqrt(var1)*randn(size(t))+media;    % generare semnal v1
    v2=sqrt(var2)*randn(size(t))+media;    % generare semnal v2
    d=zeros(size(t)),    % initializare matrice semnal dorit
    x=zeros(size(t)),    % initializare matrice semnal iesire canal
    u=zeros(size(t)),    % initializare matrice semnal iesire canal + zgomot
    y=zeros(size(t)),    % initializare matrice iesire filtru predictie
    k1=zeros(size(t)),   % initializare matrice coeficient k1
    k2=zeros(size(t)),   % initializare matrice coeficient k2
    ef1=zeros(size(t)),  % initializare matrice eroare "inainte" pas 1
    ef2=zeros(size(t)),  % initializare matrice eroare "inainte" pas 2
    eb1=zeros(size(t)),  % initializare matrice eroare "inapoi" pas 1
    eb2=zeros(size(t)),  % initializare matrice eroare "inapoi" pas 2
    teta1=zeros(size(t)), % initializare matrice coeficienti unghiulari pas 1
    teta2=zeros(size(t)), % initializare matrice coeficienti unghiulari pas 2
        d(1)=v1(1),           %
        for i=2:1000         % generare semnal
            d(i)=v1(i)-0.7943*d(i-1); % dorit
        end                 %
        x(1)=d(1),           %
        for i=2:1000         % generare semnal
            x(i)=d(i)+0.9443*x(i-1); % iesire canal
        end                 %
        for i=1:1000         % generare semnal
            u(i)=x(i)+v2(i);   % iesire canal +
        end                 % zgomot

    k1(1)=0;
    k2(1)=0;
    ef1(1)=u(1);
    eb1(1)=k1(1)*u(1);
    ef2(1)=0;
    eb2(1)=0;
    teta1(1)=0.

```

```

teta2(1)=0;
  for p=2:999
    teta1(p)=teta1(p-1)+m*ef1(p-1)*eb1(p-1);
    teta2(p)=teta2(p-1)+m*ef2(p-1)*eb2(p-1);
    ef1(p)=cosh(teta1(p))*u(p)-sinh(teta1(p))*u(p-1);
    eb1(p)=-sinh(teta1(p))*u(p)+cosh(teta1(p))*u(p-1);
    ef2(p)=cosh(teta2(p))*ef1(p)-sinh(teta2(p))*eb1(p-1);
    eb2(p)=-sinh(teta2(p))*ef1(p)+cosh(teta2(p))*eb1(p-1);
    k1(p)=k1(p-1)+m*ef1(p)*eb1(p)/(1-k1(p-1)*k1(p-1));
    k2(p)=k2(p-1)+m*ef2(p)*eb2(p)/(1-k2(p-1)*k2(p-1));
    kf1(p)=kf1(p)+k1(p);
    kf2(p)=kf2(p)+k2(p);
    er2(p)=er2(p)+ef2(p)*ef2(p);
    er1(p)=er1(p)+ef1(p)*ef1(p);
  end
end
for i=1:1000
    er1(i)=er1(i)/n;
    er2(i)=er2(i)/n;
    kf1(i)=kf1(i)/n;
    kf2(i)=kf2(i)/n;
end
m1=mean(er1)
m2=mean(er2)
k1=mean(kf1)
k2=mean(kf2)
durata=etime(clock,t0)
save kf11.dat kf1 -ascii
save kf21.dat kf2 -ascii
save er11.dat er1 -ascii
save er21.dat er2 -ascii
subplot(3,1,1);
plot(t,kf1,'c');
title('Variatia coeficientilor filtrului ');
xlabel('Numar de esantioane');
ylabel('Coeficient k1 ');
grid;
subplot(3,1,2);
plot(t,kf2,'m');
title('Variatia coeficientilor filtrului ');
xlabel('Numar de esantioane');
ylabel('Coeficient k2 ');
grid;
subplot(3,1,3);
plot(t,er1,'c',t,er2,'m');
title('Variatia erorii medii patratice pentru 2 pasi ');
xlabel('Numar de esantioane');
ylabel('Eroarea medie patratice ');
grid,

```

```

function filtru_mlms=filtru1

t=1:1:1000;    % numar de esantioane
m=0.01;       % parametru de ajustare
media=0;      % media semnalelor zgomot
var1=0.3;     % varianta zgomotului v1
var2=0.1;     % varianta zgomotului v2
n=200;        % numar de evenimente
lung=3;       % latimea ferestrei de filtrare

er=zeros(size(t)),          % initializare matrice eroare medie patratica
wf1=zeros(size(t)),        % initializare matrice coef. w1 mediu
wf2=zeros(size(t)),        % initializare matrice coef. w2 mediu

t0=clock;
for k=1:n

    v1=sqrt(var1)*randn(size(t))+media;    % generare semnal v1
    v2=sqrt(var2)*randn(size(t))+media;    % generare semnal v2

    d=zeros(size(t)),          % initializare matrice semnal dorit
    x=zeros(size(t)),          % initializare matrice semnal iesire canal
    u=zeros(size(t)),          % initializare matrice semnal iesire canal + zgomot
    w1=zeros(size(t)),         % initializare matrice coeficient 1 predictie
    w2=zeros(size(t)),         % initializare matrice coeficient 2 predictie
    y=zeros(size(t)),          % initializare matrice iesire filtru predictie
    sec1=zeros(size(t)),       % initializare matrice 1 produs eroare*date
    sec2=zeros(size(t)),       % initializare matrice 2 produs eroare*date
    med1=zeros(size(lung)),    % initializare matrice fereastra 1
    med2=zeros(size(lung));    % initializare matrice fereastra 2

        d(1)=v1(1);            %
        for i=2:1000           % generare semnal
            d(i)=v1(i)-0.7943*d(i-1); % dorit
        end                    %

        x(1)=d(1);             %
        for i=2:1000           % generare semnal
            x(i)=d(i)+0.9443*x(i-1); % iesire canal
        end                    %

        for i=1:1000           % generare semnal
            u(i)=x(i)+v2(i);    % iesire canal +
        end                    % zgomot
    end
end

```

```

u(400)=u(400)+1;
u(600)=u(600)+2;
u(800)=u(800)+3;
    w1(1)=0; %
    wf1(1)=wf1(1)+w1(1); %
    w2(1)=0; %
    wf2(1)=wf1(1)+w1(1); %
    y(1)=w1(1)*u(1); %
    e(1)=d(1)-y(1); %
    med1(1)=e(1)*u(1); %
    med2(1)=0; %
    er(1)=e(1)*e(1)+er(1); %
    w1(2)=w1(1); %
    wf1(2)=wf1(2)+w1(2); %
    w2(2)=w2(1); % calcul coeficienti
    wf2(2)=wf2(2)+w2(2); % de predictie
    y(2)=w1(2)*u(2)+w2(2)*u(1); % si eroare
    e(2)=d(2)-y(2); % medie patratica
    med1(2)=e(2)*u(2); %
    med2(2)=e(2)*u(1); %
    er(2)=e(2)*e(2)+er(1); %
    for i=3:1000 %
        w1(i)=w1(i-1)+m*median(med1); %
        w2(i)=w2(i-1)+m*median(med2); %
        wf1(i)=wf1(i)+w1(i); %
        wf2(i)=wf2(i)+w2(i); %
        p=i-lung*floor((i-1)/lung); %
        y(i)=w1(i)*u(i)+w2(i)*u(i-1); %
        e(i)=d(i)-y(i); %
        er(i)=e(i)*e(i)+er(i); %
        med1(p)=e(i)*u(i); %
        med2(p)=e(i)*u(i-1); %
    end %
end

    for i=1:1000 %
        er(i)=er(i)/n; % calcul medie coeficienti
        wf1(i)=wf1(i)/n; % si medie eroare
        wf2(i)=wf2(i)/n; %
    end %

durata=etime(clock,t0)
save er3.dat er -ascii
save wf13.dat wf1 -ascii
save wf23.dat wf2 -ascii

```



```
axis('equal');  
subplot(3,1,1);  
plot(t,wf1);  
title('Variatia coeficientului de predictie w1');  
xlabel('Numar de esantioane');  
ylabel('Coeficient w1');  
grid;
```

```
subplot(3,1,2);  
plot(t,wf2);  
title('Variatia coeficientului de predictie w2');  
xlabel('Numar de esantioane');  
ylabel('Coeficient w2');  
grid;
```

```
subplot(3,1,3);  
plot(t,er);  
title('Eroarea medie patratica');  
xlabel('Numar de esantioane');  
ylabel('Eroarea');  
grid;
```

```
function filtru_oslms1=filtru1
```

```
t=1:1:1000;    % numar de esantioane
m=0.05;       % parametru de ajustare
media=0;      % media semnalelor zgomot
var1=0.3;     % varianta zgomotului v1
var2=0.1;     % varianta zgomotului v2
n=200;        % numar de evenimente
lung=13;      % latimea ferestrei de filtrare
er=zeros(size(t));           % initializare matrice eroare medie patratice
wf1=zeros(size(t));         % initializare matrice coef. w1 mediu
wf2=zeros(size(t));         % initializare matrice coef. w2 mediu
gamaf=zeros(size(t));       % initializare matrice gama mediu
t0=clock;

for k=1:n
    v1=sqrt(var1)*randn(size(t))+media;    % generare semnal v1
    v2=sqrt(var2)*randn(size(t))+media;    % generare semnal v2
    d=zeros(size(t));                     % initializare matrice semnal dorit
    x=zeros(size(t));                     % initializare matrice semnal iesire canal
    u=zeros(size(t));                     % initializare matrice semnal iesire canal + zgomot
    w1=zeros(size(t));                    % initializare matrice coeficient 1 predictie
    w2=zeros(size(t));                    % initializare matrice coeficient 2 predictie
    y=zeros(size(t));                     % initializare matrice iesire filtru predictie
    sec1=zeros(size(t));                   % initializare matrice 1 produs eroare*date
    sec2=zeros(size(t));                   % initializare matrice 2 produs eroare*date
    med1=zeros(size(lung));                % initializare matrice fereastră 1
    med2=zeros(size(lung));                % initializare matrice fereastră 2
    fer=zeros(size(lung));                 % initializare matrice fereastră semnal
    gama=zeros(size(t));                   % initializare matrice gama

    d(1)=v1(1);                            %
    for i=2:1000                             % generare semnal
        d(i)=v1(i)-0.7943*d(i-1);           % dorit
    end                                        %

    x(1)=d(1);                               %
    for i=2:1000                             % generare semnal
        x(i)=d(i)+0.9443*x(i-1);           % iesire canal
    end                                        %

    for i=1:1000                             % generare semnal
        u(i)=x(i)+v2(i);                     % iesire canal +
    end                                        % zgomot
```

```

u(400)=u(400)+1;
u(600)=u(600)+2;
u(800)=u(800)+3;

for i=1:lung %
    fer(i)=0; %
end %
fer(1)=u(1); %
gama(1)=0; % calcul
gamaf(1)=gamaf(1)+gama(1); % coeficient
for i=2:1000 % gama l
    p=i-lung*floor((i-1)/lung); %
    fer(p)=u(i); %
    snr=0; %
    snum=0; %
    for k=1:lung %
        snr=snr+(fer(k)-mean(fer))^4; %
        snum=snum+(fer(k)-mean(fer))^2; %
    end %
    gama(i)=lung*snr/snum/snum; %
    gamaf(i)=gamaf(i)+gama(i); %
end %
prag=mean(gama)+0.1*(max(gama)-mean(gama)); %

w1(1)=0; %
wf1(1)=wf1(1)+w1(1); %
w2(1)=0; %
wf2(1)=wf1(1)+w1(1); %
y(1)=w1(1)*u(1); %
e(1)=d(1)-y(1); %
med1(1)=e(1)*u(1); %
med2(1)=0; %
er(1)=e(1)*e(1)+er(1); %
w1(2)=w1(1); %
wf1(2)=wf1(2)+w1(2); %
w2(2)=w2(1); % calcul coeficienti
wf2(2)=wf2(2)+w2(2); % de predictie
y(2)=w1(2)*u(2)+w2(2)*u(1); % si eroare
e(2)=d(2)-y(2); % medie patratice
med1(2)=e(2)*u(2); %
med2(2)=e(2)*u(1); %
er(2)=e(2)*e(2)+er(1); %
for i=3:1000 %
if gama(i)>prag %
    w1(i)=w1(i-1)+m*median(med1); %

```

```

        w2(i)=w2(i-1)+m*median(med2); %
    else %
        w1(i)=w1(i-1)+m*mean(med1); %
        w2(i)=w2(i-1)+m*mean(med2); %
    end %
    wf1(i)=wf1(i)+w1(i); %
    wf2(i)=wf2(i)+w2(i); %
    p=i-lung*floor((i-1)/lung); %
    y(i)=w1(i)*u(i)+w2(i)*u(i-1); %
    e(i)=d(i)-y(i); %
    er(i)=e(i)*e(i)+er(i); %
    med1(p)=e(i)*u(i); %
    med2(p)=e(i)*u(i-1); %
end %
end

for i=1:1000 %
    er(i)=er(i)/n; % calcul medie coeficienti
    wf1(i)=wf1(i)/n; % si medie eroare
    wf2(i)=wf2(i)/n; %
    gamaf(i)=gamaf(i)/n; %
end %

durata=etime(clock,t0)
save gamal.dat gamaf -ascii
save wf11.dat wf1 -ascii
save wf21.dat wf2 -ascii
axis('equal');
subplot(3,1,1);
plot(t,wf1);
title('Variatia coeficientului de predictie w1');
xlabel('Numar de esantioane');
ylabel('Coeficient w1');
grid;
subplot(3,1,2);
plot(t,wf2);
title('Variatia coeficientului de predictie w2');
xlabel('Numar de esantioane');
ylabel('Coeficient w2');
grid;
subplot(3,1,3);
plot(t,gamaf);
title('Variatia coeficientului gamal');
xlabel('Numar de esantioane');
ylabel('Gamal');
grid;

```

```
function filtru_oslms2=filtru1
```

```

t=1:1:1000,    % numar de esantioane
m=0.05,       % parametru de ajustare
media=0,      % media semnalelor zgomot
var1=0.3,     % varianta zgomotului v1
var2=0.1,     % varianta zgomotului v2
n=200,        % numar de evenimente
lung=13;      % latimea ferestrei de filtrare
er=zeros(size(t)),    % initializare matrice eroare medie patratice
wf1=zeros(size(t)),  % initializare matrice coef. w1 mediu
wf2=zeros(size(t)),  % initializare matrice coef. w2 mediu
gamaf=zeros(size(t)), % initializare matrice gama mediu
t0=clock,
for k=1:n
    v1=sqrt(var1)*randn(size(t))+media;    % generare semnal v1
    v2=sqrt(var2)*randn(size(t))+media;    % generare semnal v2
    d=zeros(size(t)),    % initializare matrice semnal dorit
    x=zeros(size(t)),    % initializare matrice semnal iesire canal
    u=zeros(size(t)),    % initializare matrice semnal iesire canal + zgomot
    w1=zeros(size(t)),   % initializare matrice coeficient 1 predictie
    w2=zeros(size(t)),   % initializare matrice coeficient 2 predictie
    y=zeros(size(t)),    % initializare matrice iesire filtru predictie
    sec1=zeros(size(t)); % initializare matrice 1 produs eroare*date
    sec2=zeros(size(t)); % initializare matrice 2 produs eroare*date
    med1=zeros(size(lung)); % initializare matrice fereastră 1
    med2=zeros(size(lung)); % initializare matrice fereastră 2
    fer=zeros(size(lung)); % initializare matrice fereastră semnal
    gama=zeros(size(t));  % initializare matrice gama
        d(1)=v1(1);    %
        for i=2:1000    % generare semnal
            d(i)=v1(i)-0.7943*d(i-1); % dorit
        end    %
        x(1)=d(1);    %
        for i=2:1000    % generare semnal
            x(i)=d(i)+0.9443*x(i-1); % iesire canal
        end    %
        for i=1:1000    % generare semnal
            u(i)=x(i)+v2(i); % iesire canal +
        end    % zgomot

u(400)=u(400)+1;
u(600)=u(600)+2;
u(800)=u(800)+3;

```

```

for i=1:lung %
    fer(i)=0; %
end %
fer(1)=u(1); %
gama(1)=0; %
gamaf(1)=gamaf(1)+gama(1); %
for i=2:1000 % calcul
    p=i-lung*floor((i-1)/lung); % coeficient
    fer(p)=u(i); % gama2
    snr=max(fer)-min(fer); %
    snum=0; %
    for k=1:lung %
        snum=snum+abs(fer(k)-median(fer)); %
    end %
    gama(i)=snr/snum; %
    gamaf(i)=gamaf(i)+gama(i); %
end %
prag=mean(gama)+0.1*(max(gama)-mean(gama)); %

w1(1)=0; %
wf1(1)=wf1(1)+w1(1); %
w2(1)=0; %
wf2(1)=wf1(1)+w1(1); %
y(1)=w1(1)*u(1); %
e(1)=d(1)-y(1); %
med1(1)=e(1)*u(1); %
med2(1)=0; %
er(1)=e(1)*e(1)+er(1); %
w1(2)=w1(1); %
wf1(2)=wf1(2)+w1(2); %
w2(2)=w2(1); % calcul coeficienti
wf2(2)=wf2(2)+w2(2); % de predictie
y(2)=w1(2)*u(2)+w2(2)*u(1); % si eroare
e(2)=d(2)-y(2); % medie patratica
med1(2)=e(2)*u(2); %
med2(2)=e(2)*u(1); %
er(2)=e(2)*e(2)+er(1); %
for i=3:1000 %
    if gama(i)>prag %
        w1(i)=w1(i-1)+m*median(med1); %
        w2(i)=w2(i-1)+m*median(med2); %
    else %
        w1(i)=w1(i-1)+m*mean(med1); %
        w2(i)=w2(i-1)+m*mean(med2); %
    end %
    wf1(i)=wf1(i)+w1(i); %

```

```

    wf2(i)=wf2(i)+w2(i);           %
    p=i-lung*floor((i-1)/lung);   %
    y(i)=w1(i)*u(i)+w2(i)*u(i-1); %
    e(i)=d(i)-y(i);               %
    er(i)=e(i)*e(i)+er(i);        %
    med1(p)=e(i)*u(i);            %
    med2(p)=e(i)*u(i-1);          %
    end                             %
end
    for i=1:1000                   %
        er(i)=er(i)/n;             % calcul medie coeficienti
        wf1(i)=wf1(i)/n;          % si medie eroare
        wf2(i)=wf2(i)/n;          %
        gamaf(i)=gamaf(i)/n;      %
    end                             %
durata=etime(clock,t0)
save gamal dat gamaf -ascii
save wf1l dat wf1 -ascii
save wf2l dat wf2 -ascii
subplot(3,1,1),
plot(t,wf1),
title('Variatia coeficientului de predictie w1');
xlabel('Numar de esantioane');
ylabel('Coeficient w1');
grid,
subplot(3,1,2);
plot(t,wf2);
title('Variatia coeficientului de predictie w2');
xlabel('Numar de esantioane');
ylabel('Coeficient w2');
grid,
subplot(3,1,3),
plot(t,gamaf),
title('Variatia coeficientului gamal');
xlabel('Numar de esantioane');
ylabel('Gamal');
grid,

```

```

function gradd=filtru
m=0.05;           % parametru de ajustare
media=0;         % media semnalelor zgomot
var=0.1;         % varianta zgomotului v
n=5000;         % numar de esantioane
t=1:1:n;         %
lung=9;         % lungimea ferestrei
x=zeros(size(t)); % initializare matrice semnal intrare sistem
d=zeros(size(t)); % initializare matrice semnal dorit (iesire sistem)
a1=zeros(size(t)); % initializare matrice coeficient a1
a2=zeros(size(t)); % initializare matrice coeficient a2
b0=zeros(size(t)); % initializare matrice coeficient b0
b1=zeros(size(t)); % initializare matrice coeficient b1
b2=zeros(size(t)); % initializare matrice coeficient b2
meda1=zeros(size(lung)); % initializare matrice fereastră a1
meda2=zeros(size(lung)); % initializare matrice fereastră a2
medb0=zeros(size(lung)); % initializare matrice fereastră b0
medb1=zeros(size(lung)); % initializare matrice fereastră b1
medb2=zeros(size(lung)); % initializare matrice fereastră b2
e=zeros(size(t)); % initializare matrice eroare
y=zeros(size(t)); % initializare matrice iesire filtru
x=sqrt(var)*randn(size(t))+media;
b0c=-0.1;
b1c=-1.3;
b2c=1.6;
a1c=-1.19;
a2c=0.7;
d(1)=b0c*x(1); %
d(2)=b0c*x(2)+b1c*x(1)-a1c*d(1); % calcul
for i=3:n % semnal
    d(i)=b0c*x(i)+b1c*x(i-1)+b2c*x(i-2)-a1c*d(i-1)-a2c*d(i-2); % dorit
end
d(2000)=d(2000)+2;
d(3000)=d(3000)+3;
d(4000)=d(4000)+4;
a1(1)=0;
a2(1)=0;
b0(1)=0;
b1(1)=0;
b2(1)=0;
y(1)=b0(1)*x(1);
e(1)=d(1)-y(1);
meda1(1)=0;
meda2(1)=0;
medb0(1)=e(1)*x(1);
medb1(1)=0;
medb2(1)=0;
a1(2)=a1(1);
a2(2)=a2(1);

```



```

b0(2)=b0(1)+m*median(medb0);
b1(2)=b1(1);
b2(2)=b2(1);
y(2)=b0(2)*x(2)+b1(2)*x(1)-a1(2)*y(1);
e(2)=d(2)-y(2);
meda1(2)=-e(2)*y(1);
meda2(2)=0;
medb0(2)=e(2)*x(2);
medb1(2)=e(2)*x(1);
medb2(2)=0;
a1(3)=a1(2)+m*median(meda1);
a2(3)=a2(2);
b0(3)=b0(2)+m*median(medb0);
b1(3)=b1(2)+m*median(medb1);
b2(3)=b2(2);
y(3)=b0(3)*x(3)+b1(3)*x(2)+b2(3)*x(1)-a1(3)*y(2)-a2(3)*y(1);
e(3)=d(3)-y(3);
meda1(3)=-e(3)*y(2);
meda2(3)=-e(3)*y(1);
medb0(3)=e(3)*x(3);
medb1(3)=e(3)*x(2);
medb2(3)=e(3)*x(1);
for i=4:n
    a1(i)=a1(i-1)+m*median(meda1);
    a2(i)=a2(i-1)+m*median(meda2);
    b0(i)=b0(i-1)+m*median(medb0);
    b1(i)=b1(i-1)+m*median(medb1);
    b2(i)=b2(i-1)+m*median(medb2);
    y(i)=b0(i)*x(i)+b1(i)*x(i-1)+b2(i)*x(i-2)-a1(i)*y(i-1)-a2(i)*y(i-2);
    e(i)=d(i)-y(i);
    p=i-lung*floor((i-1)/lung);
    meda1(p)=-e(i)*y(i-1);
    meda2(p)=-e(i)*y(i-2);
    medb0(p)=e(i)*x(i);
    medb1(p)=e(i)*x(i-1);
    medb2(p)=e(i)*x(i-2);
end
subplot(1,1,1);
plot(t,a1,'y',t,a2,'g',t,b0,'m',t,b1,'b',t,b2,'r');
title('Variatia coeficientilor filtrului Median-IIR');
xlabel('Numar de esantioane');
ylabel('coef. ');
grid;

```

```
function gradl=filtru
```

```

m=0.05;           % parametru de ajustare
media=0;          % media semnalelor zgomot
var=0.1;          % varianta zgomotului v
n=5000;           % numar de esantioane
t=1:1:n;          %
lung=9;           % lungimea ferestrei

x=zeros(size(t));           % initializare matrice semnal intrare sistem
d=zeros(size(t));           % initializare matrice semnal dorit (iesire sistem)
a1=zeros(size(t));          % initializare matrice coeficient a1
a2=zeros(size(t));          % initializare matrice coeficient a2
b0=zeros(size(t));          % initializare matrice coeficient b0
b1=zeros(size(t));          % initializare matrice coeficient b1
b2=zeros(size(t));          % initializare matrice coeficient b2
k1=zeros(size(t));          % initializare matrice coeficient k1
k2=zeros(size(t));          % initializare matrice coeficient k2
medk1=zeros(size(lung));    % initializare matrice fereastră k1
medk2=zeros(size(lung));    % initializare matrice fereastră k2
medb0=zeros(size(lung));    % initializare matrice fereastră b0
medb1=zeros(size(lung));    % initializare matrice fereastră b1
medb2=zeros(size(lung));    % initializare matrice fereastră b2
e=zeros(size(t));           % initializare matrice eroare
y=zeros(size(t));           % initializare matrice iesire filtru

x=sqrt(var)*randn(size(t))+media;

b0c=-0.1;
b1c=-1.3;
b2c=1.6;
a1c=-1.19;
a2c=0.7;

d(1)=b0c*x(1);              %
d(2)=b0c*x(2)+b1c*x(1)-a1c*d(1); % calcul
for i=3:n                    % semnal
    d(i)=b0c*x(i)+b1c*x(i-1)+b2c*x(i-2)-a1c*d(i-1)-a2c*d(i-2); % dorit
end

d(2000)=d(2000)+2;
d(3000)=d(3000)+3;
d(4000)=d(4000)+4;

k1(1)=0;
k2(1)=0;
a1(1)=k1(1)*(1+k2(1));
a2(1)=k2(1);

```

```

b0(1)=0,
b1(1)=0;
b2(1)=0,
y(1)=b0(1)*x(1),
e(1)=d(1)-y(1),
medk1(1)=0;
medk2(1)=0;
medb0(1)=e(1)*x(1);
medb1(1)=0;
medb2(1)=0;

k1(2)=k1(1);
k2(2)=k2(1);
a1(2)=k1(2)*(1+k2(2));
a2(2)=k2(2);
b0(2)=b0(1)+m*median(medb0);
b1(2)=b1(1);
b2(2)=b2(1);
y(2)=b0(2)*x(2)+b1(2)*x(1)-a1(2)*y(1);
e(2)=d(2)-y(2);
medk1(2)=-e(2)*y(1);
medk2(2)=0;
medb0(2)=e(2)*x(2);
medb1(2)=e(2)*x(1);
medb2(2)=0;

k1(3)=k1(2)+m*median(medk1);
k2(3)=k2(2);
a1(3)=k1(3)*(1+k2(3));
a2(3)=k2(3);
b0(3)=b0(2)+m*median(medb0);
b1(3)=b1(2)+m*median(medb1);
b2(3)=b2(2);
y(3)=b0(3)*x(3)+b1(3)*x(2)-a1(3)*y(2)-a2(3)*y(1);
e(3)=d(3)-y(3);
medk1(3)=-e(3)*y(2);
medk2(3)=-e(3)*y(1);
medb0(3)=e(3)*x(3);
medb1(3)=e(3)*x(2);
medb2(3)=e(3)*x(1);

for i=4:n
    k1(i)=k1(i-1)+m*median(medk1);
    k2(i)=k2(i-1)+m*median(medk2);
    a1(i)=k1(i)*(1+k2(i));
    a2(i)=k2(i);
    b0(i)=b0(i-1)+m*median(medb0);
    b1(i)=b1(i-1)+m*median(medb1);
    b2(i)=b2(i-1)+m*median(medb2);

```

```
y(i)=b0(i)*x(i)+b1(i)*x(i-1)+b2(i)*x(i-2)-a1(i)*y(i-1)-a2(i)*y(i-2);
e(i)=d(i)-y(i);
p=i-lung*floor((i-1)/lung);
medk1(p)=-e(i)*y(i-1);
medk2(p)=-e(i)*y(i-2);
medb0(p)=e(i)*x(i);
medb1(p)=e(i)*x(i-1);
medb2(p)=e(i)*x(i-2);
```

end

```
subplot(1,1,1);
plot(t,a1,'y',t,a2,'g',t,b0,'m',t,b1,'b',t,b2,'r');
title('Variatia coeficientilor filtrului Median-IIR');
xlabel('Numar de esantioane');
ylabel('coef.');
```

grid;

```

function sharfd=filtru
m=0.05,           % parametru de ajustare
media=0,          % media semnalelor zgomot
var=0.1,          % varianta zgomotului v
n=5000,          % numar de esantioane
t=1:n;           %
lung=9,          % lungimea ferestrei
x=zeros(size(t)), % initializare matrice semnal intrare sistem
d=zeros(size(t)), % initializare matrice semnal dorit (iesire sistem)
a1=zeros(size(t)), % initializare matrice coeficient a1
a2=zeros(size(t)), % initializare matrice coeficient a2
b0=zeros(size(t)), % initializare matrice coeficient b0
b1=zeros(size(t)), % initializare matrice coeficient b1
b2=zeros(size(t)), % initializare matrice coeficient b2
meda1=zeros(size(lung)), % initializare matrice fereastră a1
meda2=zeros(size(lung)), % initializare matrice fereastră a2
medb0=zeros(size(lung)), % initializare matrice fereastră b0
medb1=zeros(size(lung)), % initializare matrice fereastră b1
medb2=zeros(size(lung)), % initializare matrice fereastră b2
e=zeros(size(t)), % initializare matrice eroare
c=zeros(size(t)), % initializare matrice coef. ponderare
y=zeros(size(t)); % initializare matrice iesire filtru
x=sqrt(var)*randn(size(t))+media;
b0c=-0.1,
b1c=-1.3,
b2c=1.6;
a1c=-1.19,
a2c=0.7;
d(1)=b0c*x(1), %
d(2)=b0c*x(2)+b1c*x(1)-a1c*d(1); % calcul
for i=3:n % semnal
    d(i)=b0c*x(i)+b1c*x(i-1)+b2c*x(i-2)-a1c*d(i-1)-a2c*d(i-2); % dorit
end
d(2000)=d(2000)+2,
d(3000)=d(3000)+3;
d(4000)=d(4000)+4,
a1(1)=0,
a2(1)=0,
b0(1)=0,
b1(1)=0,
b2(1)=0,
y(1)=b0(1)*x(1);
e(1)=d(1)-y(1);
c(1)=e(1);
meda1(1)=0;
meda2(1)=0;
medb0(1)=c(1)*x(1);
medb1(1)=0;
medb2(1)=0;

```

```

a1(2)=a1(1);
a2(2)=a2(1);
b0(2)=b0(1)+m*median(medb0);
b1(2)=b1(1);
b2(2)=b2(1);
y(2)=b0(2)*x(2)+b1(2)*x(1)-a1(2)*y(1);
e(2)=d(2)-y(2);
c(2)=e(2)-0.6*e(1);
meda1(2)=-c(2)*y(1);
meda2(2)=0;
medb0(2)=c(2)*x(2);
medb1(2)=c(2)*x(1);
medb2(2)=0;
a1(3)=a1(2)+m*median(meda1);
a2(3)=a2(2);
b0(3)=b0(2)+m*median(medb0);
b1(3)=b1(2)+m*median(medb1);
b2(3)=b2(2);
y(3)=b0(3)*x(3)+b1(3)*x(2)+b2(3)*x(1)-a1(3)*y(2)-a2(3)*y(1);
e(3)=d(3)-y(3);
c(3)=e(3)-0.6*e(2);
meda1(3)=-c(3)*y(2);
meda2(3)=-c(3)*y(1);
medb0(3)=c(3)*x(3);
medb1(3)=c(3)*x(2);
medb2(3)=c(3)*x(1);
for i=4:n
    a1(i)=a1(i-1)+m*median(meda1);
    a2(i)=a2(i-1)+m*median(meda2);
    b0(i)=b0(i-1)+m*median(medb0);
    b1(i)=b1(i-1)+m*median(medb1);
    b2(i)=b2(i-1)+m*median(medb2);
    y(i)=b0(i)*x(i)+b1(i)*x(i-1)+b2(i)*x(i-2)-a1(i)*y(i-1)-a2(i)*y(i-2);
    e(i)=d(i)-y(i);
    c(i)=e(i)-0.6*e(i-1);
    p=i-lung*floor((i-1)/lung);
    meda1(p)=-c(i)*y(i-1);
    meda2(p)=-c(i)*y(i-2);
    medb0(p)=c(i)*x(i);
    medb1(p)=c(i)*x(i-1);
    medb2(p)=c(i)*x(i-2);
end

subplot(1,1,1);
plot(t,a1,'y',t,a2,'g',t,b0,'m',t,b1,'b',t,b2,'r');
title('Variatia coeficientilor filtrului Median-IIR');
xlabel('Numar de esantioane');
ylabel('coef. ');
grid;

```

```
function sharfl=filtu
```

```

m=0.05,           % parametru de ajustare
media=0,          % media semnalelor zgomot
var=0.1,         % varianta zgomotului v
n=5000,          % numar de esantioane
t=1:n,           %
lung=9;          % lungimea ferestrei

x=zeros(size(t)),      % initializare matrice semnal intrare sistem
d=zeros(size(t)),      % initializare matrice semnal dorit (iesire sistem)
a1=zeros(size(t)),     % initializare matrice coeficient a1
a2=zeros(size(t)),     % initializare matrice coeficient a2
b0=zeros(size(t)),     % initializare matrice coeficient b0
b1=zeros(size(t)),     % initializare matrice coeficient b1
b2=zeros(size(t)),     % initializare matrice coeficient b2
k1=zeros(size(t)),     % initializare matrice coeficient k1
k2=zeros(size(t)),     % initializare matrice coeficient k2
medk1=zeros(size(lung)), % initializare matrice fereastră k1
medk2=zeros(size(lung)), % initializare matrice fereastră k2
medb0=zeros(size(lung)), % initializare matrice fereastră b0
medb1=zeros(size(lung)), % initializare matrice fereastră b1
medb2=zeros(size(lung)), % initializare matrice fereastră b2
e=zeros(size(t)),      % initializare matrice eroare
c=zeros(size(t)),      % initializare coeficient ponderare
y=zeros(size(t)),      % initializare matrice iesire filtru

x=sqrt(var)*randn(size(t))+media;

b0c=-0.1,
b1c=-1.3,
b2c=1.6,
a1c=-1.19,
a2c=0.7,

d(1)=b0c*x(1);
d(2)=b0c*x(2)+b1c*x(1)-a1c*d(1);
for i=3:n
    d(i)=b0c*x(i)+b1c*x(i-1)+b2c*x(i-2)-a1c*d(i-1)-a2c*d(i-2);
end

d(2000)=d(2000)+2;
d(3000)=d(3000)+3;
d(4000)=d(4000)+4;

k1(1)=0,
k2(1)=0,
a1(1)=k1(1)*(1+k2(1));

```

```

a2(1)=k2(1);
b0(1)=0;
b1(1)=0;
b2(1)=0;
y(1)=b0(1)*x(1);
e(1)=d(1)-y(1);
c(1)=e(1);
medk1(1)=0;
medk2(1)=0;
medb0(1)=c(1)*x(1);
medb1(1)=0;
medb2(1)=0;

k1(2)=k1(1);
k2(2)=k2(1);
a1(2)=k1(2)*(1+k2(2));
a2(2)=k2(2);
b0(2)=b0(1)+m*median(medb0);
b1(2)=b1(1);
b2(2)=b2(1);
y(2)=b0(2)*x(2)+b1(2)*x(1)-a1(2)*y(1);
e(2)=d(2)-y(2);
c(2)=e(2)-0.6*e(1);
medk1(2)=-c(2)*y(1);
medk2(2)=0;
medb0(2)=c(2)*x(2);
medb1(2)=c(2)*x(1);
medb2(2)=0;

k1(3)=k1(2)+m*median(medk1);
k2(3)=k2(2);
a1(3)=k1(3)*(1+k2(3));
a2(3)=k2(3);
b0(3)=b0(2)+m*median(medb0);
b1(3)=b1(2)+m*median(medb1);
b2(3)=b2(2);
y(3)=b0(3)*x(3)+b1(3)*x(2)-a1(3)*y(2)-a2(3)*y(1);
e(3)=d(3)-y(3);
c(3)=e(3)-0.6*e(2);
medk1(3)=-c(3)*y(2);
medk2(3)=-c(3)*y(1);
medb0(3)=c(3)*x(3);
medb1(3)=c(3)*x(2);
medb2(3)=c(3)*x(1);
for i=4:n
    k1(i)=k1(i-1)+m*median(medk1);
    k2(i)=k2(i-1)+m*median(medk2);
    a1(i)=k1(i)*(1+k2(i));
    a2(i)=k2(i);
    b0(i)=b0(i-1)+m*median(medb0);

```



```

b1(i)=b1(i-1)+m*median(medb1);
b2(i)=b2(i-1)+m*median(medb2);
y(i)=b0(i)*x(i)+b1(i)*x(i-1)+b2(i)*x(i-2)-a1(i)*y(i-1)-a2(i)*y(i-2);
e(i)=d(i)-y(i);
c(i)=e(i)-0.6*e(i-1);
p=i-lung*floor((i-1)/lung);
medk1(p)=-c(i)*y(i-1);
medk2(p)=-c(i)*y(i-2);
medb0(p)=c(i)*x(i);
medb1(p)=c(i)*x(i-1);
medb2(p)=c(i)*x(i-2);
end

subplot(1,1,1);
plot(t,a1,'y',t,a2,'g',t,b0,'m',t,b1,'b',t,b2,'r');
title('Variatia coeficientilor filtrului Median-IIR');
xlabel('Numar de esantioane');
ylabel('coef');
grid,

```

```
function smd=filtru
```

```

m=0.05;           % parametru de ajustare
media=0;          % media semnalelor zgomot
var=0.1;          % varianta zgomotului v
n=5000;           % numar de esantioane
t=1:1:n;          %
lung=3;           % lungimea ferestrei

x=zeros(size(t)); % initializare matrice semnal intrare sistem
d=zeros(size(t)); % initializare matrice semnal dorit (iesire sistem)
a1=zeros(size(t)); % initializare matrice coeficient a1
a2=zeros(size(t)); % initializare matrice coeficient a2
b0=zeros(size(t)); % initializare matrice coeficient b0
b1=zeros(size(t)); % initializare matrice coeficient b1
b2=zeros(size(t)); % initializare matrice coeficient b2
meda1=zeros(size(lung)); % initializare matrice fereastră a1
meda2=zeros(size(lung)); % initializare matrice fereastră a2
medb0=zeros(size(lung)); % initializare matrice fereastră b0
medb1=zeros(size(lung)); % initializare matrice fereastră b1
medb2=zeros(size(lung)); % initializare matrice fereastră b2
y=zeros(size(t)); % initializare matrice iesire filtru
x=sqrt(var)*randn(size(t))+media;
b0c=-0.1;
b1c=-1.3;
b2c=1.6;
a1c=-1.19;
a2c=0.7;
d(1)=b0c*x(1); %
d(2)=b0c*x(2)+b1c*x(1)-a1c*d(1); % calcul
for i=3:n % semnal
    d(i)=b0c*x(i)+b1c*x(i-1)+b2c*x(i-2)-a1c*d(i-1)-a2c*d(i-2); % dorit
end

a1(1)=0;
a2(1)=0;
b0(1)=0;
b1(1)=0;
b2(1)=0;
y(1)=b0(1)*x(1);
e(1)=d(1)-y(1);
meda1(1)=0;
meda2(1)=0;
medb0(1)=e(1)*x(1);
medb1(1)=0;
medb2(1)=0;
a1(2)=a1(1);
a2(2)=a2(1);
b0(2)=b0(1)+m*median(medb0);

```

```

b1(2)=b1(1);
b2(2)=b2(1);
y(2)=b0(2)*x(2)+b1(2)*x(1)-a1(2)*y(1);
e(2)=d(2)-y(2);
meda1(2)=-e(2)*d(1);
meda2(2)=0;
medb0(2)=e(2)*x(2);
medb1(2)=e(2)*x(1);
medb2(2)=0;
a1(3)=a1(2)+m*median(meda1);
a2(3)=a2(2);
b0(3)=b0(2)+m*median(medb0);
b1(3)=b1(2)+m*median(medb1);
b2(3)=b2(2);
y(3)=b0(3)*x(3)+b1(3)*x(2)+b2(3)*x(1)-a1(3)*y(2)-a2(3)*y(1);
e(3)=d(3)-y(3);
meda1(3)=-e(3)*d(2);
meda2(3)=-e(3)*d(1);
medb0(3)=e(3)*x(3);
medb1(3)=e(3)*x(2);
medb2(3)=e(3)*x(1);
for i=4:n
    a1(i)=a1(i-1)+m*median(meda1);
    a2(i)=a2(i-1)+m*median(meda2);
    b0(i)=b0(i-1)+m*median(medb0);
    b1(i)=b1(i-1)+m*median(medb1);
    b2(i)=b2(i-1)+m*median(medb2);
    y(i)=b0(i)*x(i)+b1(i)*x(i-1)+b2(i)*x(i-2)-a1(i)*y(i-1)-a2(i)*y(i-2);
    e(i)=d(i)-y(i);
    p=i-lung*floor((i-1)/lung);
    meda1(p)=-e(i)*d(i-1);
    meda2(p)=-e(i)*d(i-2);
    medb0(p)=e(i)*x(i);
    medb1(p)=e(i)*x(i-1);
    medb2(p)=e(i)*x(i-2);
end
subplot(1,1,1);
plot(t,a1,'y',t,a2,'g',t,b0,'m',t,b1,'b',t,b2,'r');
title('Variatia coeficientilor filtrului');
xlabel('Numar de esantioane');
ylabel('coef. ');
grid,

```

```
function sml=filtru
```

```

m=0.05;           % parametru de ajustare
media=0;          % media semnalelor zgomot
var=0.1;          % varianta zgomotului v
n=5000;           % numar de esantioane
t=1:1:n;          %
lung=9;           % lungimea ferestrei

x=zeros(size(t)); % initializare matrice semnal intrare sistem
d=zeros(size(t)); % initializare matrice semnal dorit (iesire sistem)
a1=zeros(size(t)); % initializare matrice coeficient a1
a2=zeros(size(t)); % initializare matrice coeficient a2
b0=zeros(size(t)); % initializare matrice coeficient b0
b1=zeros(size(t)); % initializare matrice coeficient b1
b2=zeros(size(t)); % initializare matrice coeficient b2
k1=zeros(size(t)); % initializare matrice coeficient k1
k2=zeros(size(t)); % initializare matrice coeficient k2
medk1=zeros(size(lung)); % initializare matrice fereastră k1
medk2=zeros(size(lung)); % initializare matrice fereastră k2
medb0=zeros(size(lung)); % initializare matrice fereastră b0
medb1=zeros(size(lung)); % initializare matrice fereastră b1
medb2=zeros(size(lung)); % initializare matrice fereastră b2
e=zeros(size(t)); % initializare matrice eroare
y=zeros(size(t)); % initializare matrice iesire filtru

x=sqrt(var)*randn(size(t))+media;

b0c=-0.1;
b1c=-1.3;
b2c=1.6;
a1c=-1.19;
a2c=0.7;

d(1)=b0c*x(1); %
d(2)=b0c*x(2)+b1c*x(1)-a1c*d(1); % calcul
for i=3:n % semnal
    d(i)=b0c*x(i)+b1c*x(i-1)+b2c*x(i-2)-a1c*d(i-1)-a2c*d(i-2); % dorit
end
d(2000)=d(2000)+2;
d(3000)=d(3000)+3;
d(4000)=d(4000)+4;

k1(1)=0;
k2(1)=0;
a1(1)=k1(1)*(1+k2(1));
a2(1)=k2(1);
b0(1)=0;
b1(1)=0;

```

```

b2(1)=0;
y(1)=b0(1)*x(1);
e(1)=d(1)-y(1);
medk1(1)=0;
medk2(1)=0;
medb0(1)=e(1)*x(1);
medb1(1)=0;
medb2(1)=0;

k1(2)=k1(1);
k2(2)=k2(1);
a1(2)=k1(2)*(1+k2(2));
a2(2)=k2(2);
b0(2)=b0(1)+m*median(medb0);
b1(2)=b1(1);
b2(2)=b2(1);
y(2)=b0(2)*x(2)+b1(2)*x(1)-a1(2)*y(1);
e(2)=d(2)-y(2);
medk1(2)=-e(2)*y(1);
medk2(2)=0;
medb0(2)=e(2)*x(2);
medb1(2)=e(2)*x(1);
medb2(2)=0;

k1(3)=k1(2)+m*median(medk1);
k2(3)=k2(2);
a1(3)=k1(3)*(1+k2(3));
a2(3)=k2(3);
b0(3)=b0(2)+m*median(medb0);
b1(3)=b1(2)+m*median(medb1);
b2(3)=b2(2);
y(3)=b0(3)*x(3)+b1(3)*x(2)-a1(3)*y(2)-a2(3)*y(1);
e(3)=d(3)-y(3);
medk1(3)=-e(3)*y(2);
medk2(3)=-e(3)*y(1);
medb0(3)=e(3)*x(3);
medb1(3)=e(3)*x(2);
medb2(3)=e(3)*x(1);

for i=4:n
    k1(i)=k1(i-1)+m*median(medk1);
    k2(i)=k2(i-1)+m*median(medk2);
    a1(i)=k1(i)*(1+k2(i));
    a2(i)=k2(i);
    b0(i)=b0(i-1)+m*median(medb0);
    b1(i)=b1(i-1)+m*median(medb1);
    b2(i)=b2(i-1)+m*median(medb2);
    y(i)=b0(i)*x(i)+b1(i)*x(i-1)+b2(i)*x(i-2)-a1(i)*y(i-1)-a2(i)*y(i-2);
    e(i)=d(i)-y(i);
    p=i-lung*floor((i-1)/lung);

```

```
medk1(p)=-e(i)*y(i-1);
medk2(p)=-e(i)*y(i-2);
medb0(p)=e(i)*x(i);
medb1(p)=e(i)*x(i-1);
medb2(p)=e(i)*x(i-2);
end

subplot(1,1,1);
plot(t,a1,'y',t,a2,'g',t,b0,'m',t,b1,'b',t,b2,'r');
title('Variatia coeficientilor filtrului IIR');
xlabel('Numar de esantioane');
ylabel('coef. ');
grid;
```

```
function id_on_line=filtrul

k=5,          % constanta de proportionalitate
timp=7;       % constanta de timp
tau=4;        % timp mort
lung=50,      % lungime impuls
timptot=100, % timp test
te=0.1;       % perioada de esantionare
n=timptot/te, % numar de esantioane
xc=1;         % valoare de regim permanent marime de intrare
t=1:1:n,      %

t0=clock,

x=zeros(size(t)), % initializare matrice semnal intrare
y=zeros(size(t)), % initializare matrice semnal iesire

    for i=1:lung/te
        x(i)=1+xc;
    end
    for i=lung/te+1:n
        x(i)=xc;
    end
    for i=1:tau/te
        y(i)=k*xc;
    end
    for i=tau/te+1:(lung+tau)/te
        y(i)=k*xc+k*(1-exp(-(i-tau/te)/(timp/te)));
    end
    for i=(lung+tau)/te+1:n
        y(i)=k*xc+(y((lung+tau)/te)-k*xc)*exp(-(i-(lung+tau)/te)/(timp/te));
    end

x1=0;
x2=0;
x3=0;
y1=0;
y2=0;
y3=0,

    for i=1:n
        x1=x1-(x(i)-xc);
        y1=y1-(y(i)-k*xc);
        x2=x2+i*(x(i)-xc)*te;
        y2=y2+i*(y(i)-k*xc)*te;
        x3=x3-i*i*(x(i)-xc)*te*te;
        y3=y3-i*i*(y(i)-k*xc)*te*te;
    end
end
```

```
h1=y1/x1;  
h2=(y2-h1*x2)/x1;  
h3=(y3-2*h2*x2-h1*x3)/x1;  
ampl=h1  
ct=sqrt(h3/h1-(h2/h1)*(h2/h1))  
tm=-h2/h1-ct
```

```
durata=etime(clock,t0)
```

```
axis('equal');  
subplot(2,1,1);  
plot(t,x);  
title('Variatia semnalului de intrare');  
xlabel('Numar de esantioane');  
ylabel('Amplitudine');  
grid;
```

```
subplot(2,1,2);  
plot(t,y);  
title('Variatia semnalului de iesire');  
xlabel('Numar de esantioane');  
ylabel('Amplitudine');  
grid;
```