



UNIVERSITATEA POLITEHNICA DIN BUCUREȘTI



**Școala Doctorală de Electronică, Telecomunicații și
Tehnologia Informației**

Decizie nr. 884 din 20-07-2022

REZUMAT TEZĂ DE DOCTORAT

Alexandru-George RUSU

**ALGORITMI ADAPTIVI CU APLICAȚII ÎN COMUNICAȚIILE
AUDIO ȘI DE VOCE**

**ADAPTIVE ALGORITHMS WITH APPLICATIONS IN AUDIO AND
VOICE COMMUNICATIONS**

COMISIA DE DOCTORAT

Prof. Dr. Ing. Ion MARGHESCU
Univ. Politehnica București

Președinte

Prof. Dr. Ing. Silviu CIOCHINĂ
Univ. Politehnica București

Conducător de Doctorat

Prof. Dr. Ing. Doru-Florin CHIPER
Univ. Tehnică Gheorghe Asachi din Iași

Referent

Prof. Dr. Ing. Dinu COLȚUC
Univ. Valahia din Târgoviște

Referent

**Prof. Dr. Ing. Constantin PALEO-
LOGU**
Univ. Politehnica București

Referent

BUCUREȘTI 2021

Cuprins

1	Introducere în Domeniul Cercetării	1
1.1	Prezentarea domeniului tezei de doctorat	1
1.2	Scopul tezei de doctorat	2
1.3	Conținutul tezei de doctorat	2
2	Introducere în Algoritmii Adaptivi	3
2.1	Modelul Sistemului	3
2.2	Algoritmul Least-Mean-Square (LMS)	4
2.3	Algoritmul LMS Normalizat	4
2.4	Algoritmul Proiecțiilor Afine (APA)	4
2.5	Algoritmul Recursiv în Sensul Celor mai Mici Pătrate (RLS)	5
2.6	Metrici de Performanță	5
3	Algoritmi LMS Optimizați	6
3.1	Principiul de Proiectare	6
3.2	Algoritmul LMS Optimizat (LMSO)	6
3.2.1	Matricea de Autocorelație a Erorii Coeficienților	7
3.2.2	Derivarea Pasului Optim	7
3.2.3	Rezultate Experimentale	7
3.3	Algoritmul cu Pas Diferențiat LMS Optimizat (ODSS-LMS)	7
3.3.1	Matricea de Autocorelație a Erorii Coeficienților	8
3.3.2	Derivarea Vectorului Optim al Pasului	8
3.3.3	Rezultate Experimentale	9
3.4	Considerente Practice	9
3.5	Analiză Complexității de Calcul	9
3.6	Rezumat și Concluzii	9
4	Algoritmi bazați pe Reutilizarea Datelor	11
4.1	Conceptul de Bază	11
4.2	Algoritmul Reutilizării Datelor LMSO (LMSO-DR)	11
4.2.1	Dezvoltarea Algoritmului	11
4.2.2	Rezultate Experimentale	12

4.3	Algoritmul cu Pas Variabil NLMS (VSS-NLMS) bazat pe Reutilizarea Datelor	12
4.3.1	Dezvoltarea Algoritmului	12
4.3.2	Rezultate Experimentale	13
4.4	APA cu Pas Variabil (VSS-APA)	14
4.4.1	Dezvoltarea Algoritmului	14
4.4.2	Rezultate Experimentale	14
4.5	Rezumat și Concluzii	14
5	Filtre Adaptive în Cascadă	16
5.1	Structuri Multiliniare fără Memorie	16
5.2	Structuri Multiliniare cu Memorie	17
5.3	Modelul Sistemului bazat pe Descompunerea Produsului Kronecker . .	18
5.4	Analiza în Transformata Z	18
5.5	Algoritmul NLMS Multiliniar în Cascadă	18
5.5.1	Dezvoltarea Algoritmului	19
5.5.2	Rezultate Experimentale	19
5.6	Algoritmul RLS Multiliniar în Cascadă	19
5.6.1	Dezvoltarea Algoritmului	20
5.6.2	Rezultate Experimentale	20
5.7	Analiza Complexității de Calcul	22
5.8	Rezultate și Concluzii	23
6	Concluzii	24
6.1	Rezultate Obținute	24
6.2	Contribuții Originale	24
6.3	Lista Publicațiilor Originale	25
6.4	Perspective pentru Dezvoltări Viitoare	26
	Bibliografie	27

Capitolul 1

Introducere în Domeniul Cercetării

1.1 Prezentarea domeniului tezei de doctorat

Anularea ecoului reprezintă una dintre cele mai interesante aplicații ale procesării de semnale. Chiar dacă se datorează cuplajului dintre difuzor și microfon a unui dispozitiv real (adică, ecoul acustic) sau este obținut datorită unei nepotriviri într-un circuit electric (adică ecou de linie), reflectând înapoi la sursă o parte a semnalului, una dintre soluțiile principale de anulare a acestuia se bazează pe configurația de identificare a sistemului. În configurația de identificare a sistemului, principiul este de a identifica un sistem cât mai aproape de sistemul care produce semnalul ecou. Prin urmare, chiar dacă sistemul real contribuie la ecou sau poate acustica mediului afectează într-un fel sau altul ecoul (adică, în anularea ecoului acustic), un estimat cât mai precis posibil trebuie obținut astfel încât în final, ecoul să fie eliminat. Cu siguranță, sistemul care trebuie identificat se poate schimba în timp și din această cauză, algoritmul folosit pentru identificarea sistemului este un algoritm adaptiv care este capabil să urmărească comportamentele imprevizibile.

Literatura de specialitate oferă mulți algoritmi adaptivi și chiar dacă caracterul lor adaptiv ar trebui să facă față cu succes oricărei situații, fiecare dintre aceștia este mai eficient în anumite scenarii cu o complexitate de calcul mai ridicată sau mai redusă. Unii algoritmi sunt concepute pur teoretice, utilizați pentru a deriva repere de ultimă generație și reprezintă căi către noi cercetări.

De asemenea, există sisteme care prezintă o complexitate ridicată, sau prin natura lor produc alte efecte precum reverberația, întârzierile sau neliniaritățile. În ultimii ani au fost studiate diferite tehnici, dar nu există o soluție unică convenită, în timp ce aceasta depinde de aplicația în sine și de efectele care ar trebui eliminate sau reduse. Chiar și efectele în sine pot fi de mai multe tipuri, ceea ce face dificilă găsirea unei singure soluții. Cu toate acestea, o abordare larg răspândită se bazează pe un sistem multiliniar care combină filtre adaptive paralele sau în cascadă, liniare sau neliniare [16].

1.2 Scopul tezei de doctorat

Scopul tezei este de a dezvolta algoritmi noi sau variante îmbunătățite ale celor existenți pentru identificarea sistemelor necunoscute prin intermediul filtrelor adaptive, care depășesc soluțiile existente și vizează aplicații în timp real. În plus, teza dezvoltă metode de optimizare care îmbunătățesc performanța generală a algoritmilor adaptivi. În final, cu scopul de a reduce complexitatea de calcul a algoritmilor rudimentari, sunt dezvoltate soluții pentru semnale propagate pe mai multe căi (ex., efectul de reverberație), efect care poate apărea în aplicații practice.

1.3 Conținutul tezei de doctorat

Teza conține patru capitole principale și unul final care cuprinde un rezumat al tezei, lista contribuțiilor și perspective de viitor pentru cercetare și dezvoltare. Primul capitol din cele patru capitole principale constă într-o scurtă introducere în teoria algoritmilor adaptivi ce au fost utilizați ca punct de plecare pentru cercetarea acestei teze. Mai departe, celelalte trei capitolele principale reprezintă contribuțiile principale ale tezei și introduc dezvoltarea teoretică împreună cu rezultatele și concluziile experimentale. Rezultatele sunt prezentate în fiecare capitol de contribuții pentru a facilita o mai bună structură și vizualizare a contextului. În continuare, conținutul capitolelor este rezumat.

În Capitolul 2, prezentăm câțiva dintre cei mai populari algoritmi adaptivi. În Capitolul 3, propunem doi algoritmi noi, care urmează un criteriu de optimizare mai practic pentru aplicațiile din lumea reală. Pentru fiecare algoritm, am dezvoltat o extensie pentru un anumit caz de utilizare, adică pentru cazul în care semnalul de intrare este considerat zgomot alb Gaussian.

În continuare, în Capitolul 4, folosind tehnica de reutilizare a datelor, am urmărit să îmbunătățim performanța unuia dintre algoritmii dezvoltați în Capitolul 3 și să-i menținem complexitatea de calcul liniară. Apoi, am analizat reutilizarea datelor dintr-o perspectivă diferită și am dezvoltat o metodă de generare a pașilor ficși ai algoritmilor NLMS și APA, astfel încât am obținut doi algoritmi noi cu pas variabil.

Mai departe, prin Capitolul 5 introducem mai multe structuri cu și fără memorie în contextul unui sistem cu mai multe intrări și o singură ieșire (MISO). Prin dezvoltarea acestor structuri, am obținut o modalitate de a împărți un filtru de lungime lungă într-o cascadă de filtre de lungime mai scurtă, care formează un sistem cu o singură intrare și o singură ieșire (SISO). Apoi, am prezentat două abordări de utilizare a datelor de intrare pentru a facilita două contexte pentru identificare a sistemelor liniare și multiliniare.

În final, deoarece concluziile au fost prezentate în fiecare capitol, în Capitolul 6 rezumăm doar principalele rezultate. De asemenea, prezentăm principalele contribuții și câteva perspective de viitor pentru cercetare și dezvoltare.

Capitolul 2

Introducere în Algoritmii Adaptivi

Acest capitol reprezintă o prezentare a unora dintre cei mai populari algoritmi adaptivi din literatură, care au sprijinit cercetarea și dezvoltarea ideilor originale și a algoritmilor din această teză.

2.1 Modelul Sistemului

În cadrul identificării unui sistem necunoscut, este folosită configurația prezentată în Fig. 2.1. În această configurație, $x(n)$ reprezintă semnalul de intrare la momentul discret de timp n , în timp ce $y(n)$ este ieșirea sistemului care trebuie determinat (adică, $\mathbf{h}$).

Prin filtrarea semnalului de intrare prin ambele sisteme, putem scrie $\hat{y}(n) = \hat{\mathbf{h}}^T(n-1)\mathbf{x}(n)$ și $y(n) = \mathbf{h}^T\mathbf{x}(n)$, unde $\mathbf{x}(n)$ este un vector care conține cele mai recente L eșantioane de intrare.

În final, ieșirea sistemului adaptiv este scăzută din semnalul dorit, astfel încât semnalul de eroare $e(n)$ rezultă ca $e(n) = d(n) - \hat{y}(n)$.

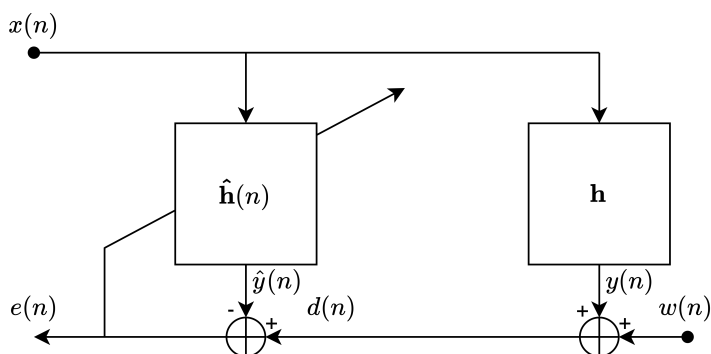


Figura 2.1 Modelul sistemului.

2.2 Algoritmul Least-Mean-Square (LMS)

Filtrul Wiener a facilitat dezvoltarea unuia dintre cei mai populari algoritmi adaptivi, algoritmul LMS. Popularitatea algoritmului LMS se datorează ușurinței de înțelegere, implementare și utilizare a acestuia. La început, filtrul Wiener urmărea să determine un filtru optim, $\hat{\mathbf{h}}_W$, ce minimizează semnalul de eroare în fiecare iterație.

În acest context, filtrul optim care minimizează semnalul de eroare este identificat prin minimizarea unei funcții de cost, eroarea medie pătratică (MSE), definită ca $J[\hat{\mathbf{h}}(n)] = E[e^2(n)]$, unde $E[\cdot]$ reprezintă operatorul de mediere statistică. După mai multe calcule, filtrul Wiener optim se obține ca $\hat{\mathbf{h}}_W = \mathbf{R}_x^{-1} \mathbf{p}_{xd}$.

Calculul vectorului de corelație $\mathbf{p}_{xd}$ și a matricei de autocorelație $\mathbf{R}_x$, implică cunoașterea unor estimări statistice, făcând algoritmul pantei cele mai abrupte inadecvat pentru utilizare practică. Dacă luăm în considerare valorile instantanee

$$\mathbf{p}_{xd}(n) = d(n)\mathbf{x}(n), \quad \mathbf{R}_x(n) = \mathbf{x}(n)\mathbf{x}^T(n), \quad (2.16)$$

rezultă următoarea ecuație

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + 2\mu\mathbf{x}(n)e(n), \quad (2.17)$$

care definește ecuația de actualizare a algoritmului LMS.

2.3 Algoritmul LMS Normalizat

Pentru a oferi un grad mai mare de libertate, o alegere bună ar fi un pas care depinde de timp. O primă posibilitate este de a normaliza pasul μ prin estimarea instantanee a semnalului de intrare și ecuația (2.17) poate fi rescrisă ca

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \frac{\mu}{\mathbf{x}^T(n)\mathbf{x}(n)} \mathbf{x}(n)e(n). \quad (2.20)$$

Ecuația (2.20) definește ecuația de actualizare a algoritmului NLMS.

2.4 Algoritmul Proiecțiilor Afine (APA)

APA vine ca o extensie a algoritmului NLMS. În timp ce algoritmi LMS și NLMS rulează pe un singur vector de intrare, APA gestionează mai mulți vectori de intrare prin matricea de intrare $\mathbf{X}(n)$. Ecuația de actualizare a APA este

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mu\mathbf{X}(n)[\mathbf{X}^T(n)\mathbf{X}(n) + \delta\mathbf{I}_P]^{-1}\mathbf{e}(n), \quad (2.30)$$

unde $\mathbf{I}_P$ este matricea identitate de dimensiune $P \times P$.

2.5 Algoritmul Recursiv în Sensul Celor mai Mici Pătrate (RLS)

Algoritmul RLS reprezintă o referință importantă în teoria filtrării adaptive datorită performanței sale ridicate și a robusteții. De această dată, mediile statistice sunt înlocuite de suma ponderată a valorilor temporale pătrate. Astfel, funcția de cost, în loc de MSE, este înlocuită cu funcția de cost a celor mai mici pătrate (LS), definită ca

$$J[\hat{\mathbf{h}}(n)] = \sum_{k=1}^n \lambda^{n-k} \varepsilon^2(k), \quad (2.31)$$

pentru care, $\varepsilon(k)$ semnifică semnalul de eroare a posteriori, cu $\varepsilon(k) = d(k) - \hat{\mathbf{h}}(n)\mathbf{x}(k)$.

Dezvoltarea funcției de cost are ca rezultat următoarea ecuație recursivă pentru filtrul optim

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mathbf{k}(n)e(n), \quad (2.45)$$

unde $\mathbf{k}(n)$ denotă câștigul Kalman definit ca

$$\mathbf{k}(n) = \frac{\mathbf{R}_x^{-1}(n-1)\mathbf{x}(n)}{\lambda + \mathbf{x}^T(n)\mathbf{R}_x^{-1}(n-1)\mathbf{x}(n)}. \quad (2.46)$$

2.6 Metrice de Performanță

Când măsurăm performanța unui algoritm adaptiv, ne referim la trei caracteristici principale: rata de convergență inițială, dezalinierea coeficienților și capacitatea de urmărire. Rata de convergență descrie cât de repede sunt determinați coeficienții sistemului necunoscut, în timp ce dezalinierea indică cât de aproape este soluția estimată față de coeficienții reali. Cu toate acestea, în unele cazuri, sistemul necunoscut poate varia sau se poate schimba complet (în special în scenarii de anulare a ecoului acustic) și prin capacitatea de urmărire analizăm cât de repede algoritmul adaptiv este capabil să determine coeficienții noului sistem. Toate aceste caracteristici pot fi analizate prin dezalinierea normalizată (NM), definită ca

$$\text{NM}(n) = 10 \log_{10} \frac{\|\mathbf{h} - \hat{\mathbf{h}}(n)\|_2^2}{\|\mathbf{h}\|_2^2} \text{ (dB)}. \quad (2.47)$$

Când coeficienții estimați (adică $\hat{\mathbf{h}}(n)$) sunt apropiați de coeficienții reali (adică $\mathbf{h}$), numitorul pentru NM tinde spre zero și raportul la fel. Prin aceasta, putem observa că valorile logaritmului merg la minus infinit.

Capitolul 3

Algoritmi LMS Optimizați

Acest capitol prezintă prima contribuție la această teză, reprezentată de doi algoritmi cu pas variabil bazați pe algoritmul LMS. Literatura de specialitate este bogată pe acest subiect, propunând diferite soluții, în general, bazate pe considerații euristice. Începând cu [15], au apărut o serie lungă de algoritmi sub denumirea de “algoritmi proporționali” [29], [20], [18]. Cu toate acestea, principiul de dezvoltare care stă în spatele algoritmului LMS (adică, minimizarea funcției de cost MSE) nu este atât de practic în contextul anulării ecoului, în timp ce în aplicațiile din viața reală am dori să obținem o estimare mai bună a semnalului de ecou, ceea ce înseamnă o dezaliniere mai mică între răspunsul la impuls estimat, $\hat{\mathbf{h}}(n)$ și răspunsul la impuls necunoscut, $\mathbf{h}(n)$. Dezvoltarea din acest capitol a fost publicată în [6] și [9].

3.1 Principiul de Proiectare

În acest cadru, considerăm că sistemul necunoscut, $\mathbf{h}(n)$, se schimbă dinamic și este descris de un model Markov de ordinul întâi [14][17], caracterizat prin zgomotul $\delta_{\mathbf{h}}(n)$.

Primul pas în această dezvoltare este înlocuirea pasului fix în ecuația de actualizare a algoritmului LMS cu o matrice de control, $\mathbf{Y}(n)$. Putem remarca că matricea $\mathbf{Y}(n)$ ar putea fi, o matrice unitară scalată, care conduce la un algoritm LMS cu pas variabil, o matrice diagonală, care conduce la un algoritm cu pas diferențial sau o matrice arbitrară. Definim matricea de autocorelație a erorii coeficienților ca $\mathbf{R}_{\mathbf{c}}(n) \triangleq \mathbb{E}[\mathbf{c}(n)\mathbf{c}^T(n)]$, unde $\mathbf{c}(n) = \hat{\mathbf{h}}(n) - \mathbf{h}(n)$, denotă dezalinierea a posteriori. Mai mult, notăm $\mathbf{\Gamma}(n) \triangleq \mathbf{R}_{\mathbf{c}}(n) + \sigma_{\mathbf{h}}^2 \mathbf{I}_L$, unde $\sigma_{\mathbf{h}}^2 = \mathbb{E}[\delta_{\mathbf{h}}^2(n)]$ reprezintă varianța zgomotului sistemului, $\delta_{\mathbf{h}}(n)$.

3.2 Algoritmul LMS Optimizat (LMSO)

În această dezvoltare, considerăm primul caz dintre cele trei prezentate în subcapitolul anterior, când $\mathbf{Y}(n)$ este o matrice unitară scalată, astfel încât $\mathbf{Y}(n) = \mu(n)\mathbf{I}_L$.

3.2.1 Matricea de Autocorelație a Erorii Coeficienților

Începând cu înlocuirea dezalinierii a posteriori în matricea de autocorelație a erorii coeficienților și dezvoltând-o, se obține următorul rezultat

$$\begin{aligned} \mathbf{R}_e(n) = & \mathbf{\Gamma}(n-1) - \mu(n)[\mathbf{\Gamma}(n-1)\mathbf{R}_x + \mathbf{R}_x\mathbf{\Gamma}(n-1)] \\ & + \mu^2(n)\{2\mathbf{R}_x\mathbf{\Gamma}(n-1)\mathbf{R}_x + \mathbf{R}_x\text{tr}[\mathbf{\Gamma}(n-1)\mathbf{R}_x]\} + \mu^2(n)\sigma_w^2\mathbf{R}_x. \end{aligned} \quad (3.13)$$

3.2.2 Derivarea Pasului Optim

Pasul optim este obținut prin calculul minimului MSD (adică $m(n) \triangleq E[\|\mathbf{c}(n)\|_2^2] = \text{tr}[\mathbf{R}_e(n)]$) și obținem

$$\mu(n) = \frac{\text{tr}[\mathbf{\Gamma}(n-1)\mathbf{R}_x]}{\text{tr}\{\mathbf{\Gamma}(n-1)\mathbf{R}_x[L\sigma_x^2\mathbf{I}_L + 2\mathbf{R}_x]\} + L\sigma_x^2\sigma_w^2}. \quad (3.31)$$

Când semnalul de intrare este zgomot alb Gaussian, pasul optim poate fi obținut ca

$$\mu_o(n) = \frac{m_o(n-1) + L\sigma_h^2}{\sigma_x^2[m_o(n-1) + L\sigma_h^2](L+2) + L\sigma_w^2}. \quad (3.34)$$

Putem concluziona că am obținut doi algoritmi, primul care se bazează pe ecuația generală (3.31) numit algoritmul „general” optimizat least-mean-square (LMSO-G), iar al doilea care se bazează pe ecuația simplificată (3.34) numit algoritmul „alb” optimizat least-mean-square (LMSO-W).

3.2.3 Rezultate Experimentale

Simulările au fost efectuate în contextul anulării ecoului acustic. Putem remarca în Figura 3.6 că parametrul σ_h^2 joacă un rol important în guvernarea performanței algoritmilor LMSO-G și LMSO-W. În ceea ce privește capacitatea de urmărire, o valoare mică de σ_h^2 inhibă re-convergența atunci când calea ecoului se schimbă, în timp ce o valoare mai mare permite algoritmilor să determine noul sistem într-un timp rezonabil.

3.3 Algoritmul cu Pas Diferențiat LMS Optimizat (ODSS-LMS)

În această dezvoltare, luăm în considerare al doilea caz dintre cele trei prezentate în Secțiunea 3.1 considerând pasul $\mathbf{Y}(n) = \text{matdiag}[\mathbf{y}(n)]$, unde $\text{matdiag}(\cdot)$ desemnează o matrice diagonală, având pe diagonala principală elementele vectorului $\mathbf{y}(n)$, conducând la un pas diferențiat sau proporțional.

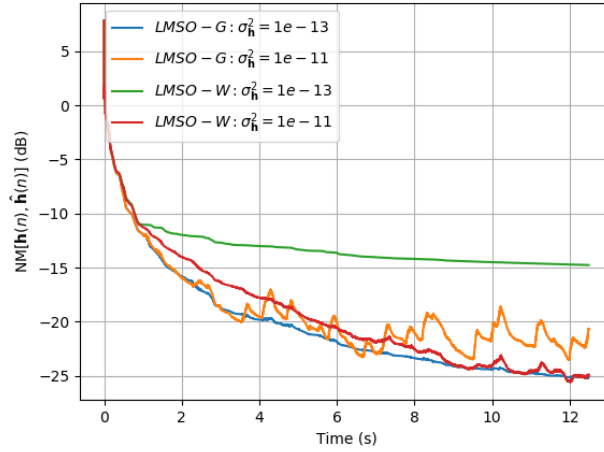


Figura 3.6 Dezalinierea normalizată a algoritmilor LMSO-W și LMSO-G pentru diferite valori constante ale σ_h^2 . Semnalul de intrare este o secvență vocală și $RSZ = 30\text{dB}$.

3.3.1 Matricea de Autocorelație a Erorii Coeficienților

Procedând în același mod ca în secțiunea anterioară, matricea de autocorelație a erorii coeficienților rezultă ca

$$\begin{aligned} \mathbf{R}_c(n) = & \mathbf{\Gamma}(n-1) - \mathbf{\Gamma}(n-1)\mathbf{R}_x\mathbf{Y}(n) - \mathbf{Y}(n)\mathbf{R}_x\mathbf{\Gamma}(n-1) \\ & + \mathbf{Y}(n)\{2\mathbf{R}_x\mathbf{\Gamma}(n-1)\mathbf{R}_x + \mathbf{R}_x\text{tr}[\mathbf{\Gamma}(n-1)\mathbf{R}_x]\}\mathbf{Y}(n) + \mathbf{Y}(n)\mathbf{R}_x\mathbf{Y}(n)\sigma_w^2. \end{aligned} \quad (3.41)$$

3.3.2 Derivarea Vectorului Optim al Pasului

În continuare, obținem pasul optim calculând gradientul MSD în raport cu $\mathbf{y}(n)$, având soluția:

$$\begin{aligned} \mathbf{y}_o(n) = & \left\{ \sigma_x^2 \mathbf{I}_L \{ \text{tr}[\mathbf{\Gamma}(n-1)\mathbf{R}_x] + \sigma_w^2 \} + 2\text{matdiag}[\mathbf{R}_x\mathbf{\Gamma}(n-1)\mathbf{R}_x] \right\}^{-1} \\ & \times \text{diag}[\mathbf{R}_x\mathbf{\Gamma}(n-1)]. \end{aligned} \quad (3.52)$$

Din nou, dacă considerăm că semnalul de intrare este zgomot alb Gaussian și $\mathbf{R}_x = \sigma_x^2 \mathbf{I}_L$, obținem următorul rezultat pentru pas

$$\mathbf{y}_o(n) = \left\{ \sigma_x^2 \mathbf{I}_L [m(n-1) + \sigma_h^2(L+2)] + 2\sigma_x^2 \text{matdiag}[\mathbf{R}_c(n-1)] + \sigma_w^2 \mathbf{I}_L \right\}^{-1} \gamma(n-1), \quad (3.55)$$

cu $\gamma(n) \triangleq \text{diag}[\mathbf{\Gamma}(n)]$.

În acest moment, putem concluziona că am obținut doi algoritmi, primul definit de ecuația (3.52) numit algoritmul optimizat diferențiat least-mean-square “general” (ODSS-LMS-G) și al doilea bazat pe ecuația (3.55) numit algoritmul optimizat diferențiat least-mean-square “alb” (ODSS-LMS-W).

3.3.3 Rezultate Experimentale

Fig. 3.14 evidențiază faptul că ambii algoritmi ODSS-LMS depășesc algoritmi JO-NLMS [27] și IPNLMS [20] în ceea ce privește nivelul de dezaliniere normalizat. Putem remarca performanța bună a algoritmului ODSS-LMS-W care atinge un nivel normalizat de dezaliniere de -30dB în mai puțin de 1 secundă.

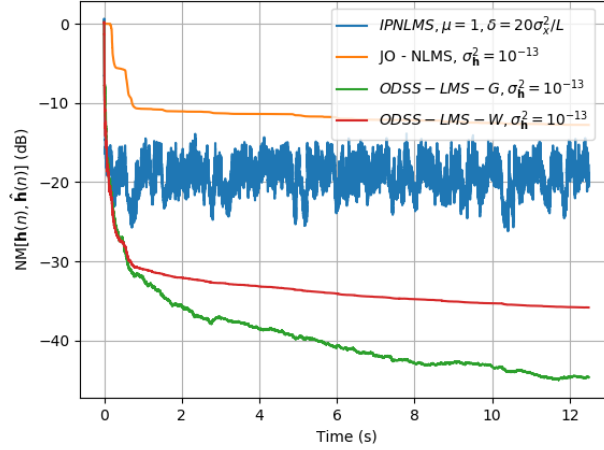


Figura 3.14 Dezalinierea normalizată a algoritmilor IPNLMS, JO-NLMS, ODSS-LMS-G și ODSS-LMS-W. Semnalul de intrare este o secvență de vocală. Sistemul necunoscut este calea ecoului de rețea.

3.4 Considerente Practice

Pe parcursul dezvoltării algoritmilor LMSO și ODSS-LMS, am introdus câțiva parametrii care necesită a fi estimați atunci când ne referim la aplicații în timp real. Primul este puterea semnalului de intrare, σ_x^2 , care poate fi estimată cu ușurință ca $\sigma_x^2 = \frac{1}{L} \mathbf{x}^T(n) \mathbf{x}(n)$, sau printr-un filtru exponențial cu un pol. Apoi, matricea de autocorelație a semnalului de intrare poate fi estimată printr-o fereastră de ponderare $\tilde{\mathbf{R}}_x(n) = \lambda \tilde{\mathbf{R}}_x(n-1) + (1 - \lambda) \mathbf{x}(n) \mathbf{x}^T(n)$, unde λ denotă factorul de uitare. Dacă puterea zgomotului σ_w^2 nu este cunoscută, ar trebui estimată și ea, iar pentru aceasta, ne putem folosi de [13], [25].

3.5 Analiză Complexității de Calcul

Complexitatea de calcul generală a algoritmilor dezvoltați este rezumată în Tabelul 3.1 și Tabelul 3.2.

3.6 Rezumat și Concluzii

Folosind algoritmul LMS ca punct de plecare și luând în considerare minimizarea funcției de cost MSD, am dezvoltat algoritmi LMSO-G și ODSS-LMS-G. Pe baza

Tabel 3.1 Comparația complexității de calcul între algoritmi LMS și LMSO

Operații \ Algoritm	LMS	LMSO-G	LMSO-W
Înmulțiri	$2L + 1$	$4L^3 + 7L^2 + 2L + 4$	$2L + 7$
Adunări	$2L$	$2L^3 + 2L^2 + 3L^2(L - 1) + 5L$	$2L + 3$
Împărțiri	0	1	1

Tabel 3.2 Comparația complexității de calcul între algoritmi LMS și ODSS-LMS

Operații \ Algoritm	LMS	ODSS-LMS-G	ODSS-LMS-W
Înmulțiri	$2L + 1$	$3L^3 + 2L^2 + 16L$	$10L + 3$
Adunări	$2L$	$3L^3 - 2L^2 + 9L$	$10L + 1$
Împărțiri	0	L	L

unui anumit caz de utilizare (adică, considerând că semnalul de intrare este zgomot alb Gaussian), am obținut versiunile simplificate, algoritmi LMSO-W și ODSS-LMS-W. Versiunile generale, LMSO-G și ODSS-LMS-G, prezintă o complexitate de calcul $O(L^3)$, în timp ce versiunile simplificate, LMSO-W și ODSS-LMS-W, sunt caracterizate printr-o complexitate de calcul liniară (adică $O(L)$).

După cum simulările prezintă, algoritmi LMSO-G și ODSS-LMS-G sunt robuști atunci când manipulează semnale de intrare corelate și prezintă performanțe bune în ceea ce privește rata de convergență, dezalinierea normalizată și capacitatea de urmărire. Cu toate acestea, datorită complexității de calcul ridicate, aceștia sunt mai potriviți pentru “benchmarking” decât pentru utilizare practică. Chiar dacă versiunile simplificate au fost dezvoltate luând în considerare zgomotul alb Gaussian ca semnal de intrare, simulările au evidențiat performanțele lor bune chiar și atunci când semnalul de intrare este corelat. Desigur, am remarcat un dezavantaj în performanță pentru algoritmi LMSO-W și ODSS-LMS-W în comparație cu omologii lor LMSO-G și ODSS-LMS-G, dar în schimb, am obținut o reducere semnificativă a complexității de calcul. Astfel, putem sublinia că algoritmi LMSO-W și ODSS-LMS-W sunt candidați potriviți pentru aplicații în timp real.

De asemenea, algoritmi de tip ODSS-LMS au avantajul pasului diferențiat, fiind o alegere bună atunci când calea ecoului este mai rară. A fost introdus un nou parametru, σ_h^2 , iar simulările au prezentat importanța acestuia. Conform simulărilor, parametrul σ_h^2 trebuie reglat astfel încât să se obțină un compromis între nivelul scăzut de dezaliniere normalizat și capacitatea de urmărire.

Capitolul 4

Algoritmi bazați pe Reutilizarea Datelor

4.1 Conceptul de Bază

Inițial, această metodă a fost introdusă în [10]. Metoda reutilizării datelor presupune actualizarea coeficienților la fiecare moment de timp, luând în considerare M eşantioane din trecut, unde M reprezintă ordinul de reutilizare a datelor.

4.2 Algoritmul Reutilizării Datelor LMSO (LMSO-DR)

În capitolul anterior am introdus algoritmi LMSO-G și LMSO-W. S-a observat o reducere drastică în ceea ce privește complexitatea de calcul pentru algoritmul LMSO-W ($O(L)$ comparativ cu $O(L^3)$ pentru algoritmul LMSO-G), care a afectat performanța generală. Folosind o metodă mai robustă de reutilizare a datelor ca în [12], dorim să obținem performanța algoritmului LMSO-G și păstrarea complexității de calcul liniară.

4.2.1 Dezvoltarea Algoritmului

Luând în considerare algoritmul LMSO-W, putem aplica metoda reutilizării datelor și dezvoltăm algoritmul reutilizării datelor LMSO-W (LMSO-W-DR).

Am dori să obținem o ecuație care actualizează coeficienții filtrului într-un singur pas, astfel încât să se evite cele M iterații. Prin urmare, luând în considerare [12] ca referință, putem începe să înlocuim ecuațiile din prima iterație în ecuațiile din cea de-a doua iterație și urmând această procedură până la ordinul M de reutilizare a datelor, două ecuații echivalente rezultă

$$\mathbf{e}(n) = \mathbf{d}(n) - \mathbf{X}^T(n)\hat{\mathbf{h}}(n-1), \quad (4.15)$$

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mathbf{X}(n)\mathbf{Z}(n)\mathbf{e}(n), \quad (4.16)$$

unde $\mathbf{Z}(n)$ este matricea pasului definită ca $\mathbf{Z}(n) = \mathbf{K}^{-1}(n)\mathbf{S}(n)$, cu $\mathbf{S}(n)$, o matrice diagonală $M \times M$, construită cu pașii algoritmului LMSO-W. Se poate observa că termenul $\mathbf{K}(n)$ deschide o conexiune cu APA [22].

4.2.2 Rezultate Experimentale

Sistemul care trebuie identificat este o cale de ecou acustic. În Fig. 4.2 calea de ecou se modifică la mijlocul simulării. Analizăm impactul ordinului de reutilizare a datelor asupra ratei de convergență. După cum era de așteptat, un ordin mai mare de reutilizare a datelor, adică $M = 4$, îmbunătățește rata de convergență și în acest caz, algoritmul LMSO-W-DR este chiar mai rapid decât algoritmul LMSO-G. Se poate remarca că folosind cel mai mic ordin de reutilizare a datelor, adică $M = 2$, rata de convergență este mai rapidă decât cea a algoritmului LMSO-G.

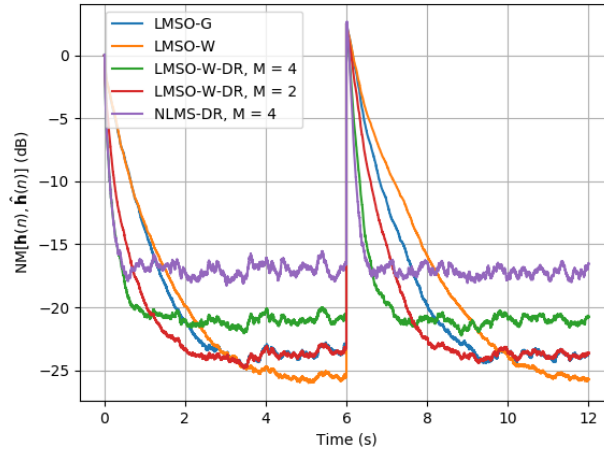


Figura 4.2 Dezalinierea normalizată a algoritmilor LMSO-W-DR, LMSO-W, LMSO-G și NLMS-DR. Pentru algoritmii de tip LMSO $\sigma_h^2 = 1e - 11$, în timp ce pentru algoritmul NLMS-DR $\mu = 1$; calea ecoului se modifică la momentul 5s. Semnalul de intrare este un proces AR(1) ($\beta = 0,9$), $L = 512$ și RSZ = 20dB.

4.3 Algoritmul cu Pas Variabil NLMS (VSS-NLMS) bazat pe Reutilizarea Datelor

Urmând conceptul introdus în Secțiunea 4.1 și luând în considerare algoritmul NLMS prezentat în Secțiunea 2.3 ca model de lucru, dezvoltăm un algoritm NLMS cu pas variabil (VSS-NLMS) [3] pentru care pasul este guvernat printr-un set simplu de reguli.

4.3.1 Dezvoltarea Algoritmului

Mai întâi de toate, să dezvoltăm ecuația (2.20) pentru fiecare iterație de reutilizare a datelor, așa cum am făcut pentru algoritmul LMSO-W-DR, dar ținând cont că datele de

intrare sunt fixe (ex., $\mathbf{x}(n)$, $d(n)$) [12], [19], [26], [21], și analizăm evoluția pasului μ . Prin dezvoltarea ecuațiilor algoritmului NLMS pentru fiecare iterație, înlocuim ecuațiile din prima iterație în ecuațiile din a doua iterație, procedând în aceeași manieră până la ordinul k . Prin urmare, ecuația de actualizare în iterație k este

$$\hat{\mathbf{h}}_k(n) = \hat{\mathbf{h}}(n-1) + \left[1 - (1 - \mu)^{k+1}\right] \frac{\mathbf{x}(n)e(n)}{\mathbf{x}^T(n)\mathbf{x}(n)}, \quad (4.30)$$

cu eroare a posteriori $e_k(n) = (1 - \mu)^k e(n)$. În acest context, ecuația de actualizare a algoritmului VSS-NLMS devine

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \frac{\mu(n)}{\mathbf{x}^T(n)\mathbf{x}(n)} \mathbf{x}(n)e(n). \quad (4.35)$$

Apoi, putem genera un set pași, fiecare fiind folosit pentru un anumit număr de iterații. Pentru a nu bloca algoritmul, putem folosi un mecanism care detectează modificări în sistemul necunoscut, cum ar fi [28] și resetează utilizarea pașilor generați.

4.3.2 Rezultate Experimentale

Pentru aceste simulări am folosit calea de ecou acustic. Simulăm algoritmul VSS-NLMS în scenariul reprezentat în Fig. 4.15, semnalul de intrare fiind o secvență vocală. În acest scenariu, RSZ se modifică după 50 de secunde de la 20 dB la 10 dB. Inițial, varianța σ_w^2 este considerată cunoscută pentru algoritmul NPVSS-NLMS, dar nu și atunci când are loc schimbarea RSZ-ului. Chiar dacă algoritmul NPVSS-NLMS depășește algoritmul VSS-NLMS în ceea ce privește rata de convergență și dezalinierea normalizată, algoritmul VSS-NLMS este cu siguranță mai robust la schimbările nivelului de putere.

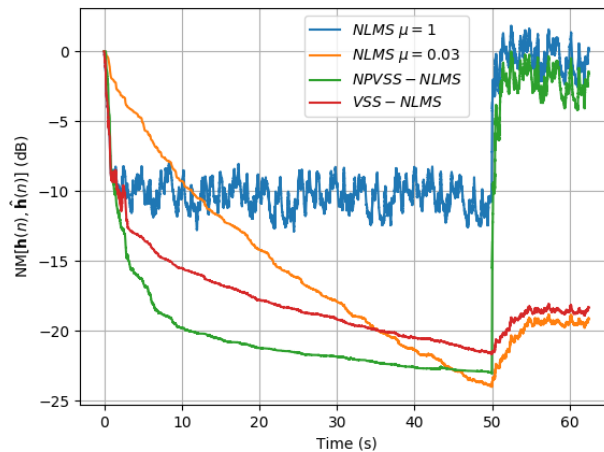


Figura 4.15 Dezalinierea normalizată a algoritmului NLMS cu $\mu = 1$ și $\mu = 0,03$, a algoritmului NPVSS-NLMS și a algoritmului VSS-NLMS folosind $l = 0,5$ și limita inferioară $\mu = 0,03$. Semnalul de intrare este o secvență vocală, $L = 512$, iar RSZ scade de la 20 dB la 10 dB după 50 de secunde.

4.4 APA cu Pas Variabil (VSS-APA)

Deoarece APA este o extensie a algoritmului NLMS, implementând un ordin de proiecție P , este firesc în acest moment să se continue dezvoltarea unui APA cu pas variabil (VSS-APA).

4.4.1 Dezvoltarea Algoritmului

Urmând modul de lucru din Secțiunea 4.3, cu siguranță putem îmbunătăți performanța generală, menținând aceeași complexitate de calcul.

Pe baza inducției matematice până la iterația k de reutilizare, obținem

$$\hat{\mathbf{h}}_k(n) = \hat{\mathbf{h}}(n-1) + [1 - (1 - \mu)^k] \mathbf{X}(n) [\mathbf{X}^T(n) \mathbf{X}(n)]^{-1} \mathbf{e}(n) \quad (4.50)$$

și $\mathbf{e}_k(n) = (1 - \mu)^{k-1} \mathbf{e}(n)$.

În acest moment, putem proiecta VSS-APA folosind același principiu folosit pentru algoritmul VSS-NLMS, generând un set de pași, fiecare fiind folosit pentru un anumit număr de iterații. Un pas mic μ ar putea bloca algoritmul să nu reacționeze la nicio modificare a sistemului necunoscut care este identificat. Din acest motiv, putem utiliza un detector [25] care resetează utilizarea pașilor generați.

4.4.2 Rezultate Experimentale

Sistemul necunoscut ce trebuie determinat este reprezentat de calea de ecou acustic.

Mai departe, semnalul de intrare este secvență vocală. În Fig. 4.22, calea de ecou se schimbă la mijlocul simulării. Chiar dacă rata de convergență a APA cu $\mu = 1$ este similară cu cea a VSS-APA, APA prezintă variații de peste 5 dB în unele scenarii. De asemenea, VSS-APA continuă să atingă un nivel de dezaliniere normalizat mult mai scăzut decât APA și menține performanțele bune după schimbarea căii de ecou.

4.5 Rezumat și Concluzii

În acest capitol, pornind de la metoda reutilizării datelor, am dezvoltat trei algoritmi adaptivi: LMSO-W-DR, VSS-NLMS și VSS-APA. Algoritmul LMSO-W-DR a fost dezvoltat prin optimizarea metodei reutilizării datelor și îmbunătățirea performanțelor acestuia în comparație cu precursorul său, adică algoritmul LMSO-W, prezentat în Secțiunea 4.2. Scopul nostru a fost să obținem un algoritm care să ofere o performanță apropiată de cea a algoritmului LMSO-G și simulările au evidențiat că am atins cu succes acest lucru în ceea ce privește rata de convergență, urmărirea și estimările bune. Câștigul în performanță al algoritmului LMSO-W-DR se realizează cu o complexitate de calcul

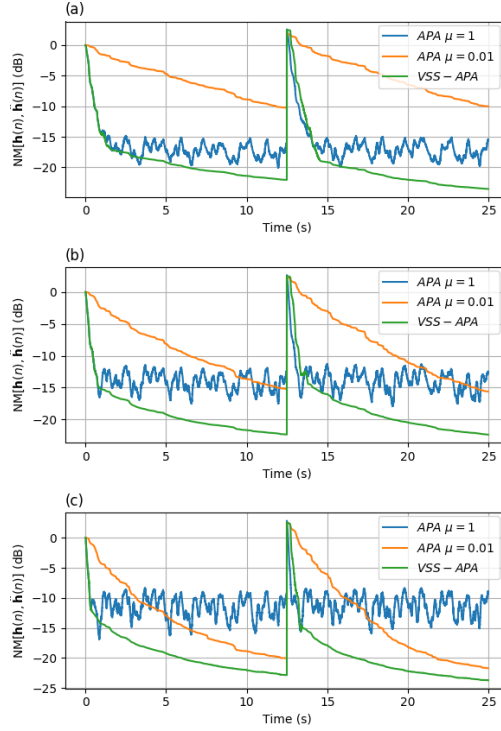


Figura 4.22 Dezalinierea normalizată a APA cu $\mu = 1$ și $\mu = 0,01$ și dezalinierea normalizată a VSS-APA folosind dimensiunea pasului $\mu = 0,01$ și $l = 0,1$, pentru diferite valori ale ordinul de proiecție (a) $P = 2$; (b) $P = 4$; și (c) $P = 8$. Semnalul de intrare este o secvență vocală, $L = 512$ și $RSZ = 30$ dB.

liniară, adică $O(ML)$, cu $M \ll L$. De asemenea, LMSO-W-DR deschide o conexiune cu APA care poate fi exploatată în cercetări viitoare.

În plus, am folosit metoda de reutilizare a datelor pentru a analiza rata de convergență a algoritmului NLMS și APA și am dezvoltat o metodă de generare a pașilor care reprezintă partea centrală a algoritmilor obținuți: VSS-NLMS și VSS-APA. Simulările au evidențiat performanțele bune obținute în comparație cu precursorii lor, algoritmii NLMS și APA, având în vedere toate perspectivele. Un avantaj remarcabil al algoritmilor VSS-NLMS și VSS-APA este robustețea în timpul variațiilor RSZ-ului, care într-o conversație în timp real reprezintă o situație obișnuită. Chiar dacă algoritmii VSS-NLMS și VSS-APA introduc utilizarea unui mecanism de resetare a pasului, complexitatea lor de calcul este încă aceeași ca și a algoritmilor NLMS și APA, introducând doar câteva operații suplimentare. Mai mult, în timp ce APA necesită un ordin de proiecție P mai mare pentru a îmbunătăți performanța, VSS-APA obține performanțe mult mai bune cu un ordin de proiecție mai mic, ceea ce reduce complexitatea de calcul.

Metodele dezvoltate și prezentate în acest capitol pot fi utilizate în algoritmi viitori, pentru a îmbunătăți performanțele oricărui tip de algoritm care implică parametrii fișii. Simulările susțin că LMSO-W-DR, VSS-NLMS și VSS-APA sunt candidați potriviți pentru aplicații în timp real, care necesită și o complexitate de calcul redusă.

Capitolul 5

Filtre Adaptive în Cascadă

Identificarea sistemului sau capacitatea de urmărire nu sunt singurele probleme care apar în aplicațiile ce implică anularea ecoului. În aplicațiile din lumea reală, calea ecoului poate avea mii de coeficienți. Prin urmare, performanța algoritmului poate fi afectată în ceea ce privește rata de convergență, precizia estimărilor sau urmărirea, dar la fel de important, complexitatea de calcul crește odată cu lungimea căii de ecou. Literatura oferă soluții care implică un filtru adaptiv cu o structură FIR, care vizează căi de ecou rare prin natura lor de a fi. În plus, mediul (adică calea de ecou) poate fi caracterizat de diferite proprietăți care pot conduce la generarea de reflexii multiple, aceasta fiind o altă provocare, iar în aplicațiile din lumea reală, se numește efect de reverberație.

În acest capitol, abordând o perspectivă multiliniară, dezvoltăm soluții la aceste probleme complexe, urmărind minimizarea complexității de calcul, obținerea unei performanțe bune și gestionarea efectului de reverberație. Pentru a evalua efectul de reverberație din punct de vedere matematic, descriem răspunsul la impuls folosind descompunerea produsului Kronecker. Dezvoltarea din acest capitol este publicată în [2], [7], [8].

5.1 Structuri Multiliniare fără Memorie

Un sistem MISO poate fi definit prin următoarea relație de intrare-ieșire

$$y(n) = \sum_{l_1=1}^{L_1} \sum_{l_2=1}^{L_2} \dots \sum_{l_N=1}^{L_N} x_{l_1 l_2 \dots l_N}(n) h_{1,l_1} h_{2,l_2} \dots h_{N,l_N}, \quad (5.1)$$

descrișă de coeficienții h_{i,l_i} . În plus, să rescriem ecuația (5.1) ca

$$y(n) = \sum_{l_1=1}^{L_1} h_{1,l_1} \dots \sum_{l_{N-1}=1}^{L_{N-1}} h_{N-1,l_{N-1}} s_{l_{N-1}}(n), \quad (5.3)$$

unde

$$s_{l_{N-1}}(n) = \sum_{l_N=1}^{L_N} x_{l_1 l_2 \dots l_N}(n) h_{N, l_N} \quad (5.4)$$

definește rezultatul structurii MISO N caracterizată de L_N intrări. Astfel, Fig. 5.1(a) descrie structura MISO $s_{l_{N-1}}$ cu intrările L_N , în timp ce Fig. 5.1(b) prezintă reprezentarea sa simbolică, fiind un sumator ponderat fără memorie.

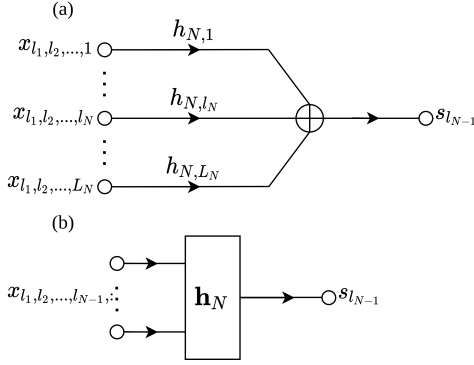


Figura 5.1 (a) Reprezentarea structurii MISO $s_{l_{N-1}}$; (b) Reprezentarea simbolică a $s_{l_{N-1}}$ (sumator ponderat fără memorie).

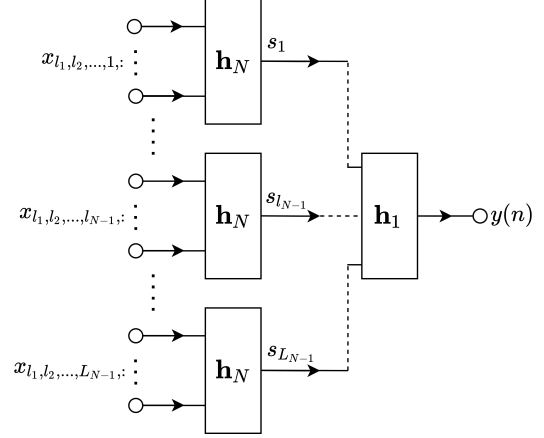


Figura 5.2 Sistemul MISO cu N nivele de sumatoare ponderate.

Pe baza (5.3) și luând în considerare reprezentarea simbolică din Fig. 5.1(b), putem schița sistemul MISO prin reprezentarea grafică din Fig. 5.2, care cuprinde N nivele de sumatoare ponderate.

5.2 Structuri Multiliniare cu Memorie

Transformarea structurii $s_{l_{N-1}}$ într-o structură cu memorie se poate realiza prin inserarea unei linii de întârziere, astfel încât putem scrie

$$s_{l_{N-1}}(n) = \sum_{l_N=1}^{L_N} x_{l_1 l_2 \dots l_N}(n - l_N + 1) h_{N, l_N}, \quad (5.5)$$

cu $x_{l_1 l_2 \dots l_N}(n) = x_{l_{N-1}}(n - l_N + 1)$, $l_N = 1, 2, \dots, L_N$.

În consecință, structura MISO $s_{l_{N-1}}$ devine o structură SISO și ieșirea sa este generată de un filtru FIR transversal $\mathbf{h}_N$. Procedând în același mod pe fiecare nivel al sistemului global, obținem în final o cascadă de filtre FIR transversale, așa cum este prezentat în Fig. 5.6.

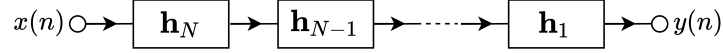


Figura 5.6 Sistem SISO în configurație cascadă.

5.3 Modelul Sistemului bazat pe Descompunerea Produsului Kronecker

Luând în considerare cadrul de identificare a sistemului ca înainte, introducem ieșirea sistemului MISO ca

$$y(n) = \mathcal{X}(n) \times_1 \mathbf{h}_1^T \times_2 \mathbf{h}_2^T \times_i \dots \times_N \mathbf{h}_N^T, \quad (5.9)$$

unde $\mathbf{h}_i$ reprezintă al i -lea filtru transversal din cascada descrisă în Fig. 5.6, unde $\times_i$ denotă operația de înmulțire cu dimensiunea, $i = 1, 2, \dots, N$ și N reprezintă gradul multiliniar.

Putem rescrie (5.9) ca $y(n) = \text{vec}^T[\mathcal{X}(n)]\text{vec}(\mathcal{H})$, unde $\mathcal{H}$ este un tensor de rang 1 de dimensiune $L_1 \times L_2 \times \dots \times L_N$ [11][24], sau $y(n) = \tilde{\mathbf{x}}^T(n)\mathbf{h}$. Vectorul de intrare $\tilde{\mathbf{x}}(n)$ și vectorul coeficienților $\mathbf{h}$ au $L_1 L_2 \dots L_N$ elemente. Putem concluziona că perspectiva multiliniară se traduce într-o problemă liniară prin modul de prelucrare a datelor de intrare.

5.4 Analiza în Transformata Z

Ieșirea sistemului global din Fig. 5.6 poate fi dezvoltată ca

$$Y(z) = S_{l_1}(z)H_1(z) = X_1(z)H_2(z^{L_1})H_1(z) = X_1(z)H(z), \quad (5.24)$$

cu funcția de transfer $H(z) = H_2(z^{L_1})H_1(z)$. În această funcție de transfer, putem sublinia că $H_2(z^{L_1})$ rezultă din interpolarea cu zerouri cu factorul L_1 al funcției $H_2(z)$ și este caracterizat de L_2 coeficienți nenuli dintr-o lungime totală de $L_1(L_2 - 1) + 1$ coeficienți.

5.5 Algoritmul NLMS Multiliniar în Cascadă

Deoarece algoritmul NLMS este foarte popular printre algoritmi adaptivi datorită simplității sale și a complexității de calcul scăzute, să-l folosim ca model de lucru și să demonstrăm abordarea filtrelor în cascadă. În această secțiune, dezvoltăm algoritmul multiliniar în cascadă NLMS (NLMS-CMF) prin minimizarea funcțiilor de cost MSE pentru sistemul în cascadă.

5.5.1 Dezvoltarea Algoritmului

În această dezvoltare obținem ecuațiile algoritmului multilinear în cascadă NLMS (NLMS-CMF)

$$\begin{aligned}
 \hat{\mathbf{h}}_1(n) &= \hat{\mathbf{h}}_1(n-1) + \frac{\mu_{\hat{\mathbf{h}}_1} \tilde{\mathbf{x}}_{\hat{\mathbf{h}}_2 \hat{\mathbf{h}}_3 \dots \hat{\mathbf{h}}_N}(n) e_{\hat{\mathbf{h}}_2 \hat{\mathbf{h}}_3 \dots \hat{\mathbf{h}}_N}(n)}{\tilde{\mathbf{x}}_{\hat{\mathbf{h}}_2 \hat{\mathbf{h}}_3 \dots \hat{\mathbf{h}}_N}^T(n) \tilde{\mathbf{x}}_{\hat{\mathbf{h}}_2 \hat{\mathbf{h}}_3 \dots \hat{\mathbf{h}}_N}(n) + \delta_{\hat{\mathbf{h}}_1}}, \\
 \hat{\mathbf{h}}_2(n) &= \hat{\mathbf{h}}_2(n-1) + \frac{\mu_{\hat{\mathbf{h}}_2} \tilde{\mathbf{x}}_{\hat{\mathbf{h}}_1 \hat{\mathbf{h}}_3 \dots \hat{\mathbf{h}}_N}(n) e_{\hat{\mathbf{h}}_1 \hat{\mathbf{h}}_3 \dots \hat{\mathbf{h}}_N}(n)}{\tilde{\mathbf{x}}_{\hat{\mathbf{h}}_1 \hat{\mathbf{h}}_3 \dots \hat{\mathbf{h}}_N}^T(n) \tilde{\mathbf{x}}_{\hat{\mathbf{h}}_1 \hat{\mathbf{h}}_3 \dots \hat{\mathbf{h}}_N}(n) + \delta_{\hat{\mathbf{h}}_2}}, \\
 &\vdots \\
 \hat{\mathbf{h}}_N(n) &= \hat{\mathbf{h}}_N(n-1) + \frac{\mu_{\hat{\mathbf{h}}_N} \tilde{\mathbf{x}}_{\hat{\mathbf{h}}_1 \hat{\mathbf{h}}_2 \dots \hat{\mathbf{h}}_{N-1}}(n) e_{\hat{\mathbf{h}}_1 \hat{\mathbf{h}}_2 \dots \hat{\mathbf{h}}_{N-1}}(n)}{\tilde{\mathbf{x}}_{\hat{\mathbf{h}}_1 \hat{\mathbf{h}}_2 \dots \hat{\mathbf{h}}_{N-1}}^T(n) \tilde{\mathbf{x}}_{\hat{\mathbf{h}}_1 \hat{\mathbf{h}}_2 \dots \hat{\mathbf{h}}_{N-1}}(n) + \delta_{\hat{\mathbf{h}}_{N-1}}}, \tag{5.34}
 \end{aligned}$$

unde $\delta_{\hat{\mathbf{h}}_i}$ denotă parametrul de regularizare pentru fiecare filtru $\hat{\mathbf{h}}_i$, $i = 1, 2, \dots, N$.

5.5.2 Rezultate Experimentale

Să demonstrăm structurile dezvoltate rulând simulările pe două filtre în cascadă ($\hat{\mathbf{h}}_1, \hat{\mathbf{h}}_2$), adică $N = 2$, acesta fiind un algoritm NLMS de formă biliniară în cascadă (NLMS-CBF). În aceste simulări, luăm în considerare un scenariu de identificare a sistemului în care sistemul necunoscut este caracterizat de prima cale de ecou din Recomandarea G168 [1] concatenată cu zerouri, $L = 512$.

În Fig. 5.9 analizăm capacitatea de urmărire și calea ecoului se schimbă la momentul de 7 secunde. Putem observa că pentru $\mu = \mu_{\hat{\mathbf{h}}_1} = \mu_{\hat{\mathbf{h}}_2} = 1$ algoritmul NLMS prezintă o rată de convergență puțin mai rapidă decât cea a algoritmul NLMS-CBF atunci când calea ecoului se modifică. Cu toate acestea, nivelul de dezaliniere normalizat este același. Pe de altă parte, se pare că pașii $\mu = \mu_1 = \mu_2 = 0.1$ sunt prea mici atunci când semnalul de intrare este corelat. Prin urmare, trebuie selectată o valoare adecvată a pasului.

5.6 Algoritm RLS Multiliniar în Cascadă

Algoritm RLS este popular datorită performanțelor sale bune în ceea ce privește rata de convergență, nivelul de dezaliniere normalizat, și capacitatea de urmărire, fiind ușor de parametrizat și utilizat în aplicații practice. În orice caz, performanța sa vine cu costul complexității de calcul, aceasta fiind $O(L^2)$. Prin urmare, ne propunem să oferim o soluție care folosește avantajul abordării filtrelor în cascadă pentru a reduce complexitatea de calcul și a menține buna performanță a algoritmului RLS, astfel încât să se obțină un algoritm RLS de formă multiliniară în cascadă (RLS-CMF). În această dezvoltare, începem prin a minimiza funcțiile de cost LS pentru sistemul în cascadă.

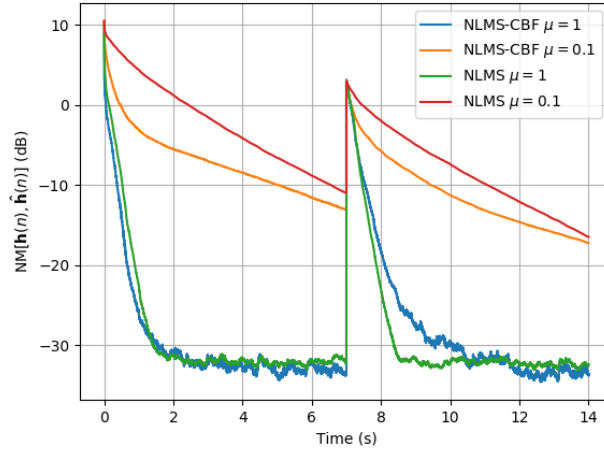


Figura 5.9 Dezalinieră normalizată a algoritmilor NLMS-CBF și NLMS. Semnalul de intrare este un proces AR(1), $\beta = 0,8$, $L = 512$, $L_1 = L_2 = L/2$ și $RSZ = 20$ dB.

5.6.1 Dezvoltarea Algoritmului

Urmând aceeași procedură ca în Secțiunea 2.5, putem identifica ecuațiile de actualizare ale algoritmului RLS multiliniar în cascadă ca

$$\begin{aligned}
 \hat{\mathbf{h}}_1(n) &= \hat{\mathbf{h}}_1(n-1) + \mathbf{s}_{\hat{\mathbf{h}}_2\hat{\mathbf{h}}_3\cdots\hat{\mathbf{h}}_N}(n)e_{\hat{\mathbf{h}}_2\hat{\mathbf{h}}_3\cdots\hat{\mathbf{h}}_N}(n), \\
 \hat{\mathbf{h}}_2(n) &= \hat{\mathbf{h}}_2(n-1) + \mathbf{s}_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_3\cdots\hat{\mathbf{h}}_N}(n)e_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_3\cdots\hat{\mathbf{h}}_N}(n), \\
 &\vdots \\
 \hat{\mathbf{h}}_N(n) &= \hat{\mathbf{h}}_N(n-1) + \mathbf{s}_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_2\cdots\hat{\mathbf{h}}_{N-1}}(n)e_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_2\cdots\hat{\mathbf{h}}_{N-1}}(n),
 \end{aligned} \tag{5.38}$$

unde $\mathbf{s}_{\hat{\mathbf{h}}_2\hat{\mathbf{h}}_3\cdots\hat{\mathbf{h}}_N}(n)$, $\mathbf{s}_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_3\cdots\hat{\mathbf{h}}_N}(n)$, $\vdots$, $\mathbf{s}_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_2\cdots\hat{\mathbf{h}}_{N-1}}(n)$ denotă termenii de actualizare.

5.6.2 Rezultate Experimentale

În această secțiune ne propunem să comparăm performanța algoritmului RLS multiliniar în cascadă cu cea a algoritmului RLS în două contexte diferite. În primul context considerăm că sistemul necunoscut este caracterizat de prima cale de ecou din Recomandarea G168 [1], în timp ce în al doilea context calea de ecou este obținută ca produs Kronecker între mai multe răspunsuri la impuls [23].

Pentru primul context, alegem trei grade diferite de forme multilinare: $N = 2$, forma biliniară (BF), $N = 3$, forma triliniară (TF), $N = 4$, forma pătratică (QF). Prin urmare, notăm algoritmul RLS multiliniar în cascadă pentru gradele alese ca algoritmul BF RLS în cascadă (RLS-CBF), algoritmul TF RLS în cascadă (RLS-CTF) și algoritmul QF RLS în cascadă (RLS-CQF).

Apoi, în Fig. 5.12 semnalul de intrare este un proces AR(1) cu $\beta = 0,85$ și calea de ecou se schimbă în mijlocul simulării. Atunci când calea de ecou se modifică, efectul

gradului multiliniar este mult mai vizibil în ceea ce privește nivelul de dezaliniere normalizat. De asemenea, se pare că algoritmul RLS-CBF atinge nivelul de dezaliniere normalizat al algoritmului RLS la un moment dat (ex., după 5 secunde de simulare).

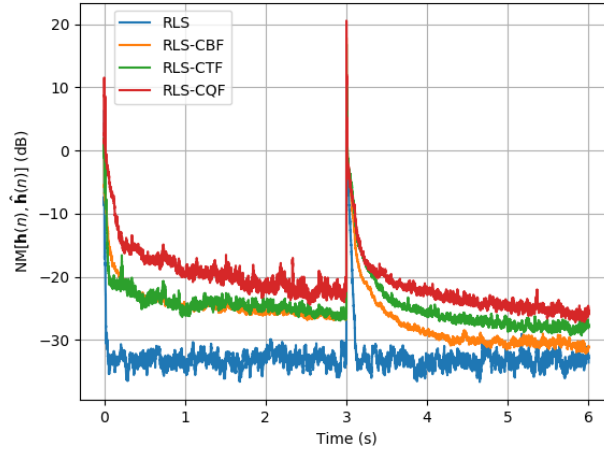


Figura 5.12 Dezalinierea normalizată a algoritmilor RLS-CBF, RLS-CTF, RLS-CQF și RLS. Semnalul de intrare este un proces AR(1) cu $\beta = 0,85$ și $RSZ = 20$ dB.

Mai departe, continuăm experimentele în cadrul celui de al doilea context, în care alegem două grade multilinare diferite, $N = 2$ (bilinar) și $N = 3$ (triliniar). Notăm algoritmul RLS multiliniar în cascadă bazat pe descompunerea produsului Kronecker ca algoritmul RLS-CKD.

Să începem cu simulările formei biliniare, unde sistemul ce trebuie identificat este obținut prin produsul Kronecker între primul răspuns la impuls, adică $\mathbf{h}_1$, din Recomandarea G168 [1] (cu $L_1 = 64$) și un răspuns la impuls exponențial, $\mathbf{h}_2$ (cu $L_2 = 8$). Răspunsul la impuls global este $\mathbf{h} = \mathbf{h}_2 \otimes \mathbf{h}_1$, cu $L = L_1 L_2 = 512$ coeficienți.

În Fig. 5.17 semnalul de intrare este o secvență vocală și calea de ecou se schimbă în mijlocul simulării. Putem remarca că algoritmul RLS atinge un nivel de dezaliniere normalizat de -20 dB după 7 secunde, ceea ce nu este de dorit într-o aplicație în timp real. Pe de altă parte, algoritmul RLS-CKD atinge un nivel de dezaliniere normalizat de -40 dB în 2 secunde. De data aceasta influența parametrului M este vizibilă de la început și afectează rata de convergență a algoritmului RLS-CKD. Cu toate acestea, putem sublinia ușurința algoritmului RLS-CKD pentru a determina noul sistem atunci când răspunsul la impuls se schimbă.

În continuare, analizăm performanța algoritmului RLS-CKD în contextul triliniar. Răspunsul la impuls ce trebuie determinat este obținut ca produs Kronecker între prima cale de ecou din Recomandarea G168 [1], adică $\mathbf{h}_1$ (cu $L_1 = 64$), un răspuns la impuls generat aleatoriu $\mathbf{h}_2$ (cu $L_2 = 8$) și un răspuns la impuls exponențial $\mathbf{h}_3$ (cu $L_3 = 4$). Pe scurt, sistemul global care trebuie identificat este $\mathbf{h} = \mathbf{h}_3 \otimes \mathbf{h}_2 \otimes \mathbf{h}_1$, cu $L = L_1 L_2 L_3 = 2048$ coeficienți.

În Fig. 5.20 semnalul de intrare este un proces AR(1) cu $\beta = 0,9$ și calea ecoului se schimbă după 4 secunde. Putem remarca că algoritmul RLS-CKD depășește algoritmul

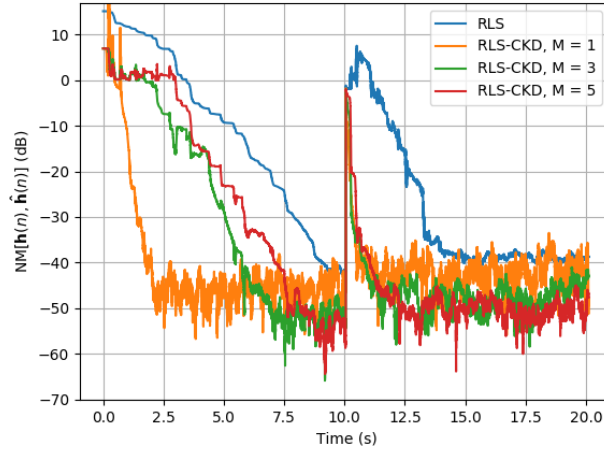


Figura 5.17 Dezalinierea normalizată a algoritmilor RLS-CKD ($L_1 = 64$, $L_2 = 8$) și RLS clasic ($L = 512$). Semnalul de intrare este o secvență vocală, RSZ = 30dB, iar răspunsul la impuls se modifică după 10 secunde de simulare.

RLS în ceea ce privește rata de convergență, nivelul de dezaliniere normalizat și capacitatea de urmărire. De data aceasta, efectul parametrului M asupra ratei de convergență a algoritmului RLS-CKD este vizibil încă de la început.

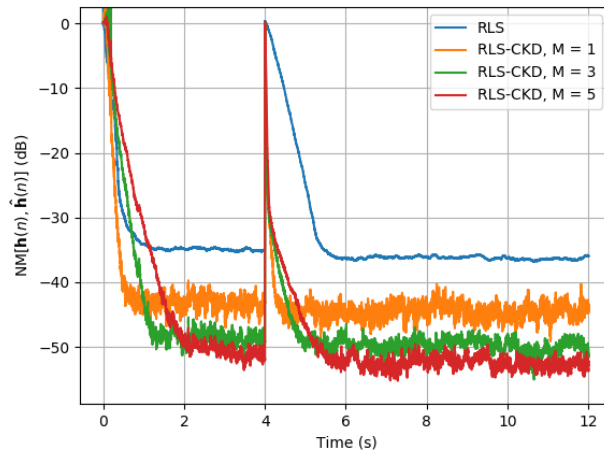


Figura 5.20 Dezalinierea normalizată a algoritmilor RLS-CKD ($L_1 = 64$, $L_2 = 8$, $L_3 = 4$) și RLS clasic ($L = 2048$). Semnalul de intrare este un proces AR(1) cu $\beta = 0,9$, RSZ = 20dB, iar răspunsul la impuls se modifică după 4 secunde de simulare.

5.7 Analiza Complexității de Calcul

Modul de împărțire a unui sistem global într-o cascadă de sisteme mai mici (în ceea ce privește lungimea răspunsului la impuls), reprezintă un avantaj semnificativ pentru reducerea complexității de calcul. Acest avantaj este mai important pentru acei algoritmi care prezintă o complexitate de calcul pătratică sau cubică, făcându-i adecvați pentru aplicații reale, așa cum este indicat în tabelul 5.1.

Tabel 5.1 Reducerea Complexității de Calcul a Algoritmilor

Algoritm	Complexitate	Exemplu de Operații
RLS	$O(L^2)$	$2048^2 = 4194304$
RLS-CBF	$\sum_{i=1}^2 O(L_i^2)$	$2 \times 1024^2 = 2097152$
RLS-CTF	$\sum_{i=1}^3 O(L_i^2)$	$3 \times 682^2 = 1395372$
RLS-CQF	$\sum_{i=1}^4 O(L_i^2)$	$4 \times 512^2 = 1048576$

Cu toate acestea, dacă răspunsul la impuls poate fi descompus prin produs Kronecker, calea de ecou poate fi identificată printr-o cascadă de filtre de lungime mai mică și beneficiile în ceea ce privește complexitatea de calcul sunt prezentate în tabelul 5.2.

Tabel 5.2 Reducerea Complexității de Calcul a Algoritmilor prin Abordarea KPD

Algoritm	Complexitate	Exemplu de Operații
RLS	$O(L^2)$	$2048^2 = 4194304$
RLS-CKD (bilinar)	$\sum_{i=1}^2 O(L_i^2)$	$64^2 + 32^2 = 5120$
RLS-CKD (trilinear)	$\sum_{i=1}^3 O(L_i^2)$	$64^2 + 8^2 + 4^2 = 4176$
RLS-CKD (pătratic)	$\sum_{i=1}^4 O(L_i^2)$	$32^2 + 8^2 + 4^2 + 2^2 = 1108$

5.8 Rezultate și Concluzii

În acest capitol, am introdus un set de structuri cu și fără memorie și am dezvoltat o modalitate de împărțire a unui filtru de lungime mare într-o cascadă de filtre de lungime mai mică. Am demonstrat teoria prin experimente și am subliniat performanțele prin algoritmi NLMS-CBF, RLS-CBF, RLS-CTF, RLS-CQF. De asemenea, am prezentat o modalitate optimizată de a construi datele de intrare, care a fost abordată în contextul căilor de ecou descrise prin KPD. Am evaluat un efect special care poate fi întâlnit în medii acustice, efectul de reverberație, abordat în contextul căilor de ecou descrise prin KPD. Am evidențiat că algoritmul RLS-CKD este un bun candidat pentru a gestiona acest tip de efect și obține performanțe bune în ceea ce privește rata de convergență, nivelul de dezaliniere normalizat și capacitatea de urmărire. Simulările au evidențiat că algoritmul propus depășește omologii săi (adică algoritmi NLMS sau RLS). De asemenea, am prezentat beneficiile din punct de vedere al complexității de calcul pentru o cale rudimentară de ecou și pentru o cale a ecoului care poate fi descrisă prin KPD.

Capitolul 6

Concluzii

6.1 Rezultate Obținute

Problema identificării sistemului este una dintre cele mai interesante în domeniul procesării de semnale, ce ridică multe provocări în zilele noastre. Această teză oferă cercetări cuprinzătoare în domeniul filtrării adaptive și introduce noi algoritmi pentru identificarea sistemului, vizând atât sisteme liniare, cât și multiliniare. Totuși, pentru o mai bună înțelegere a contextului și pentru a facilita legătura cu teoria, rezultatele obținute au fost prezentate împreună cu concluziile aferente la finalul fiecărui capitol. Prin urmare, nu este nevoie să le reiterăm aici. Cu toate acestea, principalele contribuții sunt enumerate în secțiunea următoare, subliniind principalele realizări.

6.2 Contribuții Originale

Contribuțiile originale sunt enumerate pe scurt mai jos, specificând articolul original în care a fost publicată contribuția. Mențiunea $[i]$, unde i este un număr, indică poziția în listă din secțiunea următoare.

Capitolul 3

- Dezvoltarea unui algoritm LMS optimizat cu pas variabil, algoritmul LMSO [1];
- Dezvoltarea unui algoritm LMS optimizat cu pas diferențiat, algoritmul ODSS-LMS [2];
- Dezvoltarea versiunilor simplificate pentru algoritmii LMSO și ODSS-LMS; algoritmii LMSO-W și ODSS-LMS-W, potriviți pentru aplicații în timp real [1], [2];
- Analiza convergenței și stabilității algoritmilor propuși [1], [2];
- Analiza complexității de calcul [1], [2];
- Analiza performanțelor algoritmilor propuși, în comparație cu omologii lor sau cu alte soluții recente [1], [2].

Capitolul 4

- Îmbunătățirea algoritmului LMSO-W propus anterior, prin utilizarea tehnicii de reutilizare a datelor, obținând algoritmul LMSO-W-DR [4];
- Dezvoltarea teoriei pentru a prezenta conexiunea cu APA [4];
- Analiza performanței algoritmului LMSO-W-DR [4];
- Dezvoltarea algoritmului NLMS cu pas variabil (VSS-NLMS) [7];
- Dezvoltarea APA cu pas variabil (VSS-APA) [8];
- Analiza modurilor de convergență pentru algoritmul VSS-NLMS și VSS-APA [7], [8];
- Studii experimentale pentru a evidenția îmbunătățirile obținute prin utilizarea abordărilor propuse [7], [8].

Capitolul 5

- Introducerea structurilor cu și fără memorie (MISO sau SISO), care pot fi descrise printr-o formă multiliniară de intrare-ieșire [3], [5];
- Pornind de la structura ponderată filtru-sumator, dezvoltarea unei metode de împărțire a unui filtru de lungime mare într-o cascadă de filtre de lungime mai mică, urmărind atât identificarea sistemelor lineare, cât și multiliniare [3], [5];
- O modalitate optimizată de a construi tensorul datelor de intrare pentru identificarea sistemelor multiliniare [6];
- Dezvoltarea algoritmilor NLMS și RLS multiliniari în cascadă [3];
- Validarea teoriei prin algoritmul NLMS cu formă biliniară în cascadă (NLMS-BF) [3];
- Analiza performanței bazată pe ordinul multiliniar [5];
- Studiu experimental pentru a prezenta identificarea unui sistem care se caracterizează prin efectul de reverberație folosind algoritmul propus RLS-CKD [6].

6.3 Lista Publicațiilor Originale

În această secțiune este prezentată lista publicațiilor originale cu citatele corespunzătoare din Referințe.

- [1] A.-G. Rusu, S. Ciochină, and C. Paleologu, “On the Step-Size optimization of the LMS Algorithm”, *42nd International Conference on Telecommunications and Signal Processing (TSP)*, pp. 168–173, 2019 [6] [\[ISI Proceedings\]](#);
- [2] A.-G. Rusu, S. Ciochină, C. Paleologu, and J. Benesty, “An Optimized Differential Step-Size LMS Algorithm”, *Algorithms*, 12(8):147, 2019 [9] [\[ISSN: 1999-4893, CiteScore - Q2 \(Numerical Analysis\)\]](#);

- [3] A.-G. Rusu and S. Ciochină, “Cascaded adaptive filters in a bilinear approach for system identification”, *International Symposium on Electronics and Telecommunications (ISETC)*, pp. 1–4, Nov. 2020 [2] [\[ISI Proceedings\]](#);
- [4] A.-G. Rusu, L.-M. Dogariu, S. Ciochină, and C. Paleologu, “A Data-Reuse Approach for an Optimized LMS Algorithm”, *International Conference on Speech Technology and Human-Computer Dialogue (SpeD)*, pp. 31–35, 2021 [5] [\[ISI Proceedings\]](#);
- [5] A.-G. Rusu, S. Ciochină, and C. Paleologu, “Cascaded Adaptive Filters in a Multilinear Approach for System Identification”, *International Symposium for Design and Technology in Electronic Packaging (SIITME)*, pp. 1–4, Oct. 2021 [7] [\[ISI Proceedings\]](#);
- [6] A.-G. Rusu, S. Ciochină, C. Paleologu, and J. Benesty, “Cascaded RLS Adaptive Filters Based on a Kronecker Product Decomposition”, *Electronics*, 11(3):409, 2022 [8] [\[ISSN: 2079-9292, Q3 - IF - 2.690 \]](#);
- [7] A.-G. Rusu, C. Paleologu, J. Benesty, and S. Ciochină, “A Variable Step Size Normalized Least-Mean-Square Algorithm Based on Data Reuse”, *Algorithms*, 15(4):111, 2022 [3] [\[ISSN: 1999-4893, CiteScore - Q2 \(Numerical Analysis\) \]](#);
- [8] A.-G. Rusu, L.-M. Dogariu, R.-L. Costea, C. Paleologu, J. Benesty, and S. Ciochină, “A Variable Step-Size Affine Projection Algorithm Based on Data Reuse”, *45th International Conference on Telecommunications and Signal Processing (TSP)*, 2022 [4] [\[ISI Proceedings\]](#).

6.4 Perspective pentru Dezvoltări Viitoare

Cercetările viitoare se vor concentra pe îmbunătățirea lucrării acestei teze în contextul liniar și multiliniar de identificarea a sistemului. Cu siguranță, una dintre direcții este dezvoltarea unei metode de ultimă generație pentru estimarea zgomotului sistemului și conversia algoritmilor LMSO și ODSS-LMS propuși, în algoritmi neparametrici. Prin urmare, algoritmii LMSO și ODSS-LMS ar putea deveni soluții generale pentru identificarea oricărui tip de răspuns la impuls. O altă cale de cercetare este analiza conexiunii dintre algoritmul LMSO-DR și APA, explorând noi modalități de a gestiona datele de intrare.

Ne propunem să extindem modul de împărțire a unui filtru de lungime mare într-o cascadă de filtre de lungime mai mică, pentru a analiza diferite tipuri de neliniarități care pot apărea în aplicațiile din lumea reală.

Bibliografie

- [1] (2002). *Digital Network Echo Cancellers, ITU-T Recommendations G.168*.
- [2] A.-G. Rusu and S. Ciochină (Nov. 2020). Cascaded adaptive filters in a bilinear approach for system identification. In *International Symposium on Electronics and Telecommunications (ISETC)*, pages 1–4.
- [3] A.-G. Rusu, C. Paleologu, J. Benesty, and S. Ciochină (2022). A Variable Step Size Normalized Least-Mean-Square Algorithm Based on Data Reuse. *Algorithms*, 15(4).
- [4] A.-G. Rusu, L.-M. Dogariu, R.-L. Costea, C. Paleologu, J. Benesty, and S. Ciochină (2022). A Variable Step-Size Affine Projection Algorithm Based on Data Reuse. In *45th International Conference on Telecommunications and Signal Processing (TSP)*.
- [5] A.-G. Rusu, L.-M. Dogariu, S. Ciochină, and C. Paleologu (2021). A Data-Reuse Approach for an Optimized LMS Algorithm. In *International Conference on Speech Technology and Human-Computer Dialogue (SpeD)*, pages 31–36.
- [6] A.-G. Rusu, S. Ciochină, and C. Paleologu (2019). On the Step-Size optimization of the LMS Algorithm. In *42nd International Conference on Telecommunications and Signal Processing (TSP)*, pages 168–173.
- [7] A.-G. Rusu, S. Ciochină, and C. Paleologu (Oct. 2021). Cascaded Adaptive Filters in a Multilinear Approach for System Identification. In *International Symposium for Design and Technology in Electronic Packaging (SIITME)*, pages 1–4.
- [8] A.-G. Rusu, S. Ciochină, C. Paleologu, and J. Benesty (2022). Cascaded RLS Adaptive Filters Based on a Kronecker Product Decomposition. *Electronics*, 11(3).
- [9] A.-G. Rusu, S. Ciochină, C. Paleologu, Constantin, and J. Benesty (2019). An Optimized Differential Step-Size LMS Algorithm. *Algorithms*, 12(8).
- [10] B.A. Schnaufer and W.K. Jenkins (1993). New data-reusing LMS algorithms for improved convergence. In *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*, volume 2, pages 1584–1588.
- [11] C. Elisei-Iliescu, C. Paleologu, J. Benesty, C. Stanciu, C. Anghel, and S. Ciochină (May 2019). Recursive least-squares algorithms for the identification of low-rank systems. *IEEE/ACM Trans. Audio, Speech, Language Processing*, 27:903–918.
- [12] C. Paleologu and J. Benesty (Aug. 2012). A practical data-reuse adaptive algorithm for acoustic echo cancellation. *Proc. 20th European Signal Processing Conference*, pages 31–34.
- [13] C. Paleologu, S. Ciochină, and J. Benesty (2008). Variable step-size NLMS algorithm for under-modeling acoustic echo cancellation. *IEEE Signal Processing Lett.*, 15:5–8.

- [14] C. Paleologu, S. Ciochină, J. Benesty, and S. L. Grant (2015). An overview on optimized NLMS algorithms for acoustic echo cancellation. *EURASIP Journal Advances Signal Processing*.
- [15] D. L. Duttweiler (Sept. 2000). Proportionate normalized least-mean-squares adaptation in echo cancelers. *IEEE Trans. Speech, Audio Processing*, 8:508–518.
- [16] E.-W. Bai, D. Li (Nov. 2004). Convergence of the iterative Hammerstein system identification algorithm. *IEEE Transactions on Automatic Control*, 49(11):1929–1940.
- [17] G. Enzner, H. Buchner, A. Favrot, and F. Kuech (2014). Acoustic echo control. In *Academic Press Library in Signal Processing: Image, Video Processing and Analysis, Hardware, Audio, Acoustic and Speech Processing*, volume 4, chapter 30, pages 807–877. Academic Press.
- [18] H. Deng and M. Doroslovački (May. 2006). Proportionate adaptive algorithms for network echo cancellation. *IEEE Trans. Signal Processing*, 54:1794–1803.
- [19] J. A. Apolinário, Jr., M. L. R. de Campos, and P. S. R. Diniz (Nov. 2000). Convergence analysis of the binormalized data-reusing LMS algorithm. *IEEE Trans. Signal Processing*, 48:3235–3242.
- [20] J. Benesty and S. L. Gay (2002). An improved PNLMS algorithm. *Proc. IEEE ICASSP*, pages II–1881–II–1884.
- [21] J. Benesty and T. Gänslér (2003). On data-reuse adaptive algorithms. In *Proc. IWAENC*, pages 31–34.
- [22] K. Ozeki and T. Umeda (May 1984). An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties. *Electron. Commun. Jpn.*, 67-A:19–27.
- [23] L.-M. Dogariu, C. L. Stanciu, C. Elisei-Iliescu, C. Paleologu, J. Benesty, and S. Ciochină (Apr. 2021). Tensor-based adaptive filtering algorithms. *Symmetry*, 13.
- [24] L. M. Dogariu, C. Paleologu, J. Benesty, and S. Ciochină (Jan. 2020). An efficient Kalman filter for the identification of low-rank systems. *Signal Processing*, 166.
- [25] M. A. Iqbal and S. L. Grant (2008). Novel variable step size NLMS algorithms for echo cancellation. In *Proc. IEEE ICASSP*, pages 241–244.
- [26] P. S. R. Diniz and S. Werner (Jan. 2003). Set-membership binormalized data-reusing LMS algorithms. *IEEE Trans. Signal Processing*, 51(1):124–134.
- [27] S. Ciochină, C. Paleologu, and J. Benesty (2016). An optimized NLMS algorithm for system identification. *Signal Processing*, 118:115–121.
- [28] S.-H. Leung and C. F. So (Aug. 2005). Gradient-based variable forgetting factor RLS algorithm in time-varying environments. *IEEE Trans. Signal Processing*, 53:3141–3150.
- [29] S. L. Gay (1998). An efficient, fast converging adaptive filter for network echo cancellation. *Proc. IEEE Asilomar Conference on Signals, Systems, and Computers*, pages 394–398.