

Contribuții la teoria și implementarea metodelor de reducere adaptivă a zgomotului de fond în sistemele multi-canal de comunicații de voce

Contract nr. 331/01.10.2007, Cod ID_777

Sinteza 2007 – 2010

În cadrul sistemelor de comunicații cu “mâini-libere” (“*hands-free*”), utilizate în numeroase aplicații precum telefonie mobilă și teleconferințe, sunt necesare reducerea unor perturbații aferente mediului acustic, cum ar fi zgomotul de fond și ecoul acustic. Schemele ce stau la baza acestor aplicații au ca bloc principal filtrul adaptiv.

Sistemele de control activ al zgomotului (*ANC – active noise control*) [1] funcționează pe principiul reducerii interferenței dintre un câmp sonor perturbator (primar), preluat de către un număr de senzori (microfoane) și un al doilea câmp sonor (secundar) generat de către un număr de difuzoare. O categorie de algoritmi adaptivi rapizi bazați pe proiecții afine (*FAP – fast affine projection*) au fost analizați în contextul sistemelor ANC multi-canal [2], [3]. Acești algoritmi oferă o soluție de compromis avantajoasă între viteza de convergență și complexitatea aritmetică implicată. Chiar dacă din punct de vedere al vitezei de convergență sunt inferiori algoritmilor bazați pe optimizarea în sensul celor mai mici patrate (*LS – least squares*), algoritmi FAP compensează printr-o complexitate aritmetică mai redusă comparativ cu algoritmi de tip LS. Există însă o serie de probleme de natură numerică ale algoritmilor FAP, ce le limitează într-o anumită măsură utilizarea la scară largă în cadrul aplicațiilor din domeniul acustic. Structura matematică a acestor algoritmi implică rezolvarea unui sistem de ecuații liniare. Numărul ecuațiilor este dat de ordinul de proiecție afină al algoritmului. Alegerea unui ordin mare de proiecție afină conduce la o soluție trivială din punct de vedere al complexității numerice. Rezolvarea acestui set de ecuații liniare din cadrul algoritmului FAP presupune însă utilizarea unui algoritm de tip LS rapid (*fast LS*), ce poate conduce la probleme de instabilitate numerică specifice, în special în cazul multi-canal. În consecință, au fost investigate o serie de soluții alternative. Metoda Gauss-Seidel constituie una din tehnicile utilizate în acest context [3]. Ulterior, în scopul reducerii complexității aritmetice, se utilizează proiecțiile pseudo-afine (*PAP – pseudo-affine projection*), rezultând o nouă familie de algoritmi, denumiți GS-PAP (*Gauss-Seidel pseudo-affine projection*) [4], [5]. De asemenea, pentru o reducere suplimentară a complexității aritmetice au fost utilizate metode bazate pe iterații DCD [6]. S-a demonstrat că algoritmul MFX-DCDAP (*Modified Filtered-x Dichotomous Coordinate Descent Affine Projection*) obține performanțe similare cu algoritmul MFX-AP (*Modified Filtered-x Affine Projection Algorithm*), deși beneficiază de o complexitate mult mai redusă [7], [8]. În acest context, o soluție bazată pe PAP a condus la dezvoltarea algoritmului MFX-DCDPAP (*Modified Filtered-x Dichotomous Coordinate Descent Pseudo Affine Projection*), analizat în [6]. Recent, o nouă tehnică de reducere a complexității aritmetice pentru algoritmi de tip AP a fost propusă în [9].

Echipa noastră de cercetare a abordat aceste tehnici în contextul aplicațiilor ANC [10], [11]. În Fig. 1 sunt prezentate câteva evaluări ale complexității aritmetice într-o aplicație de ANC multi-canal. Notația I reprezintă numărul de senzori de referință, J este numărul dispozitivelor de control (*actuators*), K se referă la numărul senzorilor de eroare, L reprezintă lungimea filtrelor FIR cu coeficienți fiși, M este lungimea căilor (modelelor), N_u reprezintă numărul de iterații efectuate de procedura DCD, iar M_b este numărul de biți. În Fig. 2 sunt prezentate rezultatele unui experiment, utilizând parametrii $I = 1, J = 3, K = 2, L = 150, M = 64, M_b = 16$ și $\text{SNR} = 10\text{dB}$.

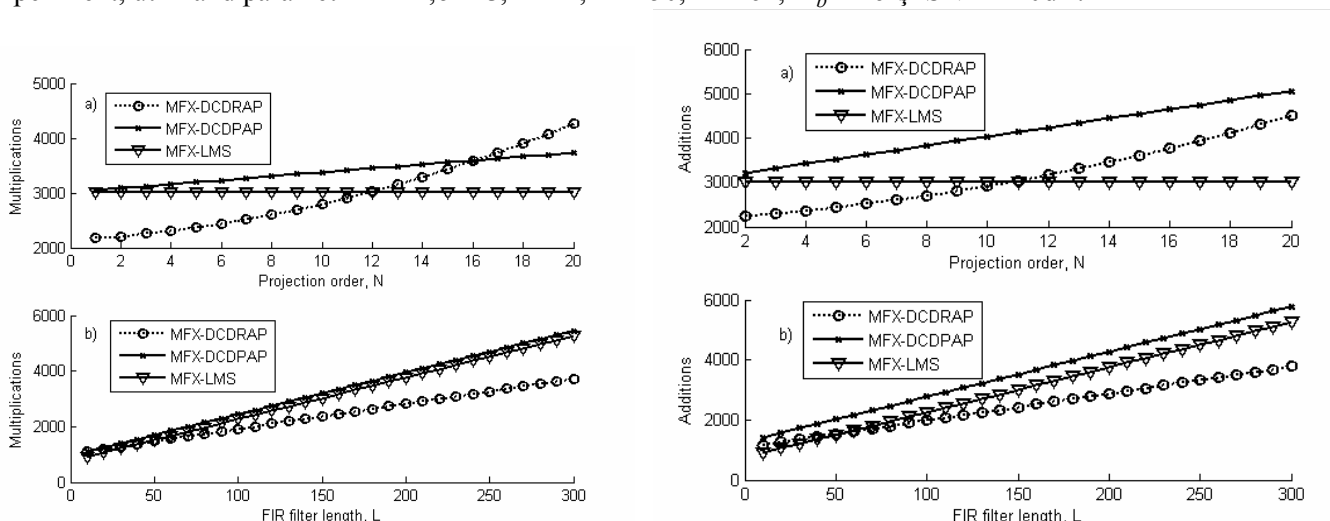


Fig. 1. Numărul de înmulțiri (stânga) și adunări (adunări) pentru algoritmi MFX-LMS, MFX-DCDRAP și MFX-DCDPAP în două situații: (a) N variabil ($I = 1, J = 3, K = 2, L = 150, M = 64, N_u = 4, M_b = 16$), (b) L variabil ($I = 1, J = 3, K = 2, N = 5, M = 64, N_u = 4, M_b = 16$)

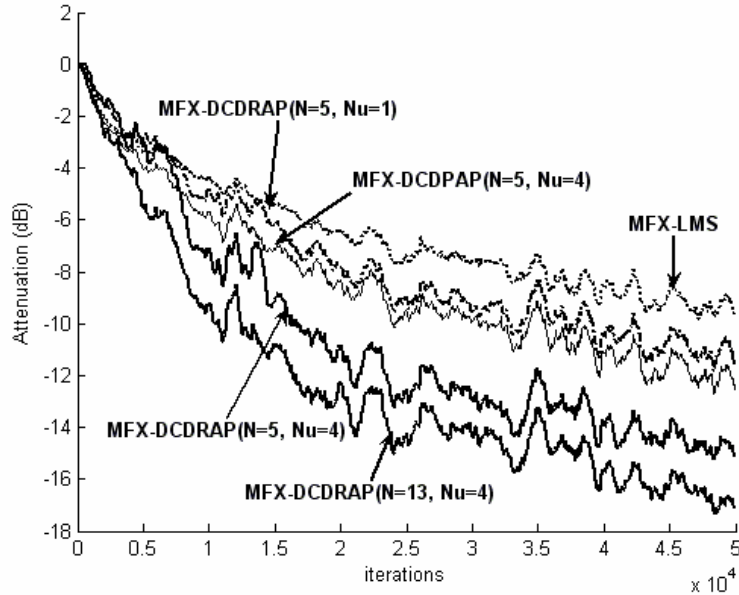


Fig. 2. Convergența algoritmilor MFX-LMS, MFX-DCDRAP și MFX-DCDPAP în cadrul aplicației ANC multi-canal.

În cadrul algoritmilor bazați pe proiecții afine, pasul de adaptare al algoritmului reprezintă un parametru cheie, deoarece stă la baza realizării compromisului între criteriile de performanță (viteză de convergență și capacitate de urmărire *versus* dezadaptare și robustețe). Algoritmii de adaptare cu pas variabil (*VSS – variable step-size*) urmăresc rezolvarea acestei probleme. Soluția propusă în cadrul acestui proiect de cercetare are ca rezultat o formulă *non-parametrică* a pasului de adaptare, ceea ce constituie o facilitate importantă pentru aplicabilitatea sa practică [12]-[17]. Algoritmii propuși, de tip FAP/PAP ce au la bază metoda de reducere a complexității GS și pas de adaptare variabil (VSS) au fost prezentați în [18]. Algoritmii similari, bazați de metoda DCD au fost prezentați în [19]. Spre exemplificare, în dezvoltarea algoritmilor bazați pe metoda GS sunt utilizate următoarele notații:

- L este lungimea filtrului adaptive;
- K este ordinal de proiecție al APA;
- δ este factorul de regularizare;
- λ este un factor de ponderare;
- \mathbf{I}_K este matricea identitate de dimensiune $K \times K$ identity matrix;
- $\mu(n)$ este pasul de adaptare variabil al algoritmului;
- $\mathbf{x}(n) = [x(n), x(n-1), \dots, x(n-L+1)]^T$ este vectorul semnalului de intrare la momentul de timp n (operatorul T semnifică operația de transpunere);
- $\mathbf{R}(n)$ este matricea de autocorelație a semnalului de intrare (de dimensiune $K \times K$);
- $\xi(n) = [x(n), x(n-1), \dots, x(n-K+1)]^T$ este un vector de dimensiune $K \times 1$;
- $\mathbf{u}(n) = [u(n), \dots, u(n-L+1)]^T$ este vectorul aproximat;
- $\bar{\mathbf{u}}(n)$ conține primele $L-1$ elemente ale vectorului $\mathbf{u}(n)$;
- $\mathbf{b} = \begin{bmatrix} 1 & \mathbf{0}_{(K-1) \times 1}^T \end{bmatrix}^T$;
- $\hat{\mathbf{h}}(n) = [\hat{h}_0(n), \dots, \hat{h}_{L-1}(n)]^T$ este vectorul coeficienților filtrului adaptive;
- $\mathbf{w}(n) = [w_0(n), \dots, w_{L-1}(n)]^T$ este vectorul coeficienților filtrului auxiliar;
- $\mathbf{r}(n)$ este vectorul de autocorelației (de dimensiune $K \times 1$);
- $\tilde{\mathbf{r}}(n)$ conține ultimele $K-1$ elemente ale vectorului $\mathbf{r}(n)$;
- $\mathbf{p}(n)$ este vectorul soluție al problemei LP (de dimensiune $K \times 1$);
- $p_i(n)$ (cu $i = 1, \dots, K$) reprezintă elementul i din vectorul $\mathbf{p}(n)$;
- $\mathbf{e}(n)$ este un vector de dimensiune $K \times 1$;
- $e_K(n)$ reprezintă ultimul element al vectorului $\mathbf{e}(n)$;

- $\bar{\mathbf{e}}(n)$ conține primele $K-1$ elemente ale vectorului $\mathbf{e}(n)$.

Matricea $\mathbf{R}(n)$ este reactualizată prin înlocuirea primul rand și al primei coloane cu elementele vectorului $\mathbf{r}(n)$, iar cealaltă parte a matricei [de dimensiune $(K-1) \times (K-1)$] este înlocuită cu matricea de dimensiune $(K-1) \times (K-1)$ rezultată din primele $K-1$ linii și coloane ale matricei $\mathbf{R}(n-1)$.

Pașul de adaptare variabil pentru algoritmi VSS-GS-PAP și VSS-GSFAP este calculat pe baza următoarelor relații:

$$\hat{\sigma}_s^2(n) = \lambda \hat{\sigma}_s^2(n-1) + (1-\lambda) [d^2(n) - \hat{y}^2(n)] \quad (1)$$

$$\hat{\sigma}_e^2(n) = \lambda \hat{\sigma}_e^2(n-1) + (1-\lambda) e^2(n) \quad (2)$$

$$\mu(n) = \left| 1 - \frac{\hat{\sigma}_s^2(n)}{\varepsilon + \hat{\sigma}_e^2(n)} \right| \quad (3)$$

unde $\hat{\sigma}_e^2(0) = 0$, $\hat{\sigma}_s^2(0) = 0$, ε este o constantă pozitivă de valoare foarte mică, iar λ este un factor de ponderare.

Relațiile care definesc algoritmul VSS-GS-PAP propus sunt următoarele:

Inițializare

$$\mathbf{x}(0) = \mathbf{0}_{L \times 1}, \mathbf{R}(0) = \delta \mathbf{I}_K, \mathbf{u}(0) = \mathbf{0}_{L \times 1}, \hat{\mathbf{h}}(0) = \mathbf{0}_{L \times 1}, \quad (4)$$

$$\mathbf{p}(0) = \mathbf{0}_{K \times 1}, \xi(0) = \mathbf{0}_{K \times 1}, \mathbf{r}(0) = \left[\delta, \mathbf{0}_{(K-1) \times 1}^T \right]^T$$

For $n = 1, 2, \dots$

$$\mathbf{r}(n) = \mathbf{r}(n-1) + x(n)\xi(n) - x(n-L)\xi(n-L) \quad (5)$$

$$\text{reactualizează } \mathbf{R}(n) \text{ utilizând } \mathbf{r}(n) \quad (6)$$

$$\text{rezolvă } \mathbf{R}(n)\mathbf{p}(n) = \mathbf{b} \text{ (utilizând o iterație Gauss-Seidel)} \quad (7)$$

$$\mathbf{u}(n) = \left[\xi^T(n)\mathbf{p}(n)/p_1(n) \quad \bar{\mathbf{u}}^T(n-1) \right]^T \quad (8)$$

$$\hat{y}(n) = \mathbf{x}^T(n)\hat{\mathbf{h}}(n-1) \quad (9)$$

$$e(n) = d(n) - \hat{y}(n) \quad (10)$$

calculează $\mu(n)$ conform (1)–(3)

$$\bar{\mathbf{e}}(n) = \frac{\mu(n)e(n)}{\delta + \mathbf{u}^T(n)\mathbf{x}(n)} \quad (11)$$

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mathbf{u}(n)\bar{\mathbf{e}}(n) \quad (12)$$

Algoritmul VSS-GS-PAP algorithm necesită $2L + K^2 + 3K + 10$ multiplicări, 4 împărțiri și o operație de extragere a radicalului.

Algoritmul VSS-GSFAP propus este definit de următoarele relații:

Inițializare

$$\mathbf{x}(0) = \mathbf{0}_{L \times 1}, \mathbf{R}(0) = \delta \mathbf{I}_K, \mathbf{e}(0) = \mathbf{0}_{K \times 1}, \mathbf{w}(0) = \mathbf{0}_{L \times 1}, \quad (13)$$

$$\mathbf{p}(0) = \mathbf{0}_{K \times 1}, \xi(0) = \mathbf{0}_{K \times 1}, \mathbf{r}(0) = \left[\delta, \mathbf{0}_{(K-1) \times 1}^T \right]^T, \mu(0) = 1$$

For $n = 1, 2, \dots$

$$\mathbf{r}(n) = \mathbf{r}(n-1) + x(n)\xi(n) - x(n-L)\xi(n-L)$$

reactualizează $\mathbf{R}(n)$ utilizând $\mathbf{r}(n)$

rezolvă $\mathbf{R}(n)\mathbf{p}(n) = \mathbf{b}$ (utilizând o iterație Gauss-Seidel)

$$\hat{y}(n) = \mathbf{x}^T(n)\mathbf{w}(n-1) + \mu(n-1)\tilde{\mathbf{r}}(n)\bar{\mathbf{e}}(n-1) \quad (14)$$

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

$$\mathbf{e}(n) = \left[0 \quad \bar{\mathbf{e}}^T(n-1) \right]^T + e(n)\mathbf{p}(n) \quad (15)$$

calculează $\mu(n)$ conform (1)–(3)

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \mu(n)\mathbf{x}(n-K+1)e_K(n) \quad (16)$$

Algoritmul VSS-GSFAP necesită $2L + K^2 + 4K + 10$ multiplicări, 2 împărțiri și o operație de extragere a radicalului. Relațiile necesare pentru calculul pasului de adaptare [(1)–(3)] necesită 9 multiplicări, 6 adunări, o operație de extragere a radicalului și o împărțire. Prin urmare, complexitatea aritmetică a algoritmilor propuși (VSS-GSFAP și VSS-GS-PAP) este comparativă cu cea a algoritmilor cu pas de adaptare fix (GS-FAP/PAP).

Pentru comparație s-a ales algoritmul VSS-PAP [20], care utilizează o strategie diferită de evaluare a pasului, pe baza minimizării MSD (*mean-square deviation*). Complexitatea algoritmului VSS-PAP este de $2L + K^2 + 5K + 12$ multiplicări și împărțiri. În Fig. 3–6 sunt prezentate rezultatele simulărilor efectuate într-un scenariu de compensare a ecoului acustic. Acestea indică superioritatea algoritmilor propuși, VSS-GSFAP și VSS-GS-PAP.

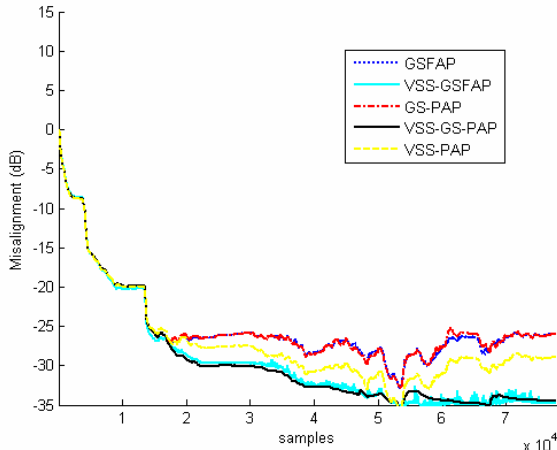


Fig. 3. Dezalinierea (misalignment) pentru algoritmi GSFAP, GS-PAP, VSS-GSFAP, VSS-PAP și VSS-GS-PAP. Cazul single-talk, $L = 512$, $K = 4$, SNR = 30dB.

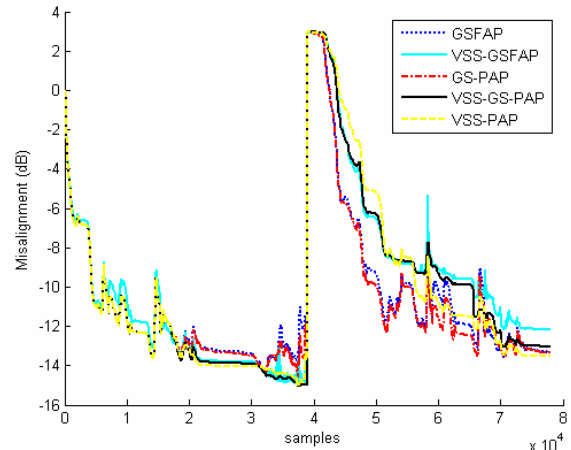


Fig. 5. Dezalinierea (misalignment) pentru algoritmi GSFAP, GS-PAP, VSS-GSFAP, VSS-PAP și VSS-GS-PAP. Cazul single-talk, $L = 512$ (cale de ecou 1024 - submodele), $K = 4$, SNR = 30dB, schimbarea căii de ecou.

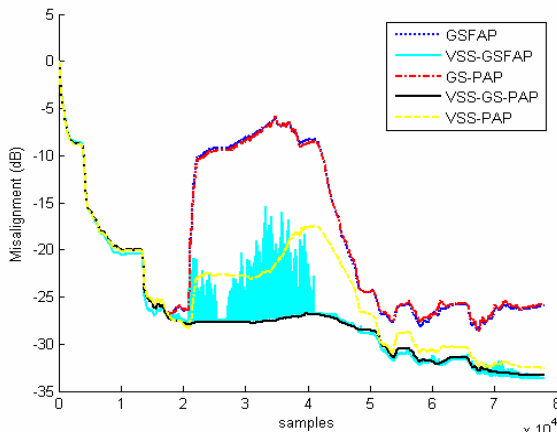


Fig. 4. Dezalinierea (misalignment) pentru algoritmi GSFAP, GS-PAP, VSS-GSFAP, VSS-PAP și VSS-GS-PAP. Cazul single-talk, $L = 512$, $K = 4$, SNR variabil (30dB-10dB).

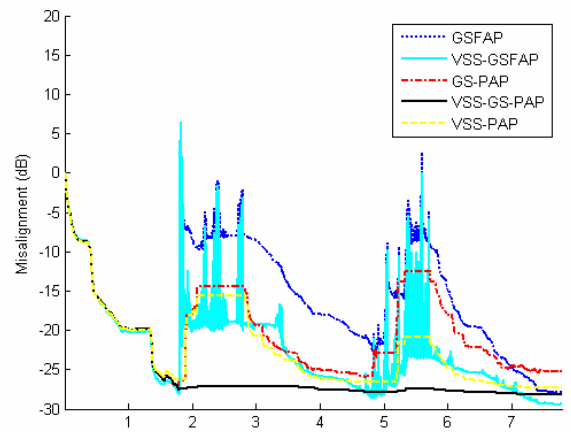


Fig. 6. Dezalinierea (misalignment) pentru algoritmi GSFAP, GS-PAP, VSS-GSFAP, VSS-PAP și VSS-GS-PAP. Cazul double-talk (cu Geigel DTD), $L = 512$, $K = 4$, SNR = 30dB.

Performanțele globale ale oricărei structuri ANC multi-canal sunt puternic influențate de performanțele sistemului de microfoane utilizate [21]–[28]. Acesta constă într-un set de microfoane (senzori) distribuite într-un mod spațial, astfel încât să se asigure caracteristici de directivitate optime. Una din cele mai importante funcții ale sistemului de microfoane este să extragă semnalul de interes dintr-o sumă de alte semnale. Metoda tipică este să se formeze un lob al caracteristicii de directivitate și să se îndrepte pe direcția dorită. Ca rezultat, semnalele de pe direcția favorizată (“privată”) sunt amplificate, iar celelalte atenuate. Domeniul de cercetare al ariilor de microfoane cu lobi programabili s-a dezvoltat foarte mult în ultimii zece ani și au apărut numeroși algoritmi de tip MAB (*Microphone-Array Beamforming*), algoritmi care deși au rezultate teoretice foarte bune oferă performanțe limitate în mediile acustice reale. Acest comportament nu a fost pe deplin înțeles de cercetători astfel încât este nevoie de o cercetare mai amplă a acestui domeniu. Una dintre abordările apărute recent consideră rețeaua de microfoane ca un sistem MIMO și studiază performanțele acestuia în îmbunătățirea semnalului.

O arie de microfoane e formată din seturi de microfoane amplasate în spațiu după diferite modele. Acestea au fost studiate de mai bine de trei decenii și au un rol important în aplicațiile practice care necesită achiziția la distanță a semnale audio, aplicații precum „audio-bridging” și teleconferințe. În astfel de aplicații cea mai importantă funcție a

ariei de microfoane este extragerea vorbirii din semnalul achiziționat care poate conține și zgomot, reverberații și surse de vorbire multiple. Modul cel mai simplu de realizare a extragerii vorbirii constă în formarea unei caracteristici de tip lob și îndreptarea acesteia într-o direcție dorită. Astfel semnalele venite din această direcție sunt îmbunătățite (amplificate) în timp ce semnalele care vin din alte direcții sunt atenuate.

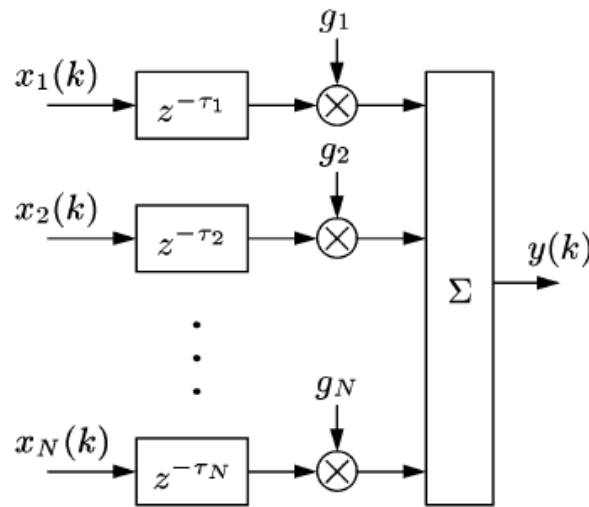


Fig. 7. Structura unui „beamformer” cu întârziere și însumare.

Pentru a analiza performanțele acestei metode presupunem că avem un șir de N microfoane ale căror ieșiri sunt notate $x_n(k)$ ($n=1,2,\dots,N$). Formarea lobului este realizată prin manipularea semnalelor $x_n(k)$. Cel mai simplu algoritm de „beamforming” folosește operații de întârziere și însumare (Fig. 7). Acest algoritm, studiat inițial în cercetarea subacvatică și a antenelor radar, se bazează pe întârzierea ieșirii fiecărui microfon cu un anumit interval de timp astfel încât componentele semnalului provenit de la sursa dorită și captat de toți senzorii vor fi sincronizate. Aceste semnale întârziate sunt ponderate și însumate. Datorită sincronizării, prin însumare componentele semnalului dorit vor fi amplificate, iar zgomotul sau semnalele provenite de la alte surse vor fi atenuate sau chiar eliminate. Coeficienții g_n utilizați pentru pondere pot avea valori fixe sau pot fi determinați adaptiv. În cazul determinării adaptive a coeficienților metodele sunt cunoscute ca metode de „beamforming” adaptiv. Intuitiv este evident că o determinare adaptivă a coeficienților va avea rezultate mai bune. Privind Fig. 7 putem demonstra acest lucru într-un mod foarte simplu. Din schema prezentată în figură se observă că la o anumită frecvență putem obține $N-1$ zerouri pe baza modelului de directivitate. Pe baza caracteristicilor semnalului și a zgomotului se pot determina adaptiv coeficienții astfel încât zerourile să fi definite și aranjate corespunzător. O astfel de determinare și aranjare a zerourilor va avea ca efect o eliminare mai bună a zgomotului și a interferențelor. Această metodă a fost definită și este utilă pentru semnale de bandă îngustă. Pentru aplicațiile acustice nu este foarte utilă deoarece vorbirea este un semnal de bandă largă. De asemenea, modelul de directivitate al formatorului de undă nu ar fi același pentru o bandă largă de frecvențe. Dacă este utilizat un astfel de „beamformer” atunci zgomotul și interferențele provenite din alte direcții decât cea selectată nu vor fi atenuate uniform în tot spectrul de frecvențe (*spectral tilt*).

Un mod simplu de eliminare a acestei probleme este utilizarea de subrețele grupate armonice (*harmonically nested subarrays*), fiecare subrețea fiind proiectată să funcționeze la o singură frecvență. Însă o astfel de soluție necesită un număr mare de microfoane și o geometrie a modelului de directivitate ieșită din comun. O altă abordare implică descompunerea semnalului de bandă largă în semnale de bandă îngustă și proiectarea de „beamformers” independente pentru fiecare frecvență. O schemă a acestei abordări este prezentată în Fig. 8. Această schemă este echivalentă cu filtrarea ieșirii fiecărui microfon cu un filtru cu răspuns finit la impuls de durată finită și apoi însumarea semnalelor filtrate (Fig. 9). Eficacitatea acestui tip de sistem poate fi explicată astfel: prin folosirea a N filtre FIR de lungime L putem obține $N-1$ zerouri la $L-1$ frecvențe diferite. Reamintim că în cazul sistemului cu întârziere și însumare se obțineau $N-1$ zerouri pentru aceeași frecvență. La fel ca în cazul precedent și pentru acest sistem coeficienții filtrelor din Fig. 9 pot fi determinați adaptiv sau neadaptiv. O abordare adaptivă nu este totdeauna foarte robustă și poate avea un efect nedorit: eliminarea semnalului util.

Deși s-au depus multe eforturi pentru studierea ariilor de microfoane, cele mai multe „beamformere” nu au rezultatele așteptate în medii acustice reale. De aceea domeniul este încă deschis cercetătorilor și cei mai mulți dintre aceștia consideră că potențialul maxim al utilizării ariilor de microfoane nu a fost atins. Una dintre abordările adoptate recent în studiul performanțelor ariilor de microfoane este tratarea acestora ca sisteme de tip MIMO (*Multiple-Input Multiple-Output*). Această abordare este studiată intensiv de grupul nostru de cercetare în cadrul acestui proiect. În continuare sunt prezentate pe scurt rezultatele obținute în urma cercetării acestui domeniu.

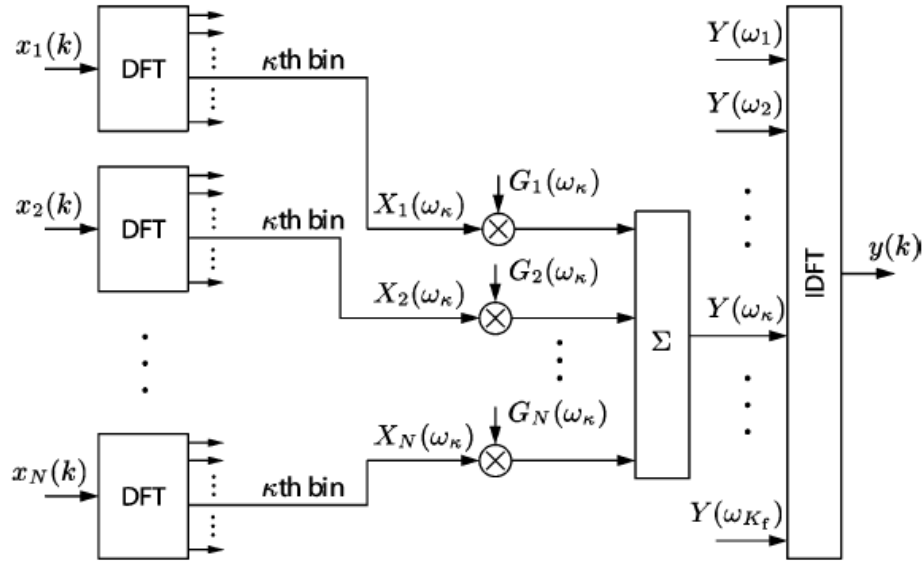


Fig. 8. Structura unui sistem cu lobi programabili de bandă largă care utilizează descompunerea în semnale de bandă îngustă.

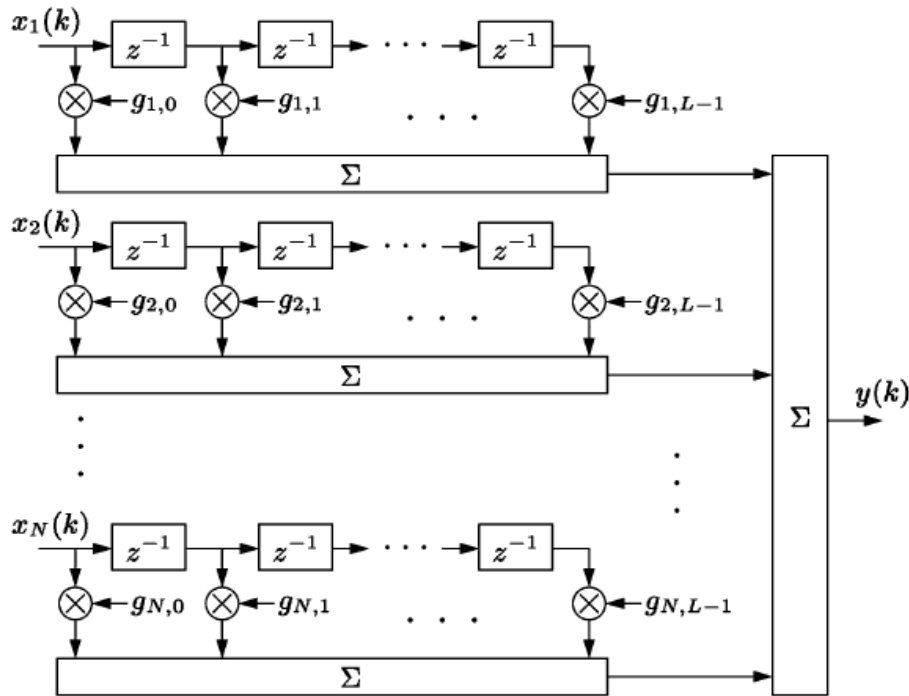


Fig. 9. Structura unui sistem cu filtrare și însumare.

Problema ariilor de microfoane cu lobi programabili este ilustrată în Fig. 10. Pe scurt putem descrie această problemă astfel: există M surse de sunet și $N-1$ microfoane, iar numărul de microfoane este mai mare sau egal cu numărul de surse de sunet. Semnalul de la ieșirea microfonului cu numărul n va avea expresia:

$$x_n(k) = \sum_{m=1}^M h_{nm} \otimes s_m(k) + b_n(k), \quad n = 1, 2, \dots, N \quad (17)$$

unde \otimes reprezintă operația de convoluție, $s_m(k)$ este sursa de semnal numărul m , h_{nm} este răspunsul la impuls al canalului acustic dintre sursa m și microfonul n , și $b_n(k)$ este zgomotul captat de microfonul n .

Putem rescrie ecuația de mai sus sub formă matriceală astfel:

$$x_a(k) = Hs(k) + b(k) \quad (18)$$

unde

$$\mathbf{x}(k) = [x_1(k) \ x_2(k) \ \dots \ x_N(k)]^T$$

$$\mathbf{H} = [\mathbf{H}_1 \ \mathbf{H}_2 \ \dots \ \mathbf{H}_M]$$

$$\mathbf{H}_m = \begin{bmatrix} \mathbf{h}_{1m}^T \\ \mathbf{h}_{2m}^T \\ \vdots \\ \mathbf{h}_{Nm}^T \end{bmatrix}$$

$$\mathbf{h}_{nm}^T = [h_{nm,0} \ h_{nm,1} \ \dots \ h_{nm,L_h-1}]^T, \quad n=1,2,\dots,N \quad m=1,2,\dots,M$$

$$\mathbf{b}(k) = [b_1(k) \ b_2(k) \ \dots \ b_N(k)]^T$$

$$\mathbf{s}(k) = [s_1^T(k) \ s_2^T(k) \ \dots \ s_M^T(k)]^T$$

$$s_m(k) = [s_m(k) \ s_m(k-1) \ \dots \ s_m(k-L_h+1)]^T$$

L_h reprezintă lungimea celui mai lung răspuns la impuls al canalelor.

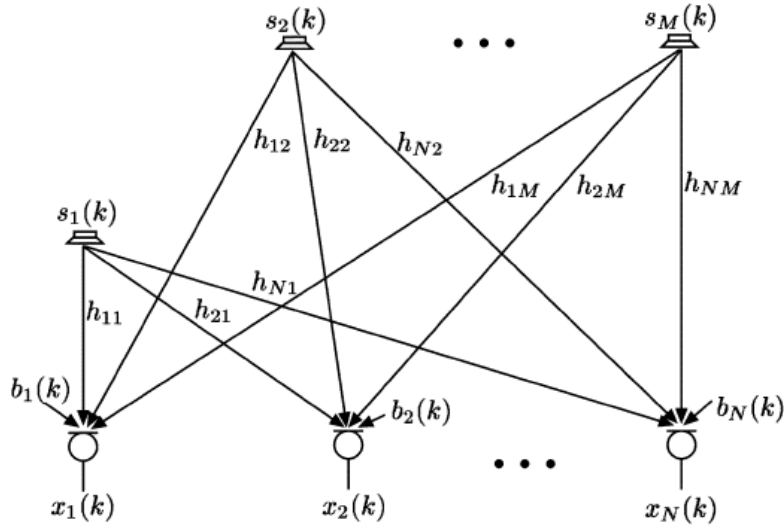


Fig. 10. Exemplu de arie de microfoane.

Dat fiind modelul de semnal de mai sus, programarea ariei de microfoane constă în estimarea unora dintre cele M surse de semnal. Pentru a simplifica prezentarea modului de estimare a semnalelor vom neglija termenii zgomotului $\mathbf{b}(k)$ din relația (18). Astfel, ieșirea microfonului cu numărul n la momentul k va fi:

$$x_n(k) = \sum_{m=1}^M \mathbf{h}_{nm}^T \mathbf{s}_m(k), \quad n=1,2,\dots,N \quad (19)$$

Să presupunem că dintre cele M surse de semnal existente avem P semnale pe care dorim să le estimăm. Putem presupune că primele P semnale, adică $s_p(k)$ cu $p=1,2,\dots,P$, sunt semnalele dorite, iar restul de Q surse de semnal $s_{p+q}(k)$ cu $q=1,2,\dots,Q$ sunt interferențe, unde $P+Q=M$. Obiectivul programării ariei constă în extragerea semnalelor dorite $s_p(k)$ din semnalele captate de microfoane $x_n(k)$ cu $n=1,2,\dots,N$. Acest lucru presupune două operații de procesare: dereverberație și suprimarea interferențelor. Să presupunem că putem obține un estimat al $s_p(k)$ prin aplicarea a N filtre celor N ieșiri ale microfoanelor, adică:

$$y_p(k) = \sum_{n=1}^N \mathbf{g}_{pn}^T \mathbf{x}_n(k), \quad p=1,2,\dots,P \quad (20)$$

unde $\mathbf{g}_{pn} = [\mathbf{g}_{pn,0} \ \mathbf{g}_{pn,1} \ \dots \ \mathbf{g}_{pn,L_g-1}]^T$, $p=1,2,\dots,P$ sunt $P \cdot N$ filtre de lungime L_g și

$$\mathbf{x}_n(k) = [x_n(k) \ x_n(k-1) \ \dots \ x_n(k-L_g+1)]^T, \quad n=1,2,\dots,N.$$

Pentru a rezolva problema estimării semnalelor dorite $s_p(k)$ trebuie determinați coeficienții g_{pn} astfel încât $y_p(k) = s_p(k - \tau_p)$, unde τ_p este o constantă de întârziere. Dintre metodele de determinare a acestor coeficienți se pot remarca: metodele *Least-Squares* și MINT, algoritmul Frost, structura generalizată de eliminare a lobului secundar și abordarea bazată pe răspunsul fără distorsiuni cu varianță minimă.

Platforma experimentală a avut la bază o platformă FPGA Xilinx Virtex 5 XC5VFX70 [29], [30]. Această tehnologie de implementare logică digitală, matricea de porți programabilă, FPGA (*Field Programmable Gate Array*), a cunoscut o răspândire tot mai mare în ultimul timp. Avantajul principal pe care îl oferă FPGA-urile, precum și celelalte componente ale familiei circuitelor logice programabile, cum ar fi PLA și EEPROM, constă tocmai în programabilitatea lor. Aceasta implică posibilitatea de a configura un anumit circuit pentru o sarcină specifică, și, în cele mai multe cazuri, chiar de a reprograma același circuit pentru o utilizare ulterioară diferită. Datorită evoluției caracteristicilor acestui tip de circuite, o tendință foarte puternică în prezent este aceea de a exploata la maxim caracterul de reprogramabilitate al acestor circuite, și în special a FPGA-urilor, prin includerea lor în sisteme reprogramabile. La ora actuală, pentru programarea circuitelor logice se folosesc, în mod curent, limbaje de descriere a hardware-ului (*Hardware Description Language*, HDL), dintre care cele mai răspândite sunt VHDL și Verilog. VHDL (VHSIC HDL — VHSIC însemnând *Very High Speed Integrated Circuits*) a devenit standard IEEE în anul 1986 și se înrudește, din punct de vedere al sintaxei, cu limbajul de programare Ada, din care este inspirat de altfel și limbajul Pascal. Verilog a fost standardizat de IEEE în 1995, fiind asemănător la nivel general cu limbajul C, ceea ce poate fi considerat drept un argument privind ușurința învățării sale.

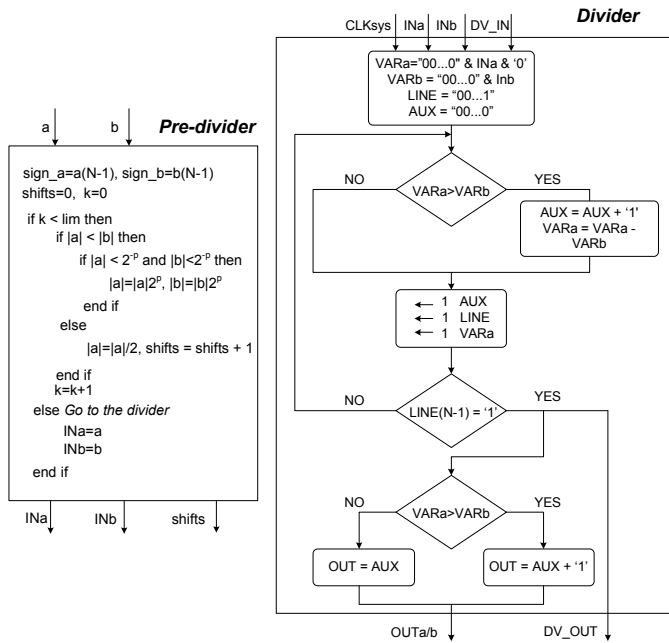


Fig. 11. Procedura de “pre-divider” și schema bloc.

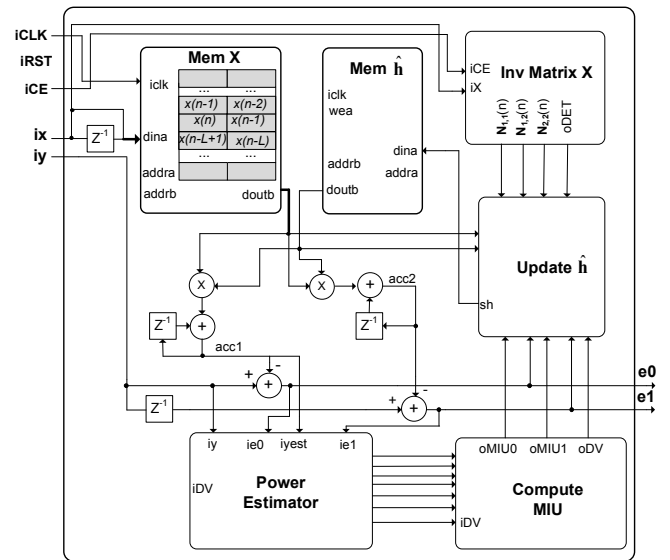


Fig. 13. Schema bloc AEC.

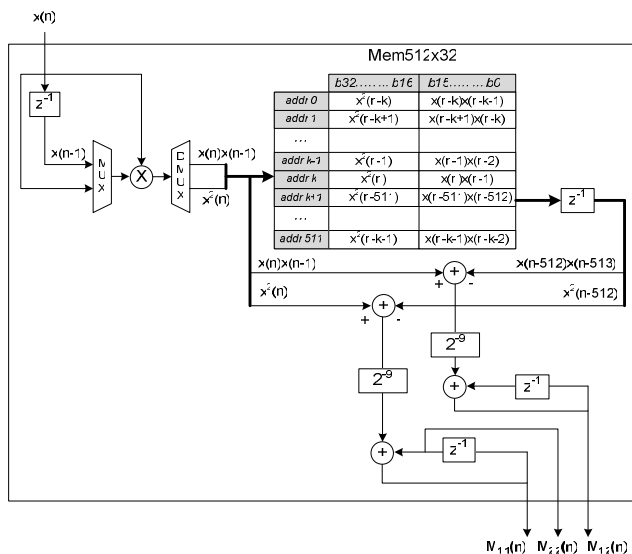


Fig. 12. Diagrama de inversare a matricei.

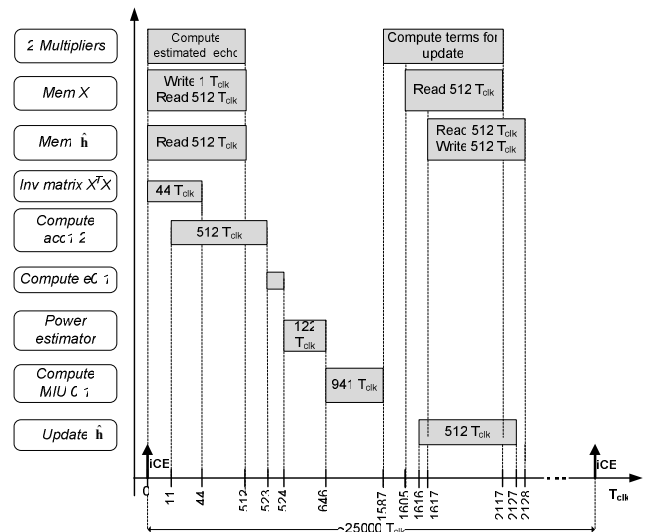


Fig. 14. Diagrama de timp.

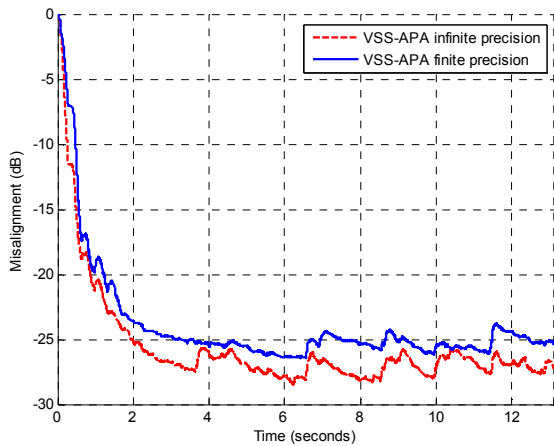


Fig. 15. Efectele cuantizării asupra dezalinierei algoritmului.

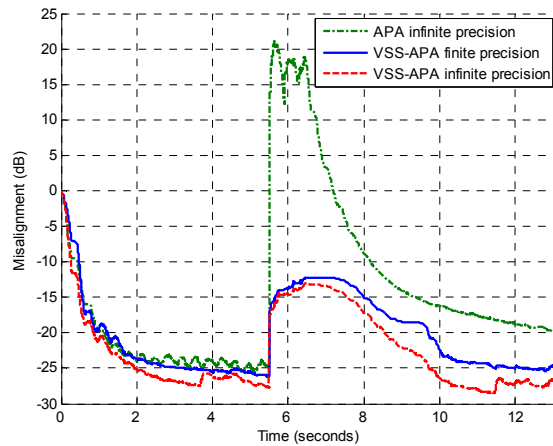


Fig. 16. Dezalinierea APA și VSS-APA. Cazul double-talk, fără DTD.

Programarea convențională a FPGA-urilor poate fi descrisă prin următoarea succesiune de pași: 1) etapa de editare, 2) etapa de simulare, 3) etapa de sinteză, 4) etapa de mapare/rutare, 5) etapa de configurare, 6) etapa de verificare post-implementare și eventual modificarea sau corectarea circuitului.

În etapa de editare, schema digitală este creată cu ajutorul unui editor sau a unui limbaj de descriere a hardware-ului. Un editor de scheme permite folosirea de simboluri grafice pentru precizarea componentelor și a circuitelor. Limbajele de tip HDL folosesc un limbaj descriptiv, cum ar fi Verilog, ABEL sau VHDL, în același scop. Orice variantă s-ar alege trebuie verificat că programul respectiv are încorporată biblioteca specifică FPGA-ului ce va fi folosit, deoarece ele trebuie să poată produce și mapa, în etapa de sinteză, *netlist*-ul asociat proiectului.

Etapa de mapare/rutare realizează convertirea *netlist*-ului produs în faza anterioară într-un fișier binar de configurare a FPGA-ului. Cei trei pași incluși în această etapă sunt: a) maparea schemei pe resursele circuitului; b) asignarea blocurilor logice, create anterior prin mapare, unor locații specifice din circuit; c) interconectarea blocurilor logice.

În final se crează o descriere exactă a configurației FPGA-ului (*Logic Cell Array File LCA*), care este convertită într-o reprezentare binară. Această reprezentare binară este ulterior folosită în etapa de verificare, care are drept scop testarea proiectului din punct de vedere logic și al sincronizării/ temporizării. O serie dintre aceste teste se pot efectua cu instrumentele de verificare/simulare, care permit o caracterizare detaliată a proiectului, prin efectuarea de simulări de funcționalitate și sincronizare. Un alt mod de testare este chiar verificarea sub tensiune, prin crearea condițiilor normale de operare. Uneori, după etapa de mai sus se pot constata inadecvări între funcționalitatea circuitului și cea dorită. În acest caz se impune reluarea ciclului de proiectare de la primul pas.

Configurarea este procesul prin care fișierul binar este propriu-zis încărcat în FPGA. Circuitele pot fi configurate prin folosirea unui PROM, cel serial fiind cel mai folosit, dar existând și variante paralele la nivel de octet. În acest caz FPGA-ul își citește în mod activ configurația din PROM-ul respectiv. O a doua posibilitate este ca datele să fie înscrise în FPGA dintr-o sursă exterioară (de obicei un PC), prin intermediul unei interfețe, cum ar fi JTAG, ce se conectează la portul paralel al unui PC printr-un cablu ByteBlaster.

Circuitele FPGA oferă următoarele avantaje față de tehnologiile alternative: 1) permit o reducere semnificativă a ciclului de proiectare și producție și 2) asigură o reducere a costului de producție al circuitelor VLSI (*Very Large Scale Integration*). Aceste avantaje, care se datorează programabilității de către utilizator a circuitelor, asigură o reducere a duratei de proiectare, deoarece se pot realiza într-un timp scurt iterații multiple de proiectare. Totuși, programabilitatea de către utilizator are și dezavantaje : densitatea logicii și performanțele de viteză ale circuitelor FPGA sunt considerabil mai reduse decât ale celorlalte alternative. Deși îmbunătățirile din ultimii ani au permis creșterea performanțelor circuitelor, sunt necesare încă eforturi de cercetare pentru a dezvolta arhitecturi optime pentru circuitele FPGA.

Rezultatele implementării sistemului AEC au fost prezentate detaliat în [31], [32]. Pricipalele blocuri de calcul și diagrama de timp sunt reprezentate în figurile 11-14. Rezultatele obținute sunt reprezentate în figurile 15 și 16, atât în cazul single-talk, cât și în double-talk. În primul rând, se poate remarca robustețea numerică algoritmului în ceea ce privește implementarea în precizie finită. Se observă din Fig. 15 că diferența față de precizia infinită (simularea în Matlab) este sub 2 dB. În al doilea rând, algoritmul își păstrează comportarea foarte bună și în cazul double-talk, fără a folosi un DTD. Utilizarea unui simplu DTD Geigel conduce la creșterea performanțelor algoritmului. Avantajul major al soluțiilor propuse constă în faptul că evaluarea parametrilor de convergență variabili nu necesită informații a priori legate de mediul în care operează sistemul adaptiv, acest aspect reprezentând o noutate în domeniu. Datorită acestei caracteristici specifice, algoritmi propuși sunt deosebit de adecvați utilizării în cadrul aplicațiilor practice.

Bibliografie

- [1] S.J. Elliot, P.A. Nelson, "Active noise control", *IEEE Signal Process. Mag.*, vol. 10, no. 4, pp. 12-35, Oct. 1993.
- [2] M. Bouchard, "Multichannel affine and fast affine projection algorithms for active noise control and acoustic equalization systems", *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 11, no. 1, pp. 54-60, Jan. 2003.
- [3] M. Bouchard, F. Albu, "The Gauss-Seidel fast affine projection algorithm for multichannel active noise control and sound reproduction systems", *Int. Journal Adapt. Control Signal Process., Special Issue Adaptive Control Sound and Vibration*, vol. 19, no. 2-3, pp. 107-123, Mar.-Apr. 2005.
- [4] R. Barret et al., *Templates for the Solutions of Linear Systems: Building Blocks for Iterative Models*, 2nd ed., SIAM, Philadelphia, 1994.
- [5] F. Albu, M. Bouchard, "The Gauss-Seidel pseudo-affine projection algorithm and its application for echo cancellation", *Proc. Asilomar*, Pacific Grove, CA, vol. 2, pp. 1303-1306, Nov. 2003.
- [6] F. Albu, M. Bouchard, Y.V. Zakharov, "Pseudo-affine projection algorithms for multichannel active noise control", *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 1044-1052, Mar. 2007.
- [7] Y. V. Zakharov, T. C. Tozer, "Multiplication-free iterative algorithm for LS problem", *Electronics Letters*, vol. 40, no. 9, pp. 567-569, April 2004.
- [8] F. Albu, "Efficient multichannel filtered-x affine projection algorithm for active noise control," *Electronics Letters*, vol. 42, no. 7, pp. 421 – 423, 2006.
- [9] Y. V. Zakharov, "Low complexity implementation of the affine projection algorithm," *IEEE Signal Processing Letters*, April 2008.
- [10] F. Albu, C. Paleologu, "A Recursive Least Square Algorithm for Active Noise Control based on the Gauss-Seidel Method", *Proc. IEEE Int. Conf. on Electronics, Circuits, and Systems (ICECS)*, pp. 830-833, Malta, 2008.
- [11] F. Albu, C. Paleologu, "New multichannel modified filtered-x algorithms for active noise control using the dichotomous coordinate descent method", *Proc. Acoustics*, pp. 5721-5725, Paris, Franța, 2008.
- [12] C. Paleologu, S. Ciochină, J. Benesty, "Variable Step-Size NLMS Algorithm for Under-Modeling Acoustic Echo Cancellation", *IEEE Signal Processing Letters*, vol. 15, pp. 5-8, 2008.
- [13] C. Paleologu, J. Benesty, S. Ciochină, "A Variable Step-Size Affine Projection Algorithm Designed for Acoustic Echo Cancellation", *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 8, pp. 1466-1478, Nov. 2008.
- [14] C. Paleologu, J. Benesty, S. Ciochină, "A Robust Variable Forgetting Factor Recursive Least-Squares Algorithm for System Identification", *IEEE Signal Processing Letters*, vol. 15, pp. 597-600, 2008.
- [15] C. Paleologu, J. Benesty, S. Ciochină, "A Variable Step-Size Proportionate NLMS Algorithm for Echo Cancellation", *Revue Roumaine des Sciences Techniques – Serie Electrotechnique et Energetique*, vol. 53, no. 3, pp. 309-317, 2008.
- [16] C. Paleologu, S. Ciochină, J. Benesty, "An Efficient Proportionate Affine Projection Algorithm for Echo Cancellation", *IEEE Signal Processing Letters*, vol. 17, no. 2, pp. 165-168, Feb. 2010.
- [17] C. Paleologu, J. Benesty, S. Ciochină, "A Family of Variable Step-Size NLMS Algorithms for Echo Cancellation", *Revue Roumaine des Sciences Techniques – Serie Electrotechnique et Energetique*, vol. 55, no. 1, pp. 49-58, 2010.
- [18] F. Albu, C. Paleologu, J. Benesty, "Gauss-Seidel based Variable Step-Size Affine Projection Algorithms for Acoustic Echo Cancellation", *Proc. European Signal Processing Conference (EUSIPCO)*, pp. 2007-2011, Glasgow, Scoția, Aug. 2009.
- [19] F. Albu, C. Paleologu, J. Benesty, Y. V. Zakharov, "Variable Step Size Dichotomous Coordinate Descent Affine Projection Algorithm", *Proc. IEEE EUROCON*, pp. 1366-1371, St. Petersburg, Rusia, 2009.
- [20] L. Lee, Y.-C. Park, D.-H. Youn, "Robust pseudo affine projection algorithm with variable step-size," *Electronics Letters*, vol. 44, pp. 250–252, Jan. 2008.
- [21] M. Branstein, D.B. Wards (ed.), *Microphone Arrays: Signal Processing Techniques and Applications*, Springer, Germany, 2001.
- [22] H. Cox, R.M. Zeskind, M.M. Owen, "Robust adaptive beamforming," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-35, no. 10, pp. 1365-1376, Oct. 1987.
- [23] W. Kellerman, "A self-steering digital microphone array," *Proc. IEEE ICASSP*, vol. 5, pp. 3581-3584, 1991.
- [24] C. Marro, Y. Mahieux, K.U. Simmer, "Analysis of noise reduction and dereverberation techniques based on microphone arrays with postfiltering," *IEEE Trans. Speech Audio Process*, vol. 6, no. 5, pp. 240-259, May 1998.
- [25] I. Cohen, "Analysis of two-channel generalized sidelobe canceller (GSC) with post-filtering," *IEEE Trans. Speech Audio Process.*, vol. 11, pp. 684-699, Nov. 2003.
- [26] J. Benesty, S. Makino, J.E. Chen, *Speech Enhancement*, Springer, Germany, 2005.
- [27] Y. Huang, J. Benesty, J. Chen, *Acoustic MIMO Signal Processing*, Springer, Boston, 2006.
- [28] J. Chen, J. Benesty, Y. Huang, "An acoustic MIMO framework for analysing microphone-array beamforming," *Proc. IEEE ICASSP*, pp. I-25–I-28, 2007.
- [29] "Xilinx Virtex 5 family user guide," www.xilinx.com.
- [30] "Xilinx ML507 evaluation platform user guide," www.xilinx.com.
- [31] C. Anghel, C. Paleologu, J. Benesty, S. Ciochină, "FPGA Implementation of an Acoustic Echo Canceller Using a VSS-NLMS Algorithm", *Proc. IEEE International Symposium on Signals, Circuits and Systems (ISSCS)*, pp. 369-372, Iași, România, 2009.
- [32] C. Anghel, C. Paleologu, J. Benesty, S. Ciochină, "FPGA Implementation of a Variable Step-Size Affine Projection Algorithm for Acoustic Echo Cancellation", *Proc. European Signal Processing Conference (EUSIPCO)*, pp. 532-536, Aalborg, Danemarca, Aug. 2010.