

Noi algoritmi de tip VFF-RLS cu complexitate aritmetică redusă

SINTEZA FAZEI UNICE / 2013

Contract nr. 7/5.08.2010, Cod TE-50

În cadrul acestei etape de cercetare a proiectului, am dezvoltat o nouă familie de algoritmi adaptivi de tip VFF-RLS (*variable-forgetting factor recursive least-squares*) cu complexitate aritmetică redusă, bazați pe metodele iterative de tip DCD (*dichotomous coordinate descent*). Contextul în care au fost aplicați algoritmi VFF-DCD-RLS propuși a fost acela de compensare a ecoului acustic stereofonic pe baza modelului WL (*widely linear*) [1], [2] (Fig. 1).

În contextul acestei aplicații, algoritmi de tip RLS sunt preferați datorită vitezei lor de convergență. Cu toate acestea, complexitatea aritmetică a algoritmului RLS “clasic” (proporțională cu L^2 , unde L este lungimea filtrului) este prohibitivă pentru implementările practice, ținând cont de dimensiunea căilor de ecou și implicit a filtrelor adaptive utilizate (sute sau chiar mii de coeficienți).

Algoritmul FRLS (*fast RLS*) [3] are o complexitate aritmetică proporțională cu lungimea filtrului, ceea ce poate fi realizabil în practică. Din păcate, acest algoritm are probleme de instabilitate numerică în special în cazul prezenței unor semnale de intrare nestaționare (cum ar fi semnalul vocal), astfel încât trebuie repornit periodic pentru a preveni asemenea situații [4].

Pentru a depăși acest inconvenient, vom utiliza algoritmi de tip DCD-RLS [5], [6]. Metoda DCD [7] reprezintă o soluție eficientă de rezolvare iterativă a ecuațiilor normale din cadrul algoritmilor RLS, necesitând pentru implementare doar operații de adunare. Ca și în cazul algoritmului RLS, performanțele algoritmilor de tip DCD-RLS sunt influențate de valoarea factorului de ponderare (sau “uitare” – *forgetting factor*). Pentru a controla acest aspect, am propus algoritmi de tip VFF-DCD-RLS, ce utilizează un factor de uitare variabil.

Punctul de plecare îl reprezintă algoritmul WL-RLS [1], sintetizat în tabelul 1. Detalii despre acest algoritm și parametrii acestuia au fost furnizate în cadrul etapei de cercetare precedente [8].

Abordarea în contextul metodei DCD se bazează pe transformarea ecuațiilor normale într-o secvență de ecuații normale auxiliare, ce urmează a fi ulterior rezolvate prin tehnici iterative. În cazul modelului WL, vom presupune că sistemul de ecuații normale este rezolvat pentru momentul de timp $n - 1$, iar soluția aproximativă $\tilde{\mathbf{h}}(n - 1)$ este disponibilă. În acest context, putem defini un vector rezidual al soluției astfel:

$$\mathbf{r}(n - 1) = \mathbf{p}_{\bar{x}d}(n - 1) - \mathbf{R}_{\bar{x}}(n - 1)\tilde{\mathbf{h}}(n - 1). \quad (1)$$

Pe baza notației

$$\Delta\tilde{\mathbf{h}}(n) = \tilde{\mathbf{h}}(n) - \tilde{\mathbf{h}}(n - 1), \quad (2)$$

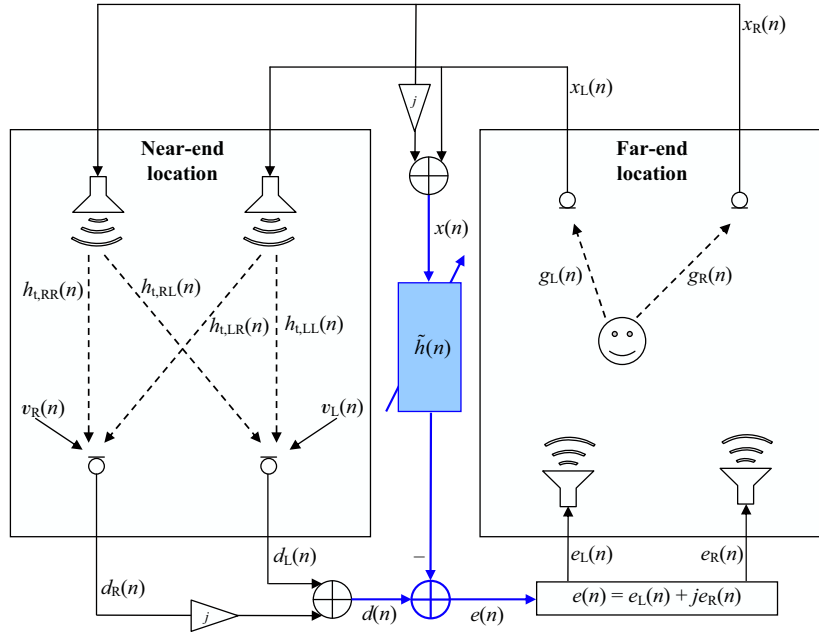


Figure 1: Modelul WL pentru compensarea ecoului acustic stereofonic.

Table 1: Algoritmul WL-RLS.

Inițializare:

$$\tilde{\mathbf{h}}(0) = \mathbf{0}$$

$$\mathbf{R}_{\tilde{\mathbf{x}}}^{-1}(0) = \delta^{-1} \mathbf{I}_{2L} \text{ (cu } \delta > 0, \text{ parametrul de regularizare)}$$

$$\lambda = 1 - \frac{1}{2CL}, \text{ factorul de uitare (cu } C \geq 3)$$

Pentru $n = 1, 2, \dots$:

$$\tilde{\mathbf{k}}(n) = \frac{\mathbf{R}_{\tilde{\mathbf{x}}}^{-1}(n-1)\tilde{\mathbf{x}}(n)}{\lambda + \tilde{\mathbf{x}}^H(n)\mathbf{R}_{\tilde{\mathbf{x}}}^{-1}(n-1)\tilde{\mathbf{x}}(n)}$$

$$e(n) = d(n) - \tilde{\mathbf{h}}^H(n-1)\tilde{\mathbf{x}}(n)$$

$$\tilde{\mathbf{h}}(n) = \tilde{\mathbf{h}}(n-1) + \tilde{\mathbf{k}}(n)e^*(n)$$

$$\mathbf{R}_{\tilde{\mathbf{x}}}^{-1}(n) = \lambda^{-1}\mathbf{R}_{\tilde{\mathbf{x}}}^{-1}(n-1) - \lambda^{-1}\tilde{\mathbf{k}}(n)\tilde{\mathbf{x}}^H(n)\mathbf{R}_{\tilde{\mathbf{x}}}^{-1}(n-1)$$

ecuațiile normale devin

$$\mathbf{R}_{\tilde{\mathbf{x}}}(n) \left[\tilde{\mathbf{h}}(n-1) + \Delta \tilde{\mathbf{h}}(n) \right] = \mathbf{p}_{\tilde{\mathbf{x}}d}(n). \quad (3)$$

În plus, utilizând notația

$$\Delta \mathbf{R}_{\tilde{\mathbf{x}}}(n) = \mathbf{R}_{\tilde{\mathbf{x}}}(n) - \mathbf{R}_{\tilde{\mathbf{x}}}(n-1), \quad (4)$$

$$\Delta \mathbf{p}_{\tilde{\mathbf{x}}d}(n) = \mathbf{p}_{\tilde{\mathbf{x}}d}(n) - \mathbf{p}_{\tilde{\mathbf{x}}d}(n-1), \quad (5)$$

Table 2: Algoritmul DCD ciclic pentru date complexe.

Inițializare:	
	$\Delta\tilde{\mathbf{h}}(n) = \mathbf{0}, \mathbf{r}(n) = \mathbf{p}(n)$
	$\eta = H, q = 0, s = 1$
Pentru $m = 1, 2, \dots, M_b$:	
Pasul 1:	$\eta = \eta/2$
Pasul 2:	$\text{flag} = 0$
Pentru $k = 1, 2, \dots, 2L$:	
Pasul 3:	Dacă $s = 1$, atunci $r_{\text{temp}} = \text{Re}[r_k(n)]$ altfel $r_{\text{temp}} = \text{Im}[r_k(n)]$
	Dacă $ r_{\text{temp}} > (\eta/2) R_{\tilde{\mathbf{x}},k}(n)$
Pasul 4:	$\Delta\tilde{h}_k(n) = \Delta\tilde{h}_k(n) + \text{sgn}(r_{\text{temp}})s\eta$
Pasul 5:	$\mathbf{r}(n) = \mathbf{r}(n) - \text{sgn}(r_{\text{temp}})s\eta\mathbf{R}_{\tilde{\mathbf{x}}}^{(k)}(n)$
Pasul 6:	$q = q + 1, \text{flag} = 1$
Pasul 7:	Dacă $q = N_u$, algoritmul se oprește
Pasul 8:	Dacă $s = 1$, atunci $s = j$, mergi in Pasul 3 altfel $s = 1$
Pasul 9:	Dacă $\text{flag} = 1$, atunci mergi in Pasul 2

și ținând cont de (1), sistemul de ecuații (3) poate fi rescris ca

$$\mathbf{R}_{\tilde{\mathbf{x}}}(n)\Delta\tilde{\mathbf{h}}(n) = \mathbf{r}(n-1) + \Delta\mathbf{p}_{\tilde{\mathbf{x}}d}(n) - \Delta\mathbf{R}_{\tilde{\mathbf{x}}}(n)\tilde{\mathbf{h}}(n-1). \quad (6)$$

Vom nota prin $\mathbf{p}(n)$ termenul din partea dreaptă a ecuației (6), adică

$$\mathbf{p}(n) = \mathbf{r}(n-1) + \Delta\mathbf{p}_{\tilde{\mathbf{x}}d}(n) - \Delta\mathbf{R}_{\tilde{\mathbf{x}}}(n)\tilde{\mathbf{h}}(n-1). \quad (7)$$

În consecință, ecuațiile normale auxiliare [5] sunt

$$\mathbf{R}_{\tilde{\mathbf{x}}}(n)\Delta\tilde{\mathbf{h}}(n) = \mathbf{p}(n). \quad (8)$$

De asemenea, ținând cont de formula de reactualizare a matricei de autocorelație, împreună cu notația (4), rezultă că

$$\Delta\mathbf{R}_{\tilde{\mathbf{x}}}(n) = (\lambda - 1)\mathbf{R}_{\tilde{\mathbf{x}}}(n-1) + \tilde{\mathbf{x}}(n)\tilde{\mathbf{x}}^H(n). \quad (9)$$

Multiplicând (9) la dreapta cu $\tilde{\mathbf{h}}(n-1)$ și ținând cont de (1) și (7), obținem

$$\mathbf{p}(n) = \lambda\mathbf{r}(n-1) + \tilde{\mathbf{x}}(n)e^*(n). \quad (10)$$

Relația (10) este utilizată pentru a evalua termenul din partea dreaptă a ecuației (8).

Se poate demonstra că rezolvarea ecuațiilor normale auxiliare (8) este echivalentă cu minimizarea funcției

$$J_s \left[\Delta\tilde{\mathbf{h}}(n) \right] = \Delta\tilde{\mathbf{h}}^H(n)\mathbf{R}_{\tilde{\mathbf{x}}}(n)\Delta\tilde{\mathbf{h}}(n) - \Delta\tilde{\mathbf{h}}^H(n)\mathbf{p}(n). \quad (11)$$

Table 3: Algoritmul WL-DCD-RLS.

Inițializare:

$$\tilde{\mathbf{h}}(0) = \mathbf{0}, \mathbf{r}(0) = \mathbf{0}$$

$$\mathbf{R}_{\tilde{\mathbf{x}}}(0) = \delta \mathbf{I}_{2L} \text{ (cu } \delta > 0, \text{ parametrul de regularizare)}$$

$$\lambda = 1 - \frac{1}{2CL}, \text{ factorul de uitare (cu } C \geq 3)$$

Pentru $n = 1, 2, \dots$:

$$\mathbf{R}_{\tilde{\mathbf{x}}}(n) = \lambda \mathbf{R}_{\tilde{\mathbf{x}}}(n-1) + \tilde{\mathbf{x}}(n) \tilde{\mathbf{x}}^H(n)$$

$$e(n) = d(n) - \tilde{\mathbf{h}}^H(n-1) \tilde{\mathbf{x}}(n)$$

$$\mathbf{p}(n) = \lambda \mathbf{r}(n-1) + \tilde{\mathbf{x}}(n) e^*(n)$$

$$\mathbf{R}_{\tilde{\mathbf{x}}}(n) \Delta \tilde{\mathbf{h}}(n) = \mathbf{p}(n) \Rightarrow \Delta \tilde{\mathbf{h}}(n), \mathbf{r}(n)$$

(de rezolvat cu iterațiile DCD – Tabelul 2)

$$\tilde{\mathbf{h}}(n) = \tilde{\mathbf{h}}(n-1) + \Delta \tilde{\mathbf{h}}(n)$$

Procedura poate fi rezumată astfel: în primul rând, se setează numărul maxim de iterații N_u (permis pentru rezolvarea sistemului de ecuații); în continuare, la fiecare iterație q (cu $q < N_u$), soluția $\Delta \tilde{\mathbf{h}}(n)$ este reactualizată în direcția $\mathbf{d}_{\mathbf{r}}(n)$, astfel încât

$$\mathbf{d}_{\mathbf{r}}^H(n) \mathbf{r}(n) \neq \mathbf{0}, \quad (12)$$

În acest caz, funcția ce trebuie minimizată este $J_s \left[\Delta \tilde{\mathbf{h}}(n) + \eta \mathbf{d}_{\mathbf{r}}(n) \right]$, unde pasul η poate fi ales în diverse moduri. Algoritmul DCD utilizează un pas variabil, ales ca putere a lui 2. De fapt, pasul respectiv ia una din cele M_b valori predefinite, corespunzătoare reprezentării binare a lui $\Delta \tilde{\mathbf{h}}(n)$ cu M_b biți în intervalul de amplitudini $[-H, H]$.

În contextul modelului WL, ecuațiile auxiliare normale sunt rezolvate pe baza algoritmului DCD ciclic pentru date complexe, prezentat în tabelul 2, iar algoritmul WL-DCD-RLS rezultat este descris în tabelul 3. Complexitatea de calcul depinde de numărul de iterații “reușite”; astfel, în cazul cel mai defavorabil, vom avea $2(2L)(2N_u + M_b - 1) + N_u$ operații.

În plus, complexitatea aritmetică poate fi redusă în primul pas al algoritmului, ținând cont de modul de reactualizare a vectorului $\tilde{\mathbf{x}}(n)$ și de faptul că matricea $\mathbf{R}_{\tilde{\mathbf{x}}}(n)$ este hermitică. În consecință, numai primele două coloane ale matricei trebuie evaluate:

$$\mathbf{R}_{\tilde{\mathbf{x}}}^{(1)}(n) = \lambda \mathbf{R}_{\tilde{\mathbf{x}}}^{(1)}(n-1) + x^*(n) \tilde{\mathbf{x}}(n), \quad (13)$$

$$\mathbf{R}_{\tilde{\mathbf{x}}}^{(2)}(n) = \lambda \mathbf{R}_{\tilde{\mathbf{x}}}^{(2)}(n-1) + x(n) \tilde{\mathbf{x}}(n). \quad (14)$$

Prin urmare, algoritmul WL-DCD-RLS necesită $7(2L)$ operații de înmulțire și $2L(4N_u + 2M_b + 3) + N_u$ adunări (în cazul cel mai defavorabil). În concluzie, complexitatea aritmetică a algoritmului WL-DCD-RLS este proporțională cu $2LN_u$ (unde $N_u \ll L$), ceea ce reprezintă o soluție atractivă în practică.

Ca și în cazul algoritmului RLS clasic, algoritmul WL-DCD-RLS prezentat anterior utilizează o valoare constantă a factorului de uitare λ , ceea ce conduce la un compromis între criteriile de performanță ale filtrului adaptiv (viteza de convergență/capacitatea de urmărire versus dezadaptare/robustete). Pentru a dezvolta o variantă VFF a acestui algoritm, se pleacă de la premisa că semnalul de referință $d(n)$ conține ecoul complex $y(n)$ și semnalul de la capătul

Table 4: Algoritmul WL-VFF-RLS.

Inițializare:

$$\begin{aligned}
 \tilde{\mathbf{h}}(0) &= \mathbf{0} \\
 \mathbf{R}_{\tilde{\mathbf{x}}}^{-1}(0) &= \delta^{-1} \mathbf{I}_{2L} \text{ (with } \delta > 0, \text{ parametrul de regularizare)} \\
 \lambda(0) &= \lambda_{\max} \text{ (foarte apropiat de 1)} \\
 \hat{\sigma}_e^2(0) &= 0 \\
 \hat{\sigma}_u^2(0) &= 0 \\
 \alpha &= 1 - \frac{1}{2KL}, \text{ with } K > 1 \\
 1 &< \gamma \leq 2 \\
 \varepsilon &= \text{o constantă pozitivă mică}
 \end{aligned}$$

Pentru $n = 1, 2, \dots$:

$$\begin{aligned}
 u(n) &= \tilde{\mathbf{x}}^H(n) \mathbf{R}_{\tilde{\mathbf{x}}}^{-1}(n-1) \tilde{\mathbf{x}}(n) \\
 \tilde{\mathbf{k}}(n) &= \frac{\mathbf{R}_{\tilde{\mathbf{x}}}^{-1}(n-1) \tilde{\mathbf{x}}(n)}{\lambda(n-1) + u(n)} \\
 e(n) &= d(n) - \tilde{\mathbf{h}}^H(n-1) \tilde{\mathbf{x}}(n) \\
 \tilde{\mathbf{h}}(n) &= \tilde{\mathbf{h}}(n-1) + \tilde{\mathbf{k}}(n) e^*(n) \\
 \mathbf{R}_{\tilde{\mathbf{x}}}^{-1}(n) &= \lambda^{-1}(n-1) \mathbf{R}_{\tilde{\mathbf{x}}}^{-1}(n-1) - \lambda^{-1}(n-1) \tilde{\mathbf{k}}(n) \tilde{\mathbf{x}}^H(n) \mathbf{R}_{\tilde{\mathbf{x}}}^{-1}(n-1) \\
 \hat{\sigma}_u^2(n) &= \alpha \hat{\sigma}_u^2(n-1) + (1-\alpha) u^2(n) \\
 \hat{\sigma}_e^2(n) &= \alpha \hat{\sigma}_e^2(n-1) + (1-\alpha) |e(n)|^2 \\
 \text{Calculează } &\hat{\sigma}_v^2(n) \\
 \text{Dacă } \hat{\sigma}_e(n) &> \gamma \hat{\sigma}_v(n), \text{ atunci} \\
 \lambda(n) &= \min \left[\frac{\hat{\sigma}_u(n) \hat{\sigma}_v(n)}{\varepsilon + |\hat{\sigma}_e(n) - \hat{\sigma}_v(n)|}, \lambda_{\max} \right] \\
 \text{altfel } \lambda(n) &= \lambda_{\max}
 \end{aligned}$$

apropiat (*near-end*), $v(n)$. În acest context, scopul filtrului adaptiv nu este de a anula eroarea, ci de a recupera semnalul de la capătul apropiat în eroarea filtrului adaptiv. Pe baza acestei abordări, a fost propus în [9] un algoritm de tip WL-VFF-RLS, sintetizat în tabelul 4. Detalii despre acest algoritm au fost prezentate în cadrul fazei precedente [8].

În continuare, vom extinde această abordare și în cazul algoritmului WL-DCD-RLS. Pentru început, relația dintre eroarea a posteriori și cea a priori poate fi dezvoltată ca

$$\begin{aligned}
 \epsilon(n) &= d(n) - \tilde{\mathbf{h}}^H(n) \tilde{\mathbf{x}}(n) \\
 &= d(n) - \left[\tilde{\mathbf{h}}^H(n-1) + \Delta \tilde{\mathbf{h}}^H(n) \right] \tilde{\mathbf{x}}(n) \\
 &= e(n) - \Delta \tilde{\mathbf{h}}^H(n) \tilde{\mathbf{x}}(n).
 \end{aligned} \tag{15}$$

De asemenea, comparând relațiile de reactualizare pentru $\tilde{\mathbf{h}}(n)$ din tabelul 1 și tabelul 3, observăm că avem

$$\Delta \tilde{\mathbf{h}}(n) = \tilde{\mathbf{k}}(n) e^*(n). \tag{16}$$

Astfel, vom obține aceeași expresie a factorului de uitare variabil ca și în cazul algoritmului WL-VFF-RLS. Principala

Table 5: Algoritmul WL-VFF-DCD-RLS.

Inițializare:

$$\tilde{\mathbf{h}}(0) = \mathbf{0}, \mathbf{r}(0) = \mathbf{0}$$

$$\mathbf{R}_{\tilde{\mathbf{x}}}(0) = \delta \mathbf{I}_{2L} \text{ (cu } \delta > 0, \text{ parametrul de regularizare)}$$

$$\lambda(0) = \lambda_{\max} \text{ (foarte apropiat de 1)}$$

$$\hat{\sigma}_e^2(0) = 0$$

$$\hat{\sigma}_u^2(0) = 0$$

$$\alpha = 1 - \frac{1}{2KL}, \text{ cu } K > 1$$

$$1 < \gamma \leq 2$$

$$\varepsilon = \text{o constantă pozitivă mică}$$

Pentru $n = 1, 2, \dots$:

$$\mathbf{R}_{\tilde{\mathbf{x}}}(n) = \lambda(n-1)\mathbf{R}_{\tilde{\mathbf{x}}}(n-1) + \tilde{\mathbf{x}}(n)\tilde{\mathbf{x}}^H(n)$$

$$e(n) = d(n) - \tilde{\mathbf{h}}^H(n-1)\tilde{\mathbf{x}}(n)$$

$$\mathbf{p}(n) = \lambda(n-1)\mathbf{r}(n-1) + \tilde{\mathbf{x}}(n)e^*(n)$$

$$\mathbf{R}_{\tilde{\mathbf{x}}}(n)\Delta\tilde{\mathbf{h}}(n) = \mathbf{p}(n) \Rightarrow \Delta\tilde{\mathbf{h}}(n), \mathbf{r}(n)$$

(de rezolvat cu iterațiile DCD – Tabelul 2)

$$\tilde{\mathbf{h}}(n) = \tilde{\mathbf{h}}(n-1) + \Delta\tilde{\mathbf{h}}(n)$$

$$u(n) = \lambda(n-1) \frac{\tilde{\mathbf{x}}^H(n)\Delta\tilde{\mathbf{h}}(n)}{e^*(n) - \tilde{\mathbf{x}}^H(n)\Delta\tilde{\mathbf{h}}(n)}$$

$$\hat{\sigma}_u^2(n) = \alpha\hat{\sigma}_u^2(n-1) + (1-\alpha)u^2(n)$$

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

Calculează $\hat{\sigma}_v^2(n)$

Dacă $\hat{\sigma}_e(n) > \gamma\hat{\sigma}_v(n)$, atunci

$$\lambda(n) = \min \left[\frac{\hat{\sigma}_u(n)\hat{\sigma}_v(n)}{\varepsilon + |\hat{\sigma}_e(n) - \hat{\sigma}_v(n)|}, \lambda_{\max} \right]$$

altfel $\lambda(n) = \lambda_{\max}$

problemă rămâne însă modul de evaluare a parametrului $u(n)$, pentru a estima ulterior $\hat{\sigma}_u(n)$.

În cazul algoritmului WL-DCD-RLS, vectorul soluție $\Delta\tilde{\mathbf{h}}(n)$ rezultă din iterațiile DCD (a se vedea tabelul 2). Pe de altă parte, în cazul algoritmului WL-VFF-RLS vom avea

$$\tilde{\mathbf{x}}^H(n)\tilde{\mathbf{k}}(n) = \frac{u(n)}{\lambda(n-1) + u(n)}. \quad (17)$$

În consecință, rezultă că

$$u(n) = \lambda(n-1) \frac{\tilde{\mathbf{x}}^H(n)\Delta\tilde{\mathbf{h}}(n)}{e^*(n) - \tilde{\mathbf{x}}^H(n)\Delta\tilde{\mathbf{h}}(n)}. \quad (18)$$

În final, algoritmul WL-VFF-DCD-RLS propus este sintetizat în tabelul 5. Legături între algoritmii de tip RLS și cei bazați pe filtrare Kalman au fost recent prezentate în [10] și [11]

References

- [1] J. Benesty, C. Paleologu, T. Gänslér, and S. Ciochină, *A Perspective on Stereophonic Acoustic Echo Cancellation*. Berlin, Germany: Springer-Verlag, 2011.
- [2] C. Stanciu, J. Benesty, C. Paleologu, T. Gänslér, and S. Ciochină, "A widely linear model for stereophonic acoustic echo cancellation," *Signal Processing*, vol. 93, pp. 511–516, Feb. 2013.
- [3] M. G. Bellanger, *Adaptive Filters and Signal Analysis*. NY: Dekker, 1988.
- [4] P. Eneroth, S. L. Gay, T. Gänslér, and J. Benesty, "A real-time implementation of a stereophonic acoustic echo canceler," *IEEE Trans. Audio, Speech Process.*, vol. 9, pp. 513–523, July 2001.
- [5] Y. V. Zakharov, G. P. White, and J. Liu, "Low-complexity RLS algorithms using dichotomous coordinate descent iterations," *IEEE Trans. Signal Process.*, vol. 56, pp. 3150–3161, July 2008.
- [6] J. Liu, Y. V. Zakharov, and B. Weaver, "Architecture and FPGA design of dichotomous coordinate descent algorithms," *IEEE Trans. Circuits and Systems I: Regular Papers*, vol. 56, pp. 2425–2438, Nov. 2009.
- [7] Y. V. Zakharov and T. C. Tozer, "Multiplication-free iterative algorithm for LS problem," *IEE Electronics Lett.*, vol. 40, pp. 567–569, Apr. 2004.
- [8] <http://www.comm.pub.ro/te-vss/>
- [9] C. Stanciu, C. Paleologu, J. Benesty, S. Ciochină, and F. Albu, "A variable-forgetting factor RLS algorithm for stereophonic acoustic echo cancellation with the widely linear model," in *Proc. EUSIPCO*, 2012, pp. 1960–1964.
- [10] C. Paleologu, J. Benesty, and S. Ciochină, "Study of the general Kalman filter for echo cancellation," *IEEE Trans. Audio, Speech, Language Processing*, vol. 21, pp. 1539–1549, Aug. 2013.
- [11] C. Paleologu, J. Benesty, and S. Ciochină, "Widely linear general Kalman filter for stereophonic acoustic echo cancellation," *Signal Processing*, vol. 94, pp. 570–575, Jan. 2014.

Director de proiect

Prof. dr. ing. Constantin Paleologu