Nouvel algorithme de type NLMS à convergence rapide pour le filtrage adaptatif

AHMED BENALLAL[1], MAJID AREZKI[2]

[1,2] Laboratoire Traitement du Signal et Image LATSI
Département d'Electronique, Faculté de Technologie, Université de BLIDA
Route de Soumaa Blida, Algérie.

[1]a_benallal@hotmail.com, [2]md_arezki@hotmail.com

Résumé - Dans cette communication, nous proposons un algorithme à complexité réduite obtenu à partir d'un algorithme des moindres carrés rapide (MCR) où le gain d'adaptation est calculé en annulant complètement les prédicteurs aller et retour de l'algorithme. L'erreur de prédiction aller, utilisée dans le calcul du gain d'adaptation, est calculée à partir du signal d'entrée indépendamment du gain d'adaptation. L'algorithme obtenu présente une complexité similaire à celle de l'algorithme NLMS (2N multiplications par itération) et montre de meilleures performances en vitesse de convergence et en capacité de poursuite que l'algorithme classique NLMS.

Abstract - A new adaptive algorithm with fast convergence and low complexity is proposed. It is the result of a simplification of the fast transversal filter algorithm . The adaptation gain of the proposed algorithm is obtained by discarding the backward and forward predictors from the fast transversal filter algorithm . By using only the calculation structure of the dual Kalman variables and a simple decorrelating technique for the input signal, we obtain an algorithm that exhibits faster convergence speed and enhanced tracking ability compared to the NLMS algorithm, whereas its computational complexity is similar to that of the NLMS algorithm.

1 Introduction

On considère dans cette communication le problème classique du filtrage adaptatif où l'actualisation du filtre transverse $\boldsymbol{h_N}(n)$, d'ordre N, est réalisée par:

$$\mathbf{h}_N(n) = \mathbf{h}_N(n-1) - \mathbf{c}_N(n)\varepsilon_N(n) \quad (1)$$

où $\varepsilon_N(n)$ est l'erreur de filtrage :

$$\varepsilon_N(n) = d(n) - \mathbf{h}_N^{\mathrm{T}}(n-1)\mathbf{x}_N(n) \qquad (2)$$

où le vecteur $\mathbf{x}_N(n)$ contient les N derniers échantillons du signal d'entrée. Le signal *d(n)* est le signal désiré (appelé aussi écho). $\boldsymbol{c_N}(n)$ est le vecteur gain d'adaptation. Différents algorithmes adaptatifs se distinguent par la façon de calculer le gain $\boldsymbol{c_N}(n)$. L'algorithme LMS normalisé (NLMS) est l'algorithme le plus utilisé en pratique à cause de sa simplicité et de sa complexité réduite. L'inconvénient majeur de cet algorithme est sa vitesse de convergence lente surtout en présence d'un signal d'excitation fortement corrélé [1].
Le gain d'adaptation de l'algorithme NLMS est :

$$\mathbf{c}_N(n) = -\mu \frac{\mathbf{x}(n)}{p_x(n) + c_0} \qquad (3)$$

où μ est le pas d'adaptation et c_0 une constante positive qui évite des divisions par de très faibles valeurs . Le terme $p_x(n) = \mathbf{x}_N^T(n)\mathbf{x}_N(n)$ se calcule récursivement, en pratique, en utilisant une fenêtre à facteur d'oubli exponentiel β:

$$p_x(n) = \beta\, p_x(n-1) + (1-\beta) N\, x^2(n) \qquad (4)$$

La condition de stabilité du NLMS est $0 < \mu < 2$ et sa vitesse de convergence est maximale pour $\mu = 1$ [8].

L'algorithme des moindres carrés récursif (RLS) et sa famille sont caractérisés par une vitesse de convergence indépendante des propriétés statistiques du signal d'entrée. En exploitant, certaines propriétés d'invariance par décalage du vecteur signal d'entrée, les versions rapides des moindres carrées (MCR) calculent la solution avec une complexité entre *7N* et *10N* par itération. Dans les versions les plus rapides, le gain d'adaptation propagé est défini par :

$$\boldsymbol{c_N}(n) = \gamma_N(n)\tilde{\boldsymbol{c}}_{\boldsymbol{N}}(n) \quad (4)$$

où les variables $\tilde{\mathbf{c}}_N(n)$ and $\gamma_N(n)$ désignent respectivement le gain de Kalman dual et la variable de vraisemblance. Ces dernières variables sont calculés grâce une analyse par prédiction linéaire aller et retour sur le signal d'entrée [2,3].

Une complexité de calculs encore plus réduite peut être obtenue en utilisant l'algorithme Fast Netwon transversal filters (FNTF) [4,5], ou the simplified Fast Transversal filter (SMFTF) algorithm [6,7]. L'idée principale derrière ces algorithmes est la modélisation du signal d'entrée par un modèle AR d'ordre *P* faible devant l'ordre du filtre *N*.

Dans cette communication, nous proposons un algorithme à complexité réduite obtenu à partir d'un algorithme MCR où le gain d'adaptation est calculé en annulant complètement des prédicteurs aller et retour de l'algorithme MCR. L'erreur de prédiction aller, utilisée dans le calcul du gain d'adaptation est obtenue à partir du signal d'entrée indépendamment du gain d'adaptation. L'algorithme obtenu présente une complexité similaire à celle de l'algorithme NLMS (2N multiplications par itération).

2 Obtention de l'algorithme

La relation d'adaptation du gain de Kalman dual peut s'écrire en faisant intervenir un prédicteur aller $\mathbf{a}_N(n)$ et prédicteur retour $\mathbf{b}_N(n)$ [2 ,3]:

$$\begin{bmatrix} \tilde{\mathbf{c}}_N(n) \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \tilde{\mathbf{c}}_N(n-1) \end{bmatrix} - \frac{\bar{e}_N(n)}{\lambda\alpha_N(n-1)} \begin{bmatrix} 1 \\ -\mathbf{a}_N(n-1) \end{bmatrix} + \frac{\bar{r}_N(n)}{\lambda\beta_N(n-1)} \begin{bmatrix} -\mathbf{b}_N(n-1) \\ 1 \end{bmatrix} \quad (5)$$

où $\bar{e}_N(n)$ et $\alpha_N(n)$ désignent respectivement l'erreur de prédiction aller et sa variance. Les variables $\bar{r}_N(n)$ et $\beta_N(n)$ sont l'erreur de prédiction retour et sa variance, respectivement. La constante λ est un facteur d'oubli exponentiel.

Dans [6], on montre que lorsque $\mathbf{a}_N(n) = \mathbf{b}_N(n) = 0$, le dual Kalman gain (5) s'écrit:

$$\begin{bmatrix} \tilde{\mathbf{c}}_N(n) \\ c(n) \end{bmatrix} = \begin{bmatrix} -\dfrac{x(n)}{\lambda.\alpha_N(n-1)} \\ \tilde{\mathbf{c}}_N(n-1) \end{bmatrix} \quad (6)$$

où la variable $c(n)$ est la quantité $-\dfrac{\bar{r}_N(n)}{\lambda\beta_N(n-1)}$ et $\bar{e}_N(n) = x(n)$. Après les N premières itérations, en régime asymptotique, le gain d'adaptation peut approximativement s'écrire :

$$\mathbf{c}_N(n) = \gamma_N(n)\tilde{\mathbf{c}}_N(n) \approx -\frac{\mathbf{x}_N(n)}{M\sigma_x^2} \quad (7)$$

où M est une constante positive. L'approximation (7) est une forme très similaire au gain d'adaptation de l'algorithme NLMS. Cette observation montre que l'algorithme MCR, sans prédicteurs aller/retour, doit fonctionner, mais peut-être avec un comportement proche de celui du NLMS.

Dans l'algorithme proposé, on élimine complètement les prédicteurs $a_N(n) = b_N(n) = 0$ et on utilise seulement une erreur de prédiction aller, notée $e(n)$ au lieu $\bar{e}_N(n)$, pour évaluer le gain de Kalman dual :

$$\begin{bmatrix} \tilde{\mathbf{c}}_N(n) \\ c(n) \end{bmatrix} = \begin{bmatrix} -\dfrac{e(n)}{\lambda\alpha(n-1) + c_0} \\ \tilde{\mathbf{c}}_N(n-1) \end{bmatrix} \quad (8)$$

Comme pour le NLMS, nous avons introduit une constante c_0 qui évite des divisions par de très faibles valeurs de la variance $\alpha(n)$. L'utilisation de l'erreur de prédiction $e(n)$ au lieu $x(n)$ dans (8) est importante pour la vitesse de convergence car elle introduit un blanchiment du signal $x(n)$ dans la génération du gain d'adaptation [8,10].

La variance $\alpha(n)$ est estimée récursivement par:

$$\alpha(n) = \lambda\alpha(n-1) + e^2(n) \quad (9)$$

L'erreur de prédiction $e(n)$ peut être calculée par n'importe quel autre algorithme adaptatif. Pour aboutir à une complexité de l'ordre de 2N, on propose ici de générer $e(n)$ à l'aide d'un prédicteur d'ordre 1:

$$e(n) = x(n) - ax(n-1) \quad (9)$$

où le coefficient de prédiction a est calculé en minimisant l'EQM $E(e^2(n))$:

$$a = \frac{E[x(n)x(n-1)]}{E[x^2(n-1)]} = \frac{r_1}{r_0} \quad (10)$$

où r_0 and r_1 désignent, respectivement, la puissance du signal d'entré et le premier coefficient de la fonction d'auto corrélation de x(n). La relation (10) est évaluée par des estimateurs récursifs comme suit:

$$a(n) = \frac{r_1(n)}{r_0(n) + c_a} \quad (11)$$

avec

$$r_1(n) = \lambda_a r_1(n-1) + x(n)x(n-1) \quad (12a)$$

$$r_0(n) = \lambda_a r_0(n-1) + x^2(n) \quad (12b)$$

où λ_a est un facteur d'oubli et c_a une constante de régularisation .

Pour calculer la variable de vraisemblance $\gamma_N(t)$, plusieurs méthodes peuvent être utilisées. Une première méthode utilise la définition directe [3] :

$$\gamma_N(n) = \frac{1}{1 - \tilde{\mathbf{c}}_N^T(n)\mathbf{x}_N(n)} \quad (13)$$

Une deuxième méthode exploite la propriété de décalage après la première composante $\tilde{c}_N^1(n)$ du gain du gain de Kalman dual (8) :

$$\gamma_N(n) = \frac{1}{1 - \sum_{i=1}^{N} v(n-i+1)} \quad (14)$$

où $v(n) = \tilde{c}_N^1(n)x(n)$ est le signal utilisé dans le processus de décalage.

Dans une troisième méthode, en exploitant la propriété d'invariance par décalage du vecteur signal d'entrée d'ordre $N+1$, nous proposons une version récursive de la variable de vraisemblance. Le vecteur (8) multiplié à gauche par $[\mathbf{x}_N^{\mathrm{T}}(n) \quad x(n-N)]$ et à droite $[x(n) \quad \mathbf{x}_N^{\mathrm{T}}(n-1)]$, fait apparaitre une forme récursive de la variable de vraisemblance :

$$\gamma_N(n) = \frac{\gamma_N(n-1)}{1 + \gamma_N(n-1)\delta(n)} \quad (15)$$

où

$$\delta(n) = c(n)x(n-N) + \frac{x(n)e(n)}{\lambda.\alpha_N(n-1) + c_0} \quad (16)$$

En utilisant (14) ou (15), la complexité des calculs de l'algorithme proposé est de $2N$ multiplications par itération. L'algorithme proposé, appelé algorithme type

NLMS à convergence rapide (noté FNLMS) est résumé comme suit :

Initialisation: $\mathbf{c}_N(0)=\tilde{\mathbf{c}}_N(0)=\mathbf{h}_N(0)=\mathbf{0}$, $\gamma_N(0)=1$, $\alpha(0)=r_0(0)=E_0$, $r_1(0)=0$

Génération de : $e(n)$

$$r_1(n)=\lambda_a r_1(n-1)+x(n)x(n-1)$$

$$r_0(n)=\lambda_a r_0(n-1)+x^2(n)$$

$$a(n)=\frac{r_1(n)}{r_0(n)+c_a}$$

$$e(n)=x(n)-a(n)x(n-1)$$

$$\alpha(n)=\lambda\alpha(n-1)+e^2(n)$$

Gain d'adaptation :

$$\begin{bmatrix}\tilde{\mathbf{c}}_N(n)\\ c(n)\end{bmatrix}=\begin{bmatrix}-\dfrac{e(n)}{\lambda.\alpha_N(n-1)+c_0}\\ \tilde{\mathbf{c}}_N(n-1)\end{bmatrix}$$

Version 1:

$$\delta(n)=c(n)x(n-N)+\frac{x(n)e(n)}{\lambda.\alpha_N(n-1)+c_0}$$

$$\gamma_N(n)=\frac{\gamma_N(n-1)}{1+\gamma_N(n-1)\delta(n)}$$

Version 2: $v(n)=\tilde{c}_N^1(n)x(n)$;

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

Filtrage:

$$\varepsilon_N(n)=d(n)-\mathbf{h}_N^{\mathbf{T}}(n-1)\mathbf{x}_N(n)$$

$$\mathbf{h}_N(n)=\mathbf{h}_N(n-1)-\mu\,\varepsilon_N(n)\gamma_N(n)\tilde{\mathbf{c}}_N(n)$$

Notons que nous avons introduit une constante μ supplémentaire pour mieux contrôler la stabilité de l'algorithme.

L'analyse théorique de la convergence des algorithmes NLMS et MCR est un problème assez difficile. Pour trouver la condition de stabilité du nouvel algorithme, nous allons utiliser la condition de stabilité sur le pas d'adaptation μ de l'algorithme NLMS obtenue dans [8]. Pour cela, nous supposons que le signal d'entrée est un signal blanc gaussien et que toutes les variables de l'algorithme ont atteints leurs valeurs asymptotiques:

$$\alpha(n)\approx\frac{\sigma_x^2}{1-\lambda}\qquad(17a)$$

$$\tilde{\mathbf{c}}_N(n)\approx-\frac{\mathbf{x}(n)}{\frac{\lambda}{1-\lambda}\sigma_x^2+c_0}\qquad(17b)$$

$$\gamma_N(n)=\frac{1}{1-\tilde{\mathbf{c}}_N^T(n)\mathbf{x}_N(n)}\approx\frac{1}{1+\dfrac{N\sigma_x^2}{\lambda\sigma_x^2/(1-\lambda)+c_0}}\qquad(17c)$$

Le gain d'adaptation de l'algorithme NLMS, avec la condition de stabilité $0<\mu<2$, est approximativement donnée par:

$$\mathbf{c}_{NLMS}(n)\approx-\frac{\mu}{N\sigma_x^2}\mathbf{x}(n)\qquad(18)$$

En utilisant les approximations (17), le gain d'adaptation de l'algorithme proposé est :

$$\mathbf{c}_{FNLMS}(n)\approx-\frac{\mu}{N\sigma_x^2\left(1+\dfrac{\lambda}{(1-\lambda)N}+\dfrac{c_0}{N\sigma_x^2}\right)}\mathbf{x}(n)\approx\frac{\mathbf{c}_{NLMS}(n)}{1+\dfrac{\lambda}{(1-\lambda)N}+\dfrac{c_0}{N\sigma_x^2}}\qquad(19)$$

La comparaison du gain (19) avec celui du NLMS (18), montre que la condition de convergence pour nouvel algorithme est donnée par:

$$0<\frac{\mu}{1+\dfrac{\lambda}{(1-\lambda)N}+\dfrac{c_0}{N\sigma_x^2}}<2\qquad(20)$$

Cette inégalité est toujours vraie pour $0<\mu<2$. D'après (20), le pas d'adaptation implicite du nouvel algorithme doit être choisi selon la relation suivante:

$$\mu_{FNLMS}=\frac{\mu}{1+\dfrac{\lambda}{(1-\lambda)N}+\dfrac{c_0}{N\sigma_x^2}}\qquad(21)$$

Figure 1 représente le pas d'adaptation (21) en fonction du facteur d'oubli λ et pour différentes valeurs de *N*. Ces courbes montrent que le pas μ_{FNLMS} décroit rapidement pour λ proche de 1 et par conséquent, le facteur λ doit être convenablement choisi, surtout pour les tailles *N* faibles, pour éviter des dégradations liées à un pas d'adaptation très faible.

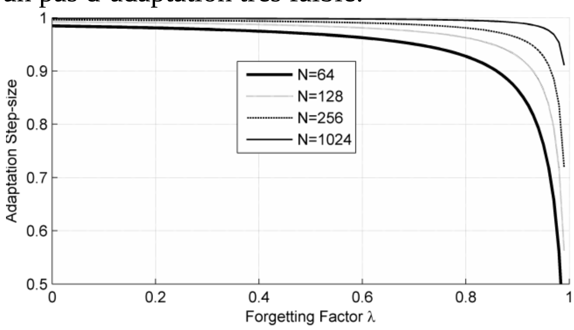

Fig. 1. Courbes du pas d'adaptation. $\sigma_x^2=c_0=1$ and $\mu=1$

3 Simulations

Dans toutes les simulations réalisées, l'algorithme proposé montre des performances en vitesse de convergence et de poursuite de non stationnarités meilleurs que celles de l'algorithme NMLS. La figure 2 montre les résultats obtenus pour le NLMS , le SFTF (l'algorithme MCR numériquement stable) [6] et l'algorithme proposé (FNLMS) pour l'identification d'une réponse implusionnelle acoustique de taille

N=256 . Le Signal d'entré est une modèle AR d'ordre 20. L'algorithme FNLMS – no prédiction correspond au cas où l'erreur de prédiction *e(n)=x(n)* , Ces résultats montrent que même le cas extrême d'un algorithme sans prédiction, l'algorithme proposé FNLMS est meilleur que l'algorithme NLMS ,

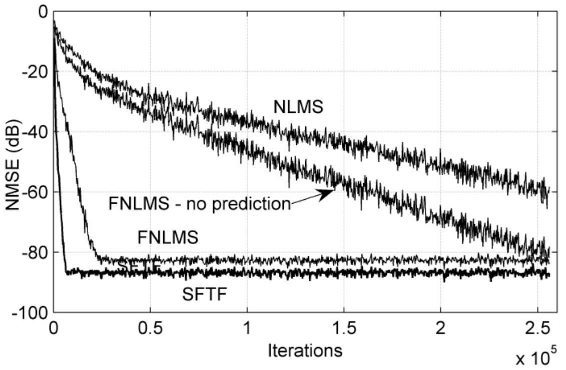

Fig.2 Résultats pour le NLMS, SFTF et FSNLMS pour un signal AR(20) and *N=256*. FNLMS: $\lambda = 0.99$. FNLMS – no prediction: $\lambda = 0.8$ SFTF: $\lambda = 0.9987$ NLMS : $\mu_{NLMS} = 1$

Figure 4 présente les résultants obtenues dans le cas de la poursuite d'une réponse impulsionnelle variable. La variation artificielle est réalisée par la multiplication du signal écho (bruit USASI filtré) par un gain variable (figure 3) . Ces résultats montrent la supériorité de l'algorithme proposé en capacité de poursuite sur les deux algorithmes SFTF et NLMS.

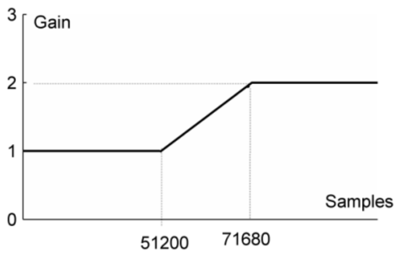

Fig. 3. Gain variation for the tracking experiment

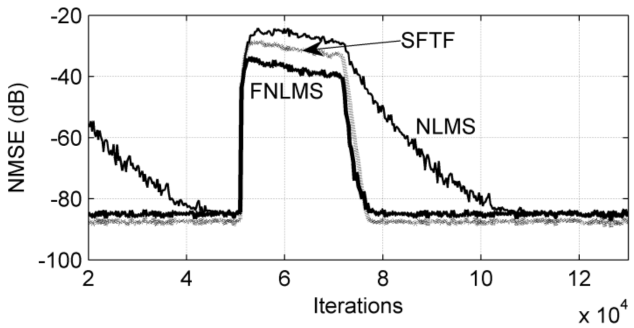

Fig. 4. Poursuite d'un filtre variable . *N=256*. FNLMS: $\lambda = 0.9925$, $\lambda_a = 0.9985$ SFTF: $\lambda = 0.9987$. NLMS : $\mu_{NLMS} = 1$

Des simulations avec un signal de parole en entrée montrent aussi la supériorité de cet algorithme par rapport à l'algorithme NLMS (figure 4). Le signal *d(n)* est l'énergie en dB du signal écho.

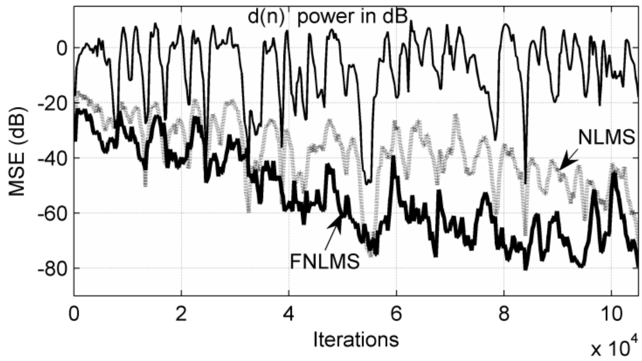

Fig. 5. Comparaison des performances du NLMS et du FNLMS avec un signal de parole et un filtre de taille *N=1500*. FNLMS: $\lambda = 09985$. NLMS : $\mu_{NLMS} = 1$

4 Conclusion

Nous avons présenté un nouvel algorithme pour le filtrage adaptatif, obtenu par simplification d'un algorithme MCR, dont la complexité des calculs est la même que celle de l'algorithme NLMS. Cet algorithme montre de meilleures performances en vitesse de convergence et en capacité de poursuite que l'algorithme classique NLMS.

Bibliographie

[1] J.R. Treichler, C.R. Johnson, JR., M.G. Larimore, *Theory and Design of the Adaptive Filters*, Prentice Hall, 2001.

[2] G. Caryannis, D. Manolakis, and N. Kalouptsidis, *A fast sequential algorithm for least squares filtering and prediction*, IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-31, pp. 1394-1402, Dec. 1983.

[3] J. Cioffi and T. Kailath, *Fast recursive least squares transversal filters for adaptive filtering*, IEEE Trans. Acoust, Speech, Signal Processing, vol. ASSP-32, pp. 304-337, Apr. 1984.

[4] G. Moustakides and S. Theodoridis, *Fast Newton transversal filters- A new class of adaptive estimation algorithms*, IEEE Trans. Signal Processing, vol. 39, no. 10, pp. 2184–2193, 1991.

[5] P. Mavridis, and G. Moustakides, *Simplified Newton-Type Adaptive Estimation Algorithms*, IEEE Trans. Signal Process, 1996, 44, (8).

[6] A. Benallal, and A. Benkrid, *A simplified FTF-type algorithm for adaptive filtering*, Signal processing, 2007, 87, (5), pp.904-917.

[7] M.Arezki, A. Benallal, P. Meyrueis, and D. Berkani, "*A New Algorithm with Low Complexity for Adaptive Filtering*", IAENG Journal, Engineering Letters, vol.18, Issue 3, 2010, pp.205-211.

[8] D.T.M.Slock , *On the convergence behaviour of the LMS and the normalized LMS algorithms*, *IEEE* Trans. Signal Processing, vol. 42, pp. 2811-2825, Mar. 1993.

[9] M. Rupp, *A family of adaptive algorithms with decorrelating proprieties*, IEEE Trans. Signal Processing, Vol. 46, NO. 3, 1998

[10] S. Gazor and T. Liu, *Adaptive filtering with decorrelating for coloured AR environments*, IEE Proceedings, 2006.