



UN ALGORITHME DU TYPE LMS A CAPACITE DE POURSUITE ET VITESSE DE CONVERGENCE ACCRUES

Jacob BENESTY, LI Sheng Wen, et Pierre DUHAMEL

CNET Centre Paris B et GRECO TDSI

38-40, rue du Général Leclerc, 92131 Issy-Les-Moulineaux, France

RÉSUMÉ

ABSTRACT

Cet article propose un algorithme de filtrage adaptatif du type gradient, de la famille du LMS. Cet algorithme requiert une charge de calcul plus faible que celle du LMS, tout en présentant une capacité de poursuite des variations du système encore accrue et une vitesse de convergence très nettement améliorée. On en prouve la convergence, et on donne les conditions pour qu'il converge plus vite que le LMS. Ces propriétés sont illustrées par des simulations de l'identification de l'écho acoustique de salle.

This paper presents a new adaptive filtering algorithm of the gradient type, LMS-like. This algorithm requires a lower arithmetic complexity than the LMS, while allowing a faster tracking of the variations of the system to be identified, and a significantly faster convergence rate. We provide a proof of convergence, together with the explicit conditions under which the new algorithm converges faster than the LMS. Simulations are provided in the context of acoustic echo cancellation.

1. INTRODUCTION

Les progrès de l'intégration des circuits aidant, on envisage maintenant l'utilisation de filtrage adaptatif de grande longueur dans de nombreuses applications. Cette grande taille pose d'une part le problème de la charge de calcul (on arrive aux limites de ce que la technologie sait faire à coût raisonnable), et d'autre part celui du comportement adaptatif (vitesse de convergence et capacité de poursuite des variations du système) qui est dégradé dans ces conditions.

Ces deux considérations motivent la recherche d'algorithmes ayant tout à la fois une charge de calcul faible et une convergence rapide.

Une réduction de charge de calcul conséquente passe obligatoirement par une recherche d'algorithmes en blocs efficaces, qui ne soient pas liés à l'utilisation de blocs trop longs, ceci à cause des problèmes de délai et de mémorisation.

Une première voie est ouverte par la disponibilité (récente) d'algorithmes de filtrage rapide en fréquence où la taille du bloc est indépendante de la taille du filtre [1]. On peut alors reprendre les équations du LMS, les réécrire vectoriellement, utiliser la FFT comme intermédiaire de calcul, et choisir un pas d'adaptation dans le domaine fréquence adapté à l'énergie du signal qui s'y trouve. L'algorithme ainsi obtenu (MDF) nécessite peu de calculs, converge vite, et offre beaucoup de flexibilité vis à vis de la taille de bloc. Cependant, il poursuit relativement mal les variations du système: le filtre reste fixe durant toute la durée d'un bloc, et l'algorithme d'adaptation "court" après le système tel qu'il était au bloc précédent. Cet algorithme a été publié simultanément par différents auteurs [2,7], et sert de référence pour l'algorithme proposé ici.

Une autre possibilité est apportée par une nouvelle formulation par blocs de l'algorithme LMS [3] qui permet d'écrire un nouveau critère par bloc qui est la minimisation de l'erreur qui aurait été fournie par le LMS durant un bloc de N sorties. Ceci correspond (fictivement) à faire fonctionner un algorithme LMS durant N échantillons, et à raffiner le résultat du gradient stochastique par un gradient exact portant sur l'ensemble des N dernières erreurs observées. L'algorithme correspondant (NBLMS) présente les mêmes caractéristiques que le MDF, avec des capacités de poursuite accrues.

Ces différents points sont illustrés par différentes simulations (LMS, MDF, NBLMS), dans le cadre de l'identification de l'écho acoustique d'une salle d'audioconférence: Une non stationnarité du système est simulée par le mouvement d'un réflecteur.

2. LE LMS ET SA VERSION EN BLOCS

Nous utiliserons les notations montrées sur le filtre adaptatif de la fig.1, le plus souvent sous la forme des notations vectorielles suivantes:

H_n : vecteur des coefficients à l'instant n

X_n : vecteur des L dernières entrées

L: taille du filtre

N: taille de bloc (on regroupera parfois les calculs correspondant à N sorties successives du filtre)

On supposera par ailleurs $L = N.K$

${}^tX_n = (X_n, X_{n-1}, \dots, X_{n-N+1})$ la matrice composée des N derniers vecteurs d'entrée

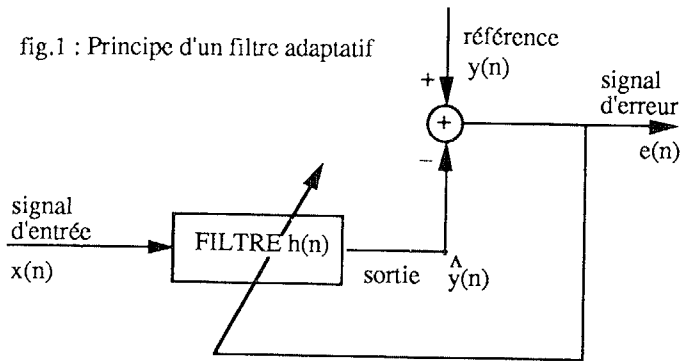
E_n : vecteur des N dernières erreurs du LMS

Y_n : vecteur des N dernières sorties de référence

Tous ces signaux sont supposés réels.



fig.1 : Principe d'un filtre adaptatif



2.1. LMS

Avec les notations ci-dessus, l'algorithme LMS s'écrit:

$$H_{n+1} = H_n + \mu X_n e(n) \tag{1}$$

$$e(n) = y(n) - \hat{X}_n H_n \tag{2}$$

$e(n)$ est appelée erreur a priori, car elle fait intervenir le filtre à l'instant n , avant adaptation. Lorsque le signal d'entrée est non-stationnaire, on choisit un pas d'adaptation $\mu(n)$ qui dépend du temps, de la manière suivante:

$$\mu(n) = \frac{\alpha}{\hat{X}_n X_n}, \quad 0 < \alpha < 2 \tag{3}$$

$\alpha < 2$ est nécessaire pour la convergence, et l'on choisit souvent $\alpha = 1$, choix qui annule l'erreur a posteriori [5] (erreur obtenue en reportant H_{n+1} dans l'eq.(2))

2.2. FELMS (Fast Exact LMS)

En filtrage fixe, une réduction de charge de calcul s'obtient toujours en regroupant les calculs correspondant à plusieurs sortie successives du filtre [1]. Il en est de même en filtrage adaptatif, et de nombreux auteurs ont proposé de réécrire les équations de l'algorithme LMS d'une manière vectorielle, ce qui fait apparaître les équations d'un filtre fixe [9]. Une réduction de la charge de calcul est alors possible, en utilisant par exemple des techniques de FFT [2]. Cependant, l'algorithme ainsi obtenu a un comportement adaptatif détérioré par rapport à l'algorithme initial, au niveau de la vitesse de convergence et de la capacité de poursuite. C'est pourquoi nous avons proposé en [3] un algorithme en bloc qui permet une réduction de la charge de calcul, tout en gardant exactement le même comportement adaptatif que le LMS dont il est dérivé. Nous en fournissons ci-dessous les équations, sans démonstration:

$$E_n = G_n (Y_n - \hat{X}_n H_{n-N+1}) \tag{4}$$

$$H_{n+1} = H_{n-N+1} + \mu \hat{X}_n E_n \tag{5}$$

Où G est défini de la manière suivante:

$$G_n = (I + \mu S_n)^{-1} \tag{6}$$

avec S_n une estimée de la matrice d'autocorrélation triangulaire inférieure:

$$S_n = \begin{bmatrix} 0 & 0 & \dots & 0 \\ s_1(n-N+2) & 0 & \dots & 0 \\ s_2(n-N+3) & s_1(n-N+3) & \dots & \vdots \\ \vdots & \vdots & \dots & \vdots \\ s_{n-1}(n) & s_{n-2}(n) & \dots & s_1(n) & 0 \end{bmatrix} \tag{7}$$

et $s_i(n) = \hat{X}_n X_{n-i}$, $i = 1, 2, \dots, N-1$

Le terme $\hat{X}_n H_{n-N+1}$ correspondant à un filtrage fixe, les techniques décrites en [1] s'appliquent, tout comme celles à base de FFT [2]. Une évaluation précise de la charge de calcul correspondante peut se trouver dans [3], montrant qu'une réduction substantielle peut être obtenue par rapport au LMS, d'autant plus notable que N est grand. L'important pour ce qui suit est que cet algorithme fournit un vecteur d'erreurs E_n qui est strictement identique à celui qui aurait été fourni par le LMS, et que cette erreur est fournie par une équation linéaire (4).

3. L'ALGORITHME PROPOSE (NBLMS)

Mathématiquement, l'algorithme que nous proposons minimise le critère suivant:

$$J_{n-N+1} = [E_n E_n] / N \tag{8}$$

où E_n est le vecteur d'erreur qui serait fourni par un LMS avec la même initialisation. Son fonctionnement sera donc équivalent à celui d'un LMS à l'intérieur d'un bloc de N données, alors que le filtre servant d'initialisation pour le bloc suivant sera celui minimisant la somme quadratique des N erreurs précédentes issues du LMS.

3.1. Dérivation de l'algorithme

Comme E_n est fourni par l'eq.(4) en fonction de H_{n-N+1} précédent, on développe:

$$J_{n-N+1} = \frac{1}{N} ([G_n Y_n] [G_n Y_n] - 2 [G_n Y_n] [G_n \hat{X}_n] H_{n-N+1} + H_{n-N+1} [G_n \hat{X}_n] [G_n \hat{X}_n] H_{n-N+1}) \tag{9}$$

dont le gradient est fourni par:

$$\hat{\nabla} = -\frac{2}{N} [G_n \hat{X}_n] E_n \tag{10}$$

L'algorithme de descente du gradient (qui est donc appliqué ici une fois par bloc) sera donc comme habituellement:

$$H_{n+1} = H_{n-N+1} - \frac{\mu_B}{2} \hat{\nabla} \tag{11}$$

L'ensemble de l'algorithme NBLMS est donc résumé comme suit:

$$G_n = [\mu S_n + I]^{-1} \tag{12}$$

$$E_n = G_n [Y_n - \hat{X}_n H_{n-N+1}] \tag{13}$$

$$H_{n+1} = H_{n-N+1} + \frac{\mu_B}{N} \hat{X}_n' G_n E_n \tag{14}$$

3.2. Charge de calcul

L'algorithme ci-dessus a une forme très voisine à celle de la formulation par blocs du §2.2, la différence résidant dans une multiplication supplémentaire par \hat{G}_n dans le calcul de la mise à jour des coefficients du filtre. Les techniques décrites dans [3] pour évaluer la charge de calcul du FELMS s'appliquent donc directement. Nous résumons les résultats ci-après, pour une

longueur de filtre telle que N soit une puissance de 2 : $L = N.K = 2^n K$:

Les coefficients de la matrice S_n se calculent récursivement, et $G_n E_n$, qui est la multiplication du vecteur E_n par l'inverse de $(I + \mu S_n)$ s'obtient très simplement par substitution, puisque cette matrice est triangulaire inférieure

Le filtrage fixe (13) se calcule par une FFT de longueur $2N$, dont les coefficients fréquentiels entrent dans des filtres complexes de longueur K , suivis d'une FFT inverse. Si tous les signaux considérés sont réels, et si la FFT est calculée par l'algorithme "split-radix", le NBLMS demande un total de:

$2(L/N)(n+2) + 2N + 3n - 9 + 6/N$ multiplications
 $2(L/N)(3n+2) + (5/2)N + 9n - 19/2 + 4L/N^2 + 12/N$ additions
 par point de sortie, ce qui correspond à une charge TOTALE de calcul (multiplications plus additions) réduite d'un facteur 4 pour un filtre de longueur $L=1024$ et une taille de bloc $N=128$.

3.3. Coherence avec le LMS

Ce gain en charge de calcul acquis, nous allons maintenant donner les conditions suffisantes pour que l'algorithme obtenu converge plus vite que le LMS.

Dans une première étape, nous supposons le signal d'entrée stationnaire, et le filtre assez long on aura alors $s_i(n) = s_i$, et $s_0 = X_n^T X_n$. On se placera également dans les conditions où l'erreur a posteriori du LMS est annulée (cf §1), soit $\mu s_0 = 1$.

La structure du NBLMS assure que, à l'intérieur d'un bloc, cette propriété est vérifiée et que H_{kN+1} , bien que jamais explicitement calculé, annule l'erreur a posteriori ε_{kN} . Nous allons maintenant montrer que cette propriété est partagée par le filtre calculé par le NBLMS pour initialiser le bloc suivant:

Soit ε'_{kN} cette erreur. D'après (5), on a:

$$\begin{aligned} e'_{kN} &= y_{kN} - X_{kN}^T H_{(k+1)N} \\ &= y_{kN} - X_{kN}^T \left(H_{kN} + \frac{\mu_B}{N} X_{kN}^T G_{kN} E_{kN} \right) \end{aligned} \quad (15)$$

choisissons $\mu_B/N = \mu$, il vient alors:

$$\begin{aligned} e_{kN} &= e_{kN} - \mu X_{kN}^T X_{kN} G_{kN} E_{kN} \\ &= e_{kN} - \mu [s_0 \quad s_1 \quad \dots \quad s_{n-1}]^T G_{kN} E_{kN} \\ &= e_{kN} - [1 \quad 0 \quad \dots \quad 0] E_{kN} \\ &= 0 \quad ! \end{aligned} \quad (16)$$

Comparé au LMS, le NBLMS garde donc les mêmes propriétés à l'intérieur d'un bloc, et permet d'annuler une deuxième fois l'erreur a posteriori tout en allant dans la direction du minimum correspondant au critère (8).

3.4. Convergence

Reprenons les équations d'adaptation du NBLMS et la réécriture en bloc du LMS donnée en (12-14). Nous allons montrer que l'écart par rapport au filtre optimal converge plus vite pour le NBLMS que pour le LMS. Soit l'écart

$$\Delta H_n = H_n - H_{opt} \quad (17)$$

Il s'écrit, pour le NBLMS:

$$\Delta H_n^{NBLMS} = \quad (18)$$

$$\begin{aligned} &H_{n-N+1} - H_{opt} + \frac{\mu_B}{N} X_n^T G_n G_n [Y_n - X_n H_{n-N+1}] \\ &= \Delta H_{n-N+1}^{NBLMS} \\ &+ \frac{\mu_B}{N} X_n^T G_n G_n [Y_n - X_n H_{n-N+1} - X_n H_{opt} + X_n H_{opt}] \\ &= \left(I - \frac{\mu_B}{N} X_n^T G_n G_n X_n \right) \Delta H_{n-N+1}^{NBLMS} \\ &+ \frac{\mu_B}{N} X_n^T G_n G_n [Y_n - X_n H_{opt}] \end{aligned}$$

et de la même manière, pour le LMS, on obtient à partir des équations de sa version en bloc (FELMS, décrite en § 2.2):

$$\begin{aligned} \Delta H_{n+1}^{LMS} &= \left(I - \mu X_n^T G_n X_n \right) \Delta H_{n-N+1}^{LMS} \\ &+ \mu X_n^T G_n [Y_n - X_n H_{opt}] \end{aligned} \quad (19)$$

Nous nous plaçons sous les mêmes conditions que dans le paragraphe précédent, en supposant que $\mu \ll 1$, ce qui se traduit par $G_n \approx I - \mu S_n$. Si l'on suppose de plus l'indépendance des variables suivantes: la suite des erreurs engendrée par le filtre optimal; le vecteur des données; et les écarts des coefficients $H_n - H_{opt}$, on peut montrer les résultats suivants:

-Les normes des écarts des solutions du LMS et du NBLMS avec l'optimum sont du même ordre de grandeur à la convergence.

-Le NBLMS converge plus vite que le LMS si la condition suivante est vérifiée:

$$\frac{\mu}{1 + \mu s_0 - \mu} \leq \mu_B \leq \frac{4\mu}{(1 + \mu s_0)^2} \quad (20)$$

Cette condition est obtenue en écrivant que le rayon spectral des matrices intervenant dans les récurrences (18) et (19) est plus petit dans le cas du NBLMS. Comme on sait par ailleurs que le LMS converge [6], cette condition est suffisante.

4. SIMULATIONS

4.1. conditions de simulation

Ces algorithmes ont été programmés pour l'identification d'une réponse impulsionnelle acoustique, mesurée dans une salle réelle (durée de la réponse: 128ms, fréquence d'échantillonnage: 16 kHz) avec un mouvement d'un réflecteur durant l'expérience, ce qui rend le système non-stationnaire. Une courbe de convergence comporte donc une convergence initiale, suivie après environ 13000 échantillons (0.8s) d'une période de croissance de l'erreur due à la poursuite des variations du système, et pour finir une deuxième convergence. Le signal d'entrée est un bruit à spectre semblable à la parole (bruit USASI). Un bruit blanc est rajouté en sortie du système, de rapport S/B égal à 40dB. Le système est modélisé par un filtre de longueur 1024, avec des blocs de taille $N=128$ pour le MDF (LMS fréquentiel-cf intro-) et le NBLMS.

4.2. convergence, système variable



Les courbes de convergence dans ce contexte réaliste montrent clairement que les algorithmes MDF et NBLMS convergent tous deux de la même manière, et sensiblement plus vite que le LMS, qui n'a pas eu le temps de converger complètement avant la période de poursuite. Le comportement du NBLMS est par ailleurs plus "lissé". De plus en période de variation du système, on voit clairement que les qualités de convergence rapide du MDF ne garantissent pas une poursuite satisfaisante du système, moins bonne en tout cas que dans le cas du NBLMS.

4.2. erreur de poursuite

Afin de quantifier ces propriétés de poursuite, nous avons placé en sortie du système à modéliser un gain qui varie linéairement de 1 à 2 entre les échantillons 26112 ($t=1.64s$) et 39168 ($t=2.46s$) [8]. La variation du système étant linéaire, les deux courbes d'erreur du LMS et du NBLMS montrent un palier horizontal durant ce mouvement. Le NBLMS améliore ce palier de près de 5 dB par rapport au LMS.

5. CONCLUSION

Nous avons présenté un algorithme -NBLMS- ,élaboré à partir du LMS, qui demande moins de calculs que le LMS, converge sensiblement plus vite, et poursuit mieux les variations du système à modéliser. Par rapport à l'implantation fréquentielle du LMS où le pas d'adaptation est choisi à chaque instant en fonction du contenu de chaque bande de fréquence (MDF), l'amélioration se situe au niveau poursuite, le prix à payer étant une charge de calcul légèrement plus élevée. Des tests ont par ailleurs été effectués avec des signaux de parole. Le MDF s'y est montré plus gêné par la non-stationnarité du signal que le NBLMS, dont les performances (auditives) peuvent se comparer à celles des algorithmes de moindres carrés récursifs, et ce avec une charge de calcul un ordre de grandeur plus faible.

REMERCIEMENTS

Ce travail a bénéficié de discussions avec K. Hilal et O. Rioul, CNET/PAB, A. Gilloire, CNET/LAA, qui nous a également fourni les données de simulation ainsi qu'avec les membres du projet "identification de réponses impulsionnelles longues" du GRECO TDSI

BIBLIOGRAPHIE

- [1] Z.J. Mou and P. Duhamel, "Short-Length FIR Filters and Their Use in Fast Nonrecursive Filtering," *IEEE Trans. on Signal. Proc.*, June 1991.
- [2] J.S. Soo and K.K. Pang, "Multidelay Block Frequency Domain Adaptive Filter" *IEEE Trans. on ASSP*, Feb. 1990, pp.373-376
- [3] J. Benesty, P. Duhamel, "A Fast Exact Least Mean Square Adaptive Algorithm" soumis pour publication à *IEEE Trans. on Signal Processing*, (version courte à ICASSP'90)
- [4] E.R. Ferrara, "Fast Implementation of LMS Adaptive Filters," *IEEE Trans. on ASSP*, Vol.30, Oct. 1982, pp.726-734.
- [5] M. Bellanger, "Analyse des signaux et filtrage numérique adaptatif," Masson et CNET-ENST, Paris, 1989
- [6] O. Macchi, M. Bellanger "Le point sur le filtrage adaptatif transverse" Actes du GRETSI, Nice, Juin 1987, pp.1G-14G.
- [7] P.C.W. Sommen, "On the convergence properties of a partitioned block frequency domain adaptive Filter (PBFDAF)" *Proc. of EUSIPCO'90*, pp.373-376.
- [8] A. Gilloire, "A Comparison of NLMS and fast RLS Algorithms for the Identification of Time-varying Systems with noisy Outputs" *Proc. of EUSIPCO'90*, pp.417-420.
- [9] G.A. Clark, S.K. Mitra, et S.R. Parker, "Block Implementation of Adaptive Digital Filters," *IEEE Trans. on CAS*, Vol.28, June. 1981, pp.377-384.

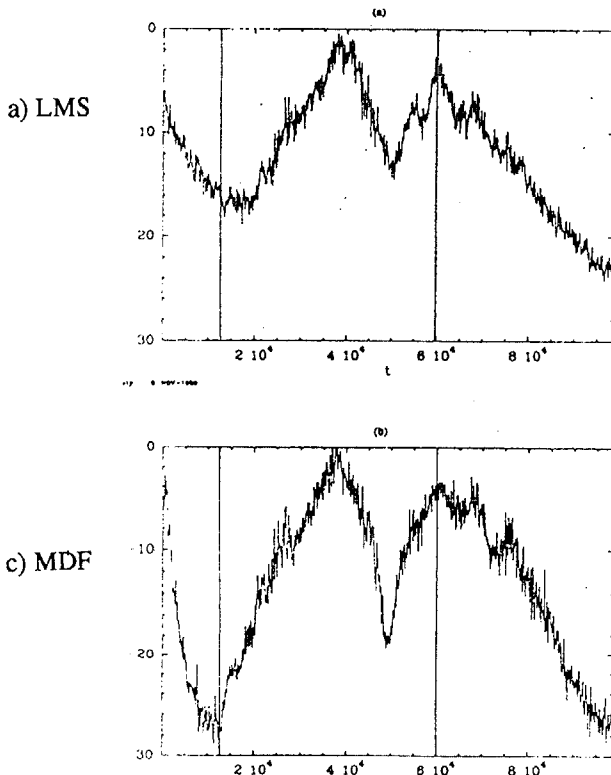


fig.2 courbes de convergence en présence d'une variation du système à modéliser. La première zone est la convergence, la deuxième une zone de poursuite, suivie d'une re-convergence. a)LMS b)NBLMS c)MDF

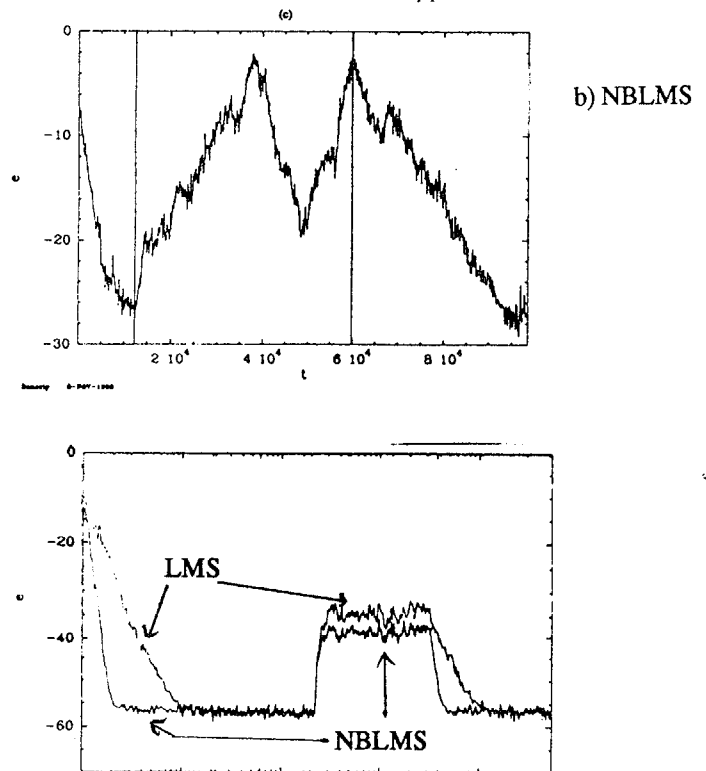


fig.3 poursuite des variations linéaires d'un système. Le plateau correspond à l'erreur de modélisation pendant les variations linéaires.