Sur la réduction de la complexité du filtre LMS adaptatif de Volterra du second ordre

Mounir Sayadi[(1)], Farhat Fnaiech[(1)] et Mohamed Najim[(2)]

[(1)] E.S.S.T.T. 5 Av. Taha Hussein, 1008, Tunis, Tunisie

[(2)] Equipe Signal & Image, ENSERB, Av. Albert Schweitzer, BP 99, F33402 Talence cedex, France
Email : najim@goelette.tsi.u-bordeaux.fr

RESUME

Dans cette communication, nous présentons une approche utilisant une convolution rapide pour réduire la complexité de calcul de l'algorithme LMS pour le filtrage adaptatif quadratique de Volterra. Les travaux utilisant la convolution rapide pour réduire la complexité de calcul de l'algorithme LMS ont uniquement porté sur le filtrage linéaire, où on arrive à réduire de 25 % le nombre de multiplications au prix d'une augmentation de 25% du nombre d'additions. Par contre, dans le cas du filtrage adaptatif de Volterra du second ordre, on obtient une réduction de la complexité de calcul de l'algorithme LMS se traduisant par une diminution du nombre de multiplications de 50% au prix d'une augmentation de 20% du nombre d'additions.

ABSTRACT

In this paper we present an efficient LMS adaptive algorithm for the quadratic filtering. A fast convolution algorithm, only applied until now in the linear case, reduces the number of multiplications by close to 50%, at the expense of 20% more additions. The algorithm decomposes the convolution equation into three parts. The first and the second parts are function of both input signals and filter coefficients, which accounts for almost all the required computation of the algorithm whereas the third part depends only on the input signal. The proposed algorithm has one more parameter vector adaptation than the classical LMS to take advantage of the fast convolution algorithm.

1. Introduction

Plusieurs approches ont été développées pour réduire la complexité de calcul de l'algorithme LMS dans le domaine temporel pour le filtrage linéaire [1]-[3]. Dans [1], Benesty et Duhamel ont présenté un algorithme adaptatif de performances comparables à celles de l'algorithme LMS classique mais permettant la réduction du nombre de multiplications avec une légère augmentation du nombre d'additions. S. G. Chen et al. ont proposé [2] une approche basée sur un algorithme de convolution rapide qui réduit la complexité de calcul du filtre LMS linéaire. Cette approche a été étendue au cas du filtrage linéaire adaptatif dans [3] où elle permet de réduire de 25 % le nombre de multiplications au prix d'une augmentation de 25% du nombre d'additions. Dans cette communication, nous mettons à profit l'idée de la convolution rapide afin de réduire la complexité de calcul de l'algorithme LMS pour le filtrage adaptatif quadratique de Volterra [4]. On montre que cette alternative va permettre de réduire le nombre de multiplications de 50% au prix d'une augmentation de 20% du nombre d'additions.

L'algorithme proposé décompose l'équation de convolution en trois parties. Les première et deuxième parties dépendent du signal d'entrée et des coefficients du filtre. Elles sont à l'origine de la plus forte complexité de calcul. La troisième partie dépend uniquement du signal d'entrée.

Par rapport au filtre LMS quadratique classique de Volterra, l'approche proposée adapte un vecteur supplémentaire de paramètres, afin de tirer avantage de la convolution rapide pour diminuer la complexité de calcul de la deuxième partie.

2. Transformation du la partie quadratique du filtre de Volterra du second ordre

La partie quadratique du filtre de Volterra du second ordre peut être exprimée par l'équation:

$$y(n) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} b_{i,j}(n) x(n-i) x(n-j)$$

$$= \sum_{i=0}^{N-1}\left[\sum_{j=0}^{N-1} b_{i,j}(n)x(n-j)\right]x(n-i) \qquad (1)$$

En utilisant l'algorithme de convolution rapide [3] pour le terme entre crochets, l'équation (1) devient:

$$y(n) = \sum_{i=0}^{N-1}\sum_{j=0}^{N/2-1}\left[\overbrace{\left(x(n-2j)+b_{i,2j+1}(n)\right)\left(x(n-2j-1)+b_{i,2j}(n)\right)}^{\text{première partie}} - \overbrace{b_{i,2j}(n)b_{i,2j+1}(n)}^{\text{deuxième parie}} - \overbrace{x(n-2j)x(n-2j-1)}^{\text{troisième partie}}\right] x(n-i) \quad (2)$$

On se propose de remplacer le terme:

$$\sum_{j=0}^{N/2-1} b_{i,2j}(n)b_{i,2j+1}(n) \qquad (0<i<N-1) \qquad (3)$$

par un vecteur additif $H(n)$ dont les éléments $h_i(n)$ sont adaptés par:

$$h_i(n+1) = h_i(n) + \mu_1 e(n)x(n-i). \qquad (4)$$

Pour le dernier terme de la décomposition (2), on définit la quantité $P(n)$ par:

$$P(n) = \sum_{j=0}^{N/2-1} x(n-2j)x(n-2j-1).$$

$P(n)$ peut être estimée récursivement par:

$$P(n) = P(n-2) + x(n)x(n-1) - x(n-N)x(n-N-1) \qquad (5)$$

On note, comme cela a été proposé en [3], que des places mémoires supplémentaires sont utilisées pour le stockage de $P(n-2)$ et des produits $x(n-1)x(n-2)$ à $x(n-N)x(n-N-1)$ pour une utilisation ultérieure dans le calcul de $P(n)$, ce qui réduit encore la complexité de calcul de $P(n)$ à seulement une seule multiplication et deux additions.

Enfin, on définit un terme m_i pour remplacer le produit $e(n)x(n-i)$ qui se répète dans le calcul de $h_i(n+1)$ et de $b_{i,j}(n+1)$.

Ainsi, la nouvelle approche de filtrage quadratique basée sur le LMS est donnée par:

Partie Filtre:

$$P(n) = P(n-2) + x(n)x(n-1) - x(n-N)x(n-N-1)$$

$$e(n) = \sum_{i=0}^{N-1}\left(\sum_{j=0}^{N/2-1}\left[\left(x(n-2j)+b_{i,2j+1}(n)\right)\left(x(n-2j-1)+b_{i,2j}(n)\right)\right] - h_i(n) - P(n)\right)x(n-i) - s(n)$$

Partie adaptation:

$$m_i = e(n)x(n-i)$$

$$h_i(n+1) = h_i(n) + \mu_1 m_i$$

$$b_{i,j}(n+1) = b_{i,j}(n) + \mu_2 m_i x(n-j)$$

La complexité de calcul de cette approche est donné dans le tableau 1. On montre qu'elle est égale à $2.5N^2+3N+1$ multiplications et $2.5N^2+3N+2$ additions. Par contre, d'aprés le tableau 2, la complexité de l'algorithme LMS classique est de $5N^2$ multiplications et $2N^2$ additions.

3. Résultats de simulations

L'approche ainsi développée est utilisée pour estimer les paramètres d'un système quadratique de Volterra du second ordre d'entrée Gaussienne $x(n)$ et de sortie $s(n)$ dans le cas stationnaire sans bruit additif avec μ_1=0.01 and μ_2=0.03.

$$s(n) = 0.7x(n)^2 - 0.4x(n)x(n-1) + 2.2x(n)x(n-2) + 1.56x(n)x(n-3) - 0.2x(n-1)^2 + 0.5x(n-1)x(n-2) - 1.26x(n-1)x(n-3) + 0.66x(n-2)^2 + 2.4x(n-2)x(n-3) + 0.44x(n-3)^2$$

La figure 1 montre, pour l'algorithme LMS classique et pour l'algorithme à complexité réduite que nous proposons, l'évolution de la norme du vecteur d'erreur sur les coefficients définie par:

$$E_1(n) = 10.\log\frac{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\left(b_{i,j}(n)-b_{i,j}^0\right)^2}{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}(b_{i,j}^0)^2}$$

où $b_{i,j}^0$ sont les coefficients du système désiré.

La figure 2 montre l'évolution de la différence en (dB) entre $h_i(n)$ et $\sum_{j=0}^{N/2-1} b_{i,2j}(n)b_{i,2j+1}(n)$ définie par:

$$E_2(n) = 10.\log\frac{\sum_{i=0}^{N-1}\left(h_i(n)-h_i^0(n)\right)^2}{\sum_{i=0}^{N-1}\left(h_i^0(n)\right)^2}$$

où $h_i^0(n) = \sum_{j=0}^{N/2-1} b_{i,2j}(n) b_{i,2j+1}(n)$.

4. Conclusion

Dans cette communication, nous avons proposé une alternative au filtre LMS adaptatif de Volterra du second ordre qui réduit considérablement la complexité de cet algorithme. Nous avons montré que cette nouvelle approche réduit le nombre de multiplications de 50% au prix d'une augmentation de 20% du nombre d'additions. Ceci peut être très bénéfique lors d'une implantation en temps réel.

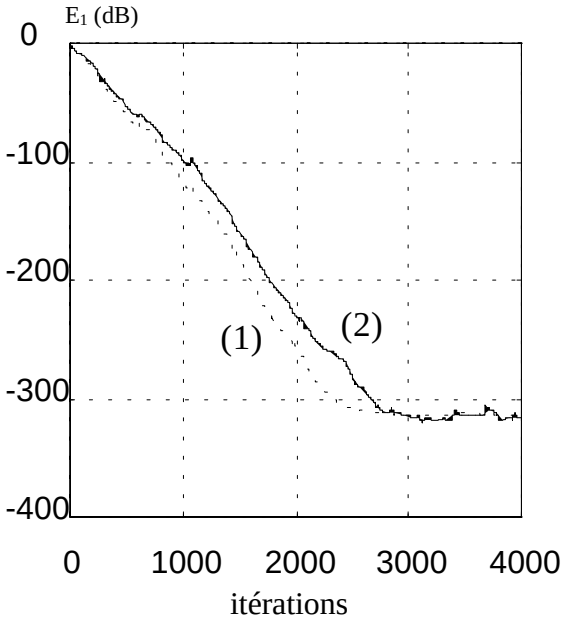

Figure 1: Norme du vecteur d'erreur sur les coefficients pour (1)LMS classique (2) nouvelle approche.

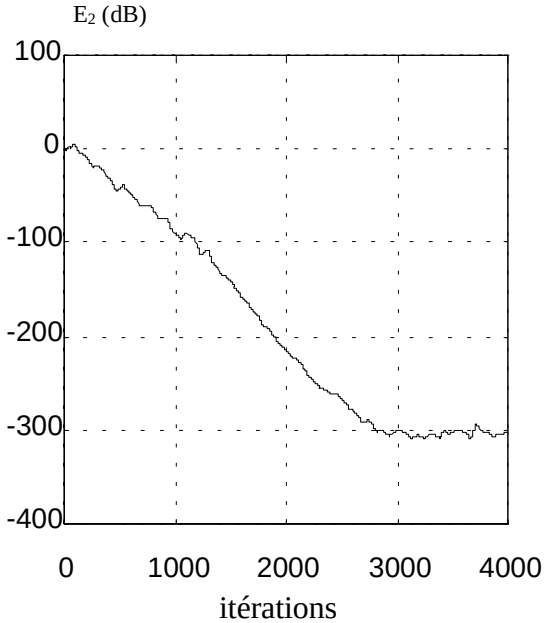

Figure 2: Convergence du terme ajouté $h_i(n)$.

5. Références:

[1] J. Benesty and P. Duhamel, "Fast exact LMS adaptive algorithm," *IEEE Trans. on Signal Processing*, Vol. SP-40, no. 12, pp.2904-2920, Dec 1992.

[2] S. G. Chen and R. T. Tsay, "A novel fast filtering algorithm and its implementation," in *Proceedings of EUSIPCO'92*, Brussels, Aug. 1992, pp.945-948.

[3] S. G. Chen, Y. A. Kao and R. T. Tsay, "A new efficient LMS adaptive filtering algorithm," *IEEE Trans. Circuits and Systems*, Vol. CAS-43, no.5, pp.372-385, May 1996.

[4] G. L. Sicuranza, "Quadratic filters for signal processing," *Proc. of IEEE*, Vol. 80, No.8, pp. 1263-1285, Aug. 1992.

	nombre de multiplications	nombre d'additions
Partie Filtre: $P(n) = P(n-2) + x(n)x(n-1) - x(n-N)x(n-N-1)$	1	2
$e(n) = \sum_{i=0}^{N-1}\left(\sum_{j=0}^{N/2-1} \left[\left(x(n-2j) + b_{i,2j+1}(n)\right)\left(x(n-2j-1) + b_{i,2j}(n)\right)\right] - h_i(n) - P(n) \right) x(n-i) - s(n)$	$0.5N^2+N$	$1.5N^2+2N$
Partie adaptation: $m_i = e(n)x(n-i)$	N	0
$h_i(n+1) = h_i(n) + \mu_1 m_i$	N	N
$b_{i,j}(n+1) = b_{i,j}(n) + \mu_2 m_i x(n-j)$	$2N^2$	N^2
Complexité totale de calcul	$2.5N^2+3N+1$	$2.5N^2+3N+2$

Tableau 1 : Complexité de calcul de la nouvelle approche de filtrage quadratique basée sur le LMS.

	nombre de multiplications	nombre d'additions
Partie Filtre: $e(n) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} b_{i,j}(n)x(n-i)x(n-j) - s(n)$	$2N^2$	N^2
Partie adaptation: $b_{i,j}(n+1) = b_{i,j}(n) + \mu e(n)x(n-i)x(n-j)$ (i,j: 0 à N-1)	$3N^2$	N^2
Complexité totale de calcul	$5N^2$	$2N^2$

Tableau 2 : Complexité de calcul de l'algorithme LMS classique.