

Un algorithme Quasi-Newton dans le domaine fréquentiel pour l'égalisation de transmission en salves

K. Berberidis¹, S. Rantos¹, J. Palicot²

¹ Dept of Computer Engineering & Informatics

University of Patras, 26500, Rio, Greece

berberid@ceid.upatras.gr, rantos@ceid.upatras.gr

² FTR&D/DMR/DDH

BP59,35512 Cesson-Sévigné, France

Jacques.palicot@francetelecom.com

Résumé - Dans ce papier, nous proposons un nouveau type d'égaliseur qui appartient à la famille des algorithmes Quasi-Newton (QN). Nous présentons un Egaliseur Linéaire Transverse et un Egaliseur à Retour de Décisions. Dans le premier cas, le Hessien est approché par un développement en série de matrices de Toeplitz. Cette formulation nous permet de développer un algorithme performant dans le domaine fréquentiel (DF). Cette même approche est utilisée pour l'algorithme d'adaptation du filtre direct de l'ERD. L'algorithme obtenu ainsi offre à la fois les avantages de vitesse de convergence des algorithmes QN et de moindre complexité due à la formulation dans le DF.

Abstract - In this paper two new adaptive equalizers are proposed which belong to the Quasi-Newton (QN) algorithmic family. The first equalizer is a Linear Transversal Equalizer (LTE) and the second is a Decision Feedback Equalizer (DFE). In the LE case the involved inverse Hessian matrix is approximated by a proper expansion consisting of the powers of a Toeplitz matrix. Due to this formulation the algorithm can be efficiently implemented in the transform domain (TD) using FFT. The same idea is applied to the Feedforward part of the DFE. The derived algorithms exhibit faster convergence (advantage of QN algorithm) and less complexity as compared to other Newton type algorithms (implementation in the TD).

1. INTRODUCTION

Dans cette étude nous nous intéressons à l'égalisation de systèmes de transmission sans fil mobile en mode TDMA. La raison principale de dégradation de la qualité du signal reçu est l'interférence entre symboles. Comme le canal peut varier entre la réception de deux salves de données, voire à l'intérieur d'une salve, il est particulièrement important dans de telles conditions que le système d'égalisation puisse poursuivre ces variations et cela le plus rapidement possible[1]. C'est la raison pour laquelle nous étudions des algorithmes qui possèdent une grande vitesse de convergence. De plus comme cette égalisation doit se faire en temps réel, la complexité arithmétique doit être la plus faible possible, ce qui explique notre proposition de réaliser l'algorithme dans le domaine transformé.

Les algorithmes de type QN sont un bon compromis entre la simplicité des algorithmes de type LMS et la vitesse de convergence de ceux de type RLS. L'équation d'adaptation du filtre par un algorithme de type QN implique d'inverser le Hessien, ou au minimum d'approcher cet inverse. Les performances de ce type d'algorithme dépendent principalement de la manière dont est inversé le Hessien[2]. Dans ce papier le Hessien est approché par une matrice de

Toeplitz, qui est construite à partir de l'estimée de la matrice d'autocorrélation du canal. Pour s'affranchir de la complexité algorithmique de l'inversion de cette matrice $O(M^2)$ opérations[3], avec M la longueur de l'égaliseur, nous approchons cet inverse par le développement d'une série de matrices de Toeplitz. Nous traitons les deux cas correspondant à l'égaliseur linéaire transverse (LTE) et à l'égaliseur à Retour de Décisions (ERD).

2. POSITION DU PROBLEME

Le canal discret équivalent est supposé être à réponse impulsionnelle finie avec du bruit additif blanc centré gaussien. L'alphabet est constitué de symboles i.i.d avec une moyenne nulle et une variance σ_d^2 . La sortie du canal échantillonné à la fréquence symbole est notée $\{x\}$ alors l'algorithme proposé pour l'égaliseur LTE est donné par les équations

$$y(n) = \mathbf{w}_M^H(n) \mathbf{x}_M(n) \quad (1)$$

$$e(n) = d(n) - y(n) \quad (2)$$

$$\mathbf{w}_M(n+1) = \mathbf{w}_M(n) + \mu \mathbf{R}^{-1} \mathbf{x}_M(n) e^*(n)$$

avec :

$$\mathbf{x}_M(n) = [x(n + M_1) \cdots x(n + M_1 - M + 1)]^T$$

le vecteur de données d'entrée de longueur M, \mathbf{W}_M le vecteur coefficient et $y(n)$ la sortie de l'égaleur.

R est la matrice de Toeplitz construite à partir de l'estimation de la séquence d'autocorrélation r_0, r_1, \dots, r_{M-1} de $\{x\}$.

$$\text{Avec } r_k(n) = \sum_{i=1+k}^n \beta^{n-i} x(i) x^*(i-k)$$

L'inversion de R par un algorithme de type Levinson requiert $O(M^2)$ opérations. Pour éviter cette difficulté nous proposons une autre façon de réaliser cette inverse, présentée ci-après.

2.1 Inversion de la matrice d'autocorrélation

Nous présentons une manière de réaliser cet inverse simplement en utilisant le théorème bien connu suivant[4]: Soient ζ_1, \dots, ζ_M les valeurs propres de la matrice S. Soit

$$\rho(\mathbf{S}) = \max_i |\zeta_i|$$

Alors la série $\sum_{k=0}^{\infty} \mathbf{S}^k$ converge ssi $\rho(\mathbf{S}) < 1$. Sous ces conditions I-S est non-singulière et la limite de la série est égale à $(\mathbf{I} - \mathbf{S})^{-1}$.

Nous considérons la matrice $\frac{1}{\gamma} \mathbf{R} = \mathbf{I} + \mathbf{A}$ avec γ constante réelle positive et avec

$$\mathbf{A} = \begin{bmatrix} \left(\frac{r_0}{\gamma} - 1 \right) & \frac{r_1}{\gamma} & \dots & \frac{r_{M-1}}{\gamma} \\ \frac{r_1^*}{\gamma} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \frac{r_1}{\gamma} \\ \frac{r_{M-1}^*}{\gamma} & \dots & \frac{r_1^*}{\gamma} & \left(\frac{r_0}{\gamma} - 1 \right) \end{bmatrix}$$

Dans notre cas $\mathbf{S} = -\mathbf{A}$ est une matrice Hermitienne alors, en appliquant le théorème précédent,

$$\left(\frac{1}{\gamma} \mathbf{R} \right)^{-1} = (\mathbf{I} + \mathbf{A})^{-1}$$

$$\mathbf{R}^{-1} = \frac{1}{\gamma} (\mathbf{I} - \mathbf{A} + \mathbf{A}^2 - \mathbf{A}^3 + \mathbf{A}^4 - \dots)$$

Pour pouvoir appliquer correctement le théorème la constante γ doit être choisie de manière à ce que le rayon spectral de S soit inférieur à 1, nous avons choisi

$$\gamma = 0.5|r_0| + \sum_{k=1}^{M-1} |r_k|$$

On peut approcher cet inverse en ne conservant que les q premiers termes de la somme alors dans ce cas

$$\mathbf{R}^{-1} \approx \frac{1}{\gamma} \sum_{i=0}^{q-1} (-1)^i \mathbf{A}^i$$

2.2 Algorithme pour le LTE

Nous approchons en ne conservant que les deux premiers termes. $\hat{\mathbf{R}}^{-1} = \frac{1}{\gamma} (\mathbf{I} - \mathbf{A} + \mathbf{A}^2)$ alors l'équation de mise à jour des coefficients devient :

$$\mathbf{w}_M(n+1) = \mathbf{w}_M(n) + \frac{\mu e^*(n)}{\gamma} (\mathbf{x}_M(n) - \mathbf{A} \mathbf{x}_M(n) + \mathbf{A}^2 \mathbf{x}_M(n))$$

Comme la matrice A est une matrice de Toeplitz nous enveloppons A dans une matrice circulante C, ce qui nous permet d'effectuer les différents produits de manière simple dans le DF. Si F est l'opérateur DFT de dimension 2M alors

$$\mathbf{F} \mathbf{C} \mathbf{F}^{-1} = \text{diag}(\mathbf{F} \mathbf{C}) = \mathbf{D}$$

$$\text{et } \begin{bmatrix} \mathbf{A} \mathbf{x}_M(n) \\ \mathbf{0}_M \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{O}_M \\ \mathbf{O}_M & \mathbf{O}_M \end{bmatrix} \mathbf{F}^{-1} \mathbf{D} \mathbf{F} \begin{bmatrix} \mathbf{x}_M(n) \\ \mathbf{0}_M \end{bmatrix}$$

L'équation de mise à jour peut être modifiée comme suit :

$$\begin{bmatrix} \mathbf{w}_M(n+1) \\ \times \end{bmatrix} = \begin{bmatrix} \mathbf{w}_M(n) \\ \times \end{bmatrix} + \frac{\mu e^*(n)}{\gamma} \left(\begin{bmatrix} \mathbf{x}_M(n) \\ \times \end{bmatrix} - \begin{bmatrix} \mathbf{A} \mathbf{x}_M(n) \\ \times \end{bmatrix} + \begin{bmatrix} \mathbf{A}^2 \mathbf{x}_M(n) \\ \times \end{bmatrix} \right)$$

en prenant la DFT de chaque membre nous obtenons

$$W(n+1) = W(n) + \frac{\mu e^*(n)}{\gamma} (Q_0(n) - Q_1(n) + Q_2(n))$$

$$\text{Avec } W(n) = \mathbf{F} \begin{bmatrix} \mathbf{w}_M(n) \\ \times \end{bmatrix}, Q_i(n) = \mathbf{F} \begin{bmatrix} \mathbf{A}^i \mathbf{x}_M(n) \\ \times \end{bmatrix}$$

$$\text{Soit le vecteur dans le DF } X(n) = \mathbf{F} \begin{bmatrix} \mathbf{x}_M(n) \\ \mathbf{0}_M \end{bmatrix}$$

Nous définissons $Q_0(n) = X(n)$ et en prenant une hypothèse de stationnarité entre plusieurs itérations nous avons :

$$X_k(n) = X_k(n-1) e^{-j \frac{\pi k}{M}} + x(n + M_1) - (-1)^k x(n + M_1 - M)$$

Nous définissons cette dernière opération par l'équation suivante avec recdfst signifiant l'opération dft inverse.

$$X(n) = \text{recdfst}(X(n-1), x(n + M_1), x(n + M_1 - M))$$

alors

$$Q_0(n) = \text{recdft}(Q_0(n-1), x(n+M_1), x(n+M_1-M))$$

De plus avec $Q_1(n) = \mathbf{F} \begin{bmatrix} \mathbf{A}\mathbf{x}_M(n) \\ \mathbf{0}_M \end{bmatrix}$ nous obtenons

$$\mathbf{D}Q_1(n) = \mathbf{F} \begin{bmatrix} \mathbf{A}^2\mathbf{x}_M(n) \\ \mathbf{B}\mathbf{A}\mathbf{x}_M(n) \end{bmatrix} \text{ et } Q_2(n) = \mathbf{D}Q_1(n)$$

En définissant $\mathbf{z}(n-1) = \mathbf{A}\mathbf{x}_M(n-1)$

On observe que

$$\mathbf{A}\mathbf{x}_M(n) = \begin{bmatrix} \mathbf{a}^T \mathbf{x}_M(n) \\ \mathbf{z}_{0:M-2}(n-1) \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{a}_{1:M-1}^* \end{bmatrix} x(n+M_1) - \begin{bmatrix} 0 \\ \mathbf{a}_{1:M-1}^R \end{bmatrix} x(n+M_1-M)$$

Nous définissons les 2 vecteurs suivants dans le

$$\text{DF } A_1 = \mathbf{F} \begin{bmatrix} 0 \\ \mathbf{a}_{1:M-1}^* \\ \mathbf{0}_M \end{bmatrix}, A_2 = \mathbf{F} \begin{bmatrix} 0 \\ \mathbf{a}_{1:M-1}^R \\ \mathbf{0}_M \end{bmatrix}$$

Alors nous pouvons écrire

$$Q_1(n) = \text{recdft}(Q_1(n-1), \mathbf{a}^T \mathbf{x}_M(n), \mathbf{a}^{RH} \mathbf{x}_M(n-1)) + A_1 x(n+M_1) - A_2 x(n+M_1-M)$$

Nous observons que l'erreur peut être calculée en utilisant des informations connues dans le DF

$$\begin{aligned} e(n) &= d(n) - w_M^H(n) x_M(n) \\ &= d(n) - \begin{bmatrix} \mathbf{w}_M(n) \\ \times \end{bmatrix}^H \begin{bmatrix} \mathbf{x}_M(n) \\ \mathbf{0}_M \end{bmatrix} \\ &= d(n) - \begin{bmatrix} \mathbf{w}_M(n) \\ \times \end{bmatrix}^H \frac{\mathbf{F}^H \mathbf{F}}{2M} \begin{bmatrix} \mathbf{x}_M(n) \\ \mathbf{0}_M \end{bmatrix} \\ &= d(n) - \frac{1}{2M} W^H(n) X(n) \end{aligned}$$

Finalement nous obtenons l'algorithme suivant pour le cas linéaire :

Variables définies à l'initialisation $\mu, \lambda, W(0), P_k(-1)$

Variables mises à jour à chaque bloc

$$r_k, \gamma, \mathbf{a}, \mathbf{D}, A_1, A_2, Q_1$$

Pour $n=0, 1, \dots$;

$$Q_0(n) = X(n) ;$$

$$Q_1'(n) = \text{recdft}(Q_1(n-1), \mathbf{a}^T \mathbf{x}_M(n), \mathbf{a}^{RH} \mathbf{x}_M(n-1))$$

$$Q_1(n) = Q_1'(n) + A_1 x(n+M_1) - A_2 x(n+M_1-M) ;$$

$$Q_2(n) = \mathbf{D}Q_1(n) ; Q(n) = Q_0(n) - Q_1(n) + Q_2(n)$$

$$e(n) = d(n) - \frac{1}{2M} W^H(n) X(n) ;$$

$$P_k(n) = \lambda P_k(n-1) + (1-\lambda) |X_k(n)|^2 ;$$

$$\mathbf{M}(n) = \text{diag}(P_1^{-1}(n), \dots, P_{2M}^{-1}(n))$$

$$W(n+1) = W(n) + \frac{\mu e^*(n)}{\gamma} \mathbf{M}(n) Q(n)$$

Nous obtenons une complexité de 15M multiplications plus 4 FFT pour chaque mise à jour de la matrice d'auto corrélation ce qui peut se faire à chaque bloc de données transmises.

2.3 Algorithme pour l'ERD

La sortie de l'ERD à l'itération n est donnée par

$$y(n) = \mathbf{c}_{M_f}^H(n) \mathbf{x}_{M_f}(n) - \mathbf{b}_{M_b}^H(n) \mathbf{d}_{M_b}(n)$$

avec \mathbf{d}_M le vecteur coefficient du filtre récursif. Cette équation peut se réécrire

$$\begin{aligned} y(n) &= \begin{bmatrix} \mathbf{c}_{M_f}^H(n) & -\mathbf{b}_{M_b}^H(n) \end{bmatrix} \begin{bmatrix} \mathbf{x}_{M_f}(n) \\ \mathbf{d}_{M_b}(n) \end{bmatrix} \\ &= \mathbf{w}_M^H(n) \mathbf{u}_M(n) \end{aligned}$$

avec le vecteur coefficient

$$\mathbf{w}_M(n) = \begin{bmatrix} \mathbf{c}_{M_f}(n) & -\mathbf{b}_{M_b}(n) \end{bmatrix}^T \text{ et le vecteur de}$$

$$\text{données } \mathbf{u}_M(n) = \begin{bmatrix} \mathbf{x}_{M_f}(n) & \mathbf{d}_{M_b}(n) \end{bmatrix}^T$$

Dans le cas d'un algorithme QN, nous ne pouvons pas appliquer directement l'algorithme précédent avec ces nouveaux vecteurs. En effet la matrice d'autocorrélation du vecteur d'entrée n'est plus une matrice de Toeplitz. Elle est définie par les relations suivantes

$$\begin{aligned} \mathbf{R}_u &= E[\mathbf{u}_M(n) \mathbf{u}_M^H(n)] \\ &= E \left[\begin{bmatrix} \mathbf{x}_{M_f}(n) \\ \mathbf{d}_{M_b}(n) \end{bmatrix} \begin{bmatrix} \mathbf{x}_{M_f}^H(n) & \mathbf{d}_{M_b}^H(n) \end{bmatrix} \right] \\ &= E \left[\begin{bmatrix} \mathbf{x}_{M_f}(n) \mathbf{x}_{M_f}^H(n) & \mathbf{x}_{M_f}(n) \mathbf{d}_{M_b}^H(n) \\ \mathbf{d}_{M_b}(n) \mathbf{x}_{M_f}^H(n) & \mathbf{d}_{M_b}(n) \mathbf{d}_{M_b}^H(n) \end{bmatrix} \right] \\ &= \begin{bmatrix} \mathbf{R}_x & \mathbf{R}_{dx}^H \\ \mathbf{R}_{dx} & \mathbf{R}_d \end{bmatrix} \end{aligned}$$

Avec \mathbf{R}_d matrice diagonale en supposant les données iid. Son inversion est triviale. La difficulté vient des matrices correspondant aux termes croisés. Nous approchons en forçant ces matrices à 0, nous avons donc

$$\hat{\mathbf{R}}_u = \begin{bmatrix} \hat{\mathbf{R}}_x & \mathbf{0}_{M_f, M_b} \\ \mathbf{0}_{M_b, M_f} & \sigma_d^2 \mathbf{I} \end{bmatrix}$$

Dans ce cas nous appliquons l'algorithme précédent au filtre direct avec un algorithme classique de LMS dans le DF pour le filtre récursif.

3. Résultats de simulation

Nous avons réaliser des simulations pour deux canaux dont les réponses impulsionnelles sont les suivantes :

Canal 1 : (0.9333, 0.5012, 0.5129, 0.5370) aux instants(1, 8, 15, 22).

Canal 2 : (0.7490, 1, .0299, 0.316, 0.055 ,0.1580) aux instants(1, 2, 18, 26, 34, 40). Clairement le second canal génère plus d'IES que le premier.

La longueur du filtre du LTE est de 128. Celle du filtre direct de l'ERD est de 16 et celle du filtre arrière de 64. Nous comparons les convergences de notre algorithme avec les algorithmes TD/LMS, NLMS et Newton/LMS. La matrice des pas de convergence pour les algorithmes en bloc est mise à jour suivant la manière proposée dans[5]. Nous constatons que notre algorithme approche les performances de l'idéal (Newton/LMS) et est meilleur que le TD/LMS (voir les figures 1 et 2 pour le LTE). Les résultats du DFE pour les canaux 1 et 2 respectivement sur les figures 3 et 4 sont eux aussi très bons, malgré la simplification apportée dans la matrice d'autocorrélation en forçant à zéro les matrices des termes croisés.

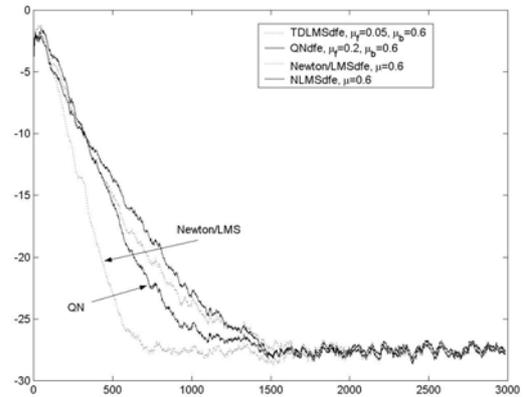


Figure3 : CANAL 1 : égaliseur à retour de décisions

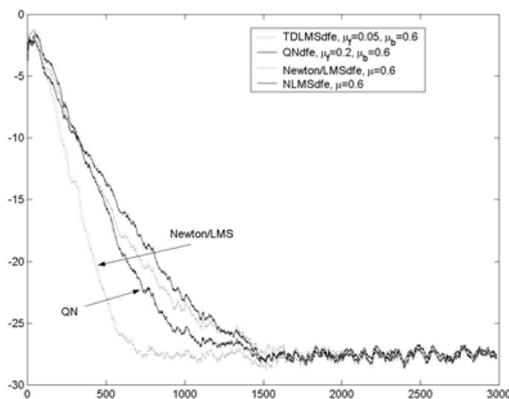


Figure 1 : CANAL 1 :Egaliseur linéaire

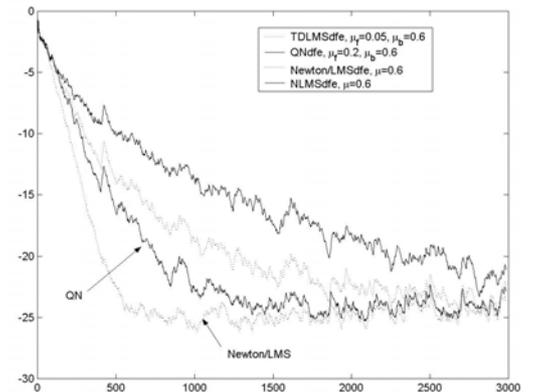


Figure4 : CANAL 2 : égaliseur à retour de décisions

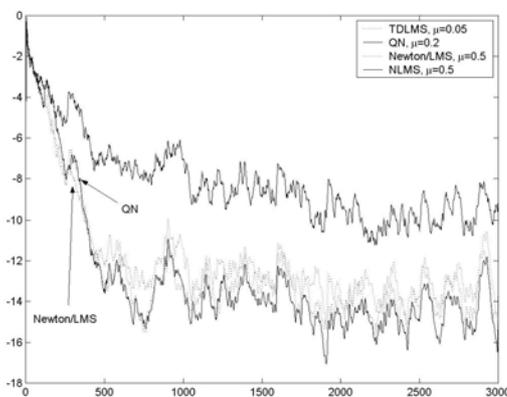


Figure 2 : CANAL 2 :Egaliseur linéaire

Références

- [1] T.S.Rappaport, « Wireless Communication : principle and practice », Prentice-hall, 1996.
- [2] G Glentis, K.Berberidis, S.Theodoridis, « efficient Least Squares Adaptive Algorithms for FIR transversal Filtering : A unified view », IEEE Signal Processing Magazine, vol 16, n°4, july 1999
- [3] D.F.Marshall, W.K.Jenkins, « A fast Quasi-newton adaptive filtering algorithm », IEEE Trans on SP, vol 40, july 1992
- [4] Y.Saad, « Iterative Methods for Sparse Linear Systems », Pws publishing company, 1996
- [5] Jae Chon Lee,Chong Kwan Un, « Performance of transform Domain LMS Adaptive Digital Filters », IEEE Tr ASSP, vol 34, June 1986