

Egaliseur autodidacte à retour de décisions à configuration variable: comparaisons entre algorithme "itératif" et "bloc"

Sébastien HOUCKE, Christophe LAOT, Joël LABAT

Département Signal et Communication
ENST-Bretagne

sebastien.houcke@enst-bretagne.fr, christophe.laot@enst-bretagne.fr,
joel.labat@enst-bretagne.fr

Résumé – Dans cet article, nous proposons une nouvelle façon de mettre en oeuvre l'égaliseur autodidacte à retour de décisions à configuration variable initialement développé dans [1] au moyen d'un algorithme itératif. Nous montrons que nous nous affranchissons alors du temps de convergence inhérente à toute méthode itérative tout en conservant les mêmes performances asymptotiques. Par contre cette amélioration nécessite un coût de calcul nettement supérieur à celui des procédures itératives, coût que nous évaluons pour les deux méthodes.

Abstract – In this article, we propose a new algorithm for the blind adaptive Decision-Feedback Equalizer with self optimized configuration. This equalizer was first introduced in [1], we now show that with our new algorithm we do not have any convergence time and reach the same asymptotical performances than before. This is achieved by an important increase of computational cost which is evaluated for both methods.

1 Introduction

Du fait de leur nombreux avantages, les égaliseurs à retour de décision (ERD) sont couramment utilisés afin d'éliminer dans les signaux de communication la distorsion induite par les canaux de propagation. En effet même lorsque le canal est sévère et fortement bruité, il est possible après convergence de l'égaliseur d'atteindre des performances très acceptables : i.e. une faible erreur quadratique moyenne en sortie de l'égaliseur et cela pour un coût de calcul beaucoup plus faible que d'autres techniques performantes tel que par exemple le maximum de vraisemblance. Le canal de propagation étant inconnu, l'ERD doit être implémenté de manière adaptative. Dans un ERD classique, l'adaptation se fait par l'intermédiaire d'une séquence d'apprentissage, ce qui diminue évidemment le débit utile de la transmission. C'est pourquoi dans [1] les auteurs ont proposé un ERD ne nécessitant aucune séquence d'apprentissage. Cet égaliseur aveugle comporte deux modes de fonctionnement : dans le mode initial, l'égaliseur est constitué de la cascade d'un filtre blanchissant purement récursif \mathcal{R} , d'un filtre transverse \mathcal{T} et d'un correcteur de phase \mathcal{CP} . L'originalité du dispositif est lié au fait que chaque étage est adapté selon un critère spécifique, ce qui confère à la fois robustesse et rapidité de convergence. Lorsque le processus d'égalisation est suffisamment avancé, ce qui peut être apprécié à l'examen de l'erreur quadratique moyenne estimée, la structure et les critères d'adaptation de l'égaliseur sont modifiés de telle sorte que le dispositif se transforme en un égaliseur à retour de décisions (ERD) conventionnel. Le caractère réversible de cette modification apporte au nouvel égaliseur un avantage substantiel au sens où chaque fois qu'il le peut, il tire profit de ses propres décisions sans risque de divergence, contrairement aux ERD conventionnels.

Dans cet article, nous nous proposons de décrire ce dispositif et de détailler l'algorithme itératif initialement développé pour

sa mise en oeuvre. Nous verrons ensuite comment s'affranchir de la période de convergence inhérente à tout algorithme itératif et évaluerons le coût de calcul supplémentaire d'une telle amélioration. Enfin des comparaisons entre ces deux procédures sont présentées.

2 Rappels sur l'ERD à structure variable

2.1 Principe

Soit $d(k)$ une séquence i.i.d. de données discrètes de moyenne nulle et de variance unité et $H(z) = \sum_{l=0}^N h(l)z^{-l}$ la fonction de transfert du canal de propagation. Le signal observé est corrompu par un bruit blanc gaussien additif $n(k)$ de variance σ_n^2 . Le signal reçu s'exprime donc ainsi :

$$s(k) = \sum_{l=0}^N d(k-l)h(l) + n(k)$$

Nous rappelons que l'égaliseur linéaire MMSE (solution de Wiener) admet comme fonction de transfert :

$$C(z) = \frac{\sigma_d^2 H^*(1/z^*)}{\sigma_d^2 H(z) H^*(1/z^*) + \sigma_n^2} \quad (1)$$

Habituellement cet égaliseur linéaire est implémenté par un filtre transverse (FIR) afin d'éviter tous risques d'instabilité. Cependant, une implémentation par un filtre récursif (IIR) comporte un certain nombre d'avantages [1]:

1. Elle nécessite en général moins de coefficients que la mise en oeuvre par filtre transverse.
2. L'ordre optimal pour le filtre à retour de décision est N .
3. Enfin, elle permet d'évoluer simplement d'un schéma d'égalisation linéaire vers un schéma non linéaire du type retour de décisions.

En effet, le dénominateur de $C(z)$ n'est autre que la densité spectrale de puissance (d.s.p.) $S_s(z)$ du signal observé $s(k)$. Donc si N représente le nombre de zéros de la fonction de transfert du canal $H(z)$ alors $S_s(z)$ possèdent $2N$ racines, N d'entre elles étant à l'intérieures du cercle unité et les N autres étant à l'extérieur. Dès lors la d.s.p. $S_s(z)$ peut se factoriser ainsi : $S_s(z) = AG(z)G^*(1/z^*)$ avec $G(z)$ un polynôme à phase minimale et A une constante (réelle) positive :

$$G(z) = \prod_{k=1}^N (1 - z_k z^{-1}) \quad \text{avec} \quad |z_k| < 1$$

La fonction de transfert de l'égaliseur $C(z)$ s'écrit alors :

$$C(z) = \frac{1}{G(z)} \frac{\sigma_d^2 H^*(1/z^*)}{A G^*(1/z^*)}$$

On peut alors implémenter le filtre $\frac{1}{G(z)}$ comme un filtre récursif stable. De plus ce filtre n'est autre que le filtre blanchissant les observations puisque la densité spectrale de puissance de sa sortie est égale à A . Donc l'égaliseur linéaire optimal (en terme EQM) peut être implémenté comme décrit par la figure 1, où $A(z) = G(z) - 1$ et $B(z) = \frac{\sigma_d^2 H^*(1/z^*)}{A G^*(1/z^*)}$.

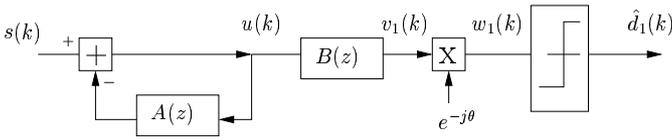


FIG. 1: Structure de l'égaliseur : mode initial

En comparant l'égaliseur linéaire optimal avec l'ERD conventionnel (figure 2), on constate que l'on peut passer facilement de l'un à l'autre par un simple changement de structure : inversion de l'ordre des filtres $A(z)$ et $B(z)$.

Dans un contexte aveugle, il reste à trouver des critères permettant d'aboutir à une solution proche de celle du MMSE.

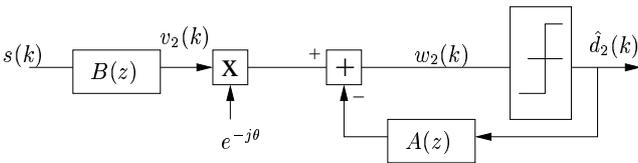


FIG. 2: Structure de l'égaliseur : mode final

2.2 Mise en oeuvre - Algorithme itératif

La solution retenue dans [1] consiste à mettre en place un algorithme itératif. Pour déterminer le filtre $A(z) = \sum_{k=1}^{L_A} a_k z^{-k}$, il suffit de minimiser

$$J_1(\mathbf{A}) = \mathbb{E}[|u(k)|^2]$$

Avec $\mathbf{A} = (a_1, a_2, \dots, a_{L_A})^T$. L'algorithme itératif permettant d'aboutir à la solution désirée se traduit alors ainsi : à l'itération k nous aurons :

$$\begin{aligned} u(k) &= s(k) - \mathbf{A}^{(k-1)T} \mathbf{U}_{L_A}(k-1) \\ \mathbf{A}^{(k)} &= \mathbf{A}^{(k-1)} + \nu_A u(k) \mathbf{U}_{L_A}^*(k-1) \end{aligned}$$

où $\mathbf{A}^{(0)} = [0, 0, \dots, 0]^T$ et $\mathbf{U}_{L_A}(k-1) = (u(k-1), \dots, u(k-L_A))^T$.

Pour déterminer le filtre

$$B(z) = \frac{\sigma_d^2 H^*(1/z^*)}{A G^*(1/z^*)}$$

, il existe plusieurs algorithmes aveugles de déconvolution. Le critère de Godard [2] dans un contexte non bruité permet d'obtenir un égaliseur $B(z)$ de type Zero-forcing. Lorsque le bruit n'est pas trop fort, le filtre $B(z)$ résultant de la minimisation du critère de Godard est proche de la solution MMSE recherchée. L'algorithme itératif permettant de minimiser le critère de Godard est le suivant :

$$\begin{aligned} v_1(k) &= \mathbf{B}^{(k-1)T} \mathbf{U}_{L_B+1}(k) \\ \mathbf{B}^{(k)} &= \mathbf{B}^{(k-1)} + \nu_B v_1(k) (1 - |v_1(k)|^2) \mathbf{U}_{L_B+1}^*(k) \end{aligned}$$

où $\mathbf{B}^{(0)} = [0, \dots, 0, 1, 0, \dots, 0]^T$ et ν_B un pas positif correctement choisi.

Etant donnée qu'en sortie du CMA, on obtient une estimée des symboles à un retard près et une constante complexe près, il est nécessaire de mettre en place une procédure de correction de phase. Le critère retenu pour cela s'écrit :

$$J_3(\theta) = \mathbb{E}[|v_1(k)e^{-j\theta} - \hat{d}_1(k)|^2] \quad (2)$$

où $\hat{d}_1(k)$ correspond au $k^{\text{ième}}$ symbole décidé. L'algorithme itératif permettant de minimiser (2) peut être trouvé dans [4].

Une fois que les filtres $A(z)$ et $B(z)$ sont proches de leur valeur finale (i.e. lorsque l'erreur quadratique estimée est inférieure à un seuil), le changement de structure se produit. Les filtres $A(z)$, $B(z)$ et θ sont alors adaptés de telle sorte à minimiser le critère (2). En reprenant les notations de la figure 2, l'algorithme du gradient stochastique correspondant s'exprime ainsi :

$$\begin{aligned} \mathbf{A}^{(k)} &= \mathbf{A}^{(k-1)} + \nu_A (\hat{d}_2(k) - w_2(k)) \hat{\mathbf{D}}^*(k) \\ \mathbf{B}^{(k)} &= \mathbf{B}^{(k-1)} + \nu_B (\hat{d}_2(k) - w_2(k)) \exp(j\theta(k-1)) \mathbf{T}^*(k) \\ \theta(k) &= \theta(k-1) + \nu_\theta \left(\epsilon(k) + \beta \sum_{i=1}^k \epsilon(i) \right) \end{aligned}$$

où :

$$\begin{aligned} y(k) &= \left(\mathbf{B}^{(k-1)T} \mathbf{T}(k) \right) \exp(j\theta(k-1)) \\ \mathbf{T}(k) &= [s(k), \dots, s(k-L_B)]^T \\ \hat{\mathbf{D}}(k) &= [\hat{d}_2(k-1), \dots, \hat{d}_2(k-L_A)]^T \\ \epsilon(k) &= \text{Im}\{w_2(k)(\hat{d}_2(k) - w_2(k))^*\} \end{aligned}$$

et où ν_θ est un pas positif correctement choisi, β un paramètre positif et $\theta(0) = 0$. Les principaux intérêts d'un tel algorithme sont tout d'abord son faible coût de calcul, sa réactivité (d'autant plus primordiale que le canal change vite). Son inconvénient majeur réside en son temps de convergence (même s'il est aussi rapide que l'algorithme classique du CMA itératif). Afin d'être compatible avec une utilisation dans des systèmes de communication par bloc, une procédure dite par 'burst'. Elle permet de réduire le temps de convergence. Elle part du constat suivant : à la fin d'un bloc, les paramètres \mathbf{A} , \mathbf{B} et θ sont plus proches de leur valeur optimale qu'au début du bloc. L'idée consiste donc à itérer la procédure sur un même bloc mais en

initialisant l'algorithme avec les valeurs obtenues en fin de bloc lors de la précédente itération. De cette manière, il est possible de réduire considérablement le temps de convergence de l'algorithme moyennant une perte de réactivité : le canal de propagation est supposé stationnaire sur la durée du bloc.

2.3 Evaluation du coût de calcul

Pour évaluer le coût de calcul, nous comptons le nombre d'opérations (additions ou multiplications) réalisées au cours de l'algorithme. Pour l'algorithme itératif classique, en fixant $L_A = 5$ et $L_B = 19$, il faut de l'ordre de $200L$ opérations avec L le nombre de symboles dans le bloc. Si maintenant on réalise l'algorithme itératif 'burst' sur des blocs de taille L et en effectuant M itérations par bloc, nous obtenons de l'ordre de $200LM$ opérations nécessaires. Lors des simulations du chapitre 4, nous avons fixé $M = 3$, en effet dans le contexte simulé, nous n'avons pas constaté d'amélioration significative lorsque $M > 3$.

3 Algorithme bloc

Une autre possibilité pour réduire le temps de convergence, consiste à mettre cet algorithme sous forme de traitement bloc. Tout comme pour la technique 'burst', on suppose le canal stationnaire sur la durée du bloc. Par contre, comme on utilise l'ensemble des données du bloc pour aboutir à la solution, la notion de "durée de convergence" n'existe plus : on obtient les mêmes performances que l'on soit en début ou en fin de bloc. Le principe d'un algorithme du gradient à pas optimal est le suivant: on veut optimiser une certaine fonction de coût $J(x)$ suivant la variable x en utilisant toutes les données du bloc.

1. Initialiser $k = 0$ et $x^{(k)} = x^{(0)}$
2. Evaluer les dérivées du critère

$$\left. \frac{\partial J}{\partial x} \right|_{x=x^{(k)}}.$$

3. Déterminer le pas optimal ν_{opt} qui minimise

$$J\left(x^{(k)} - \nu \left. \frac{\partial J}{\partial x} \right|_{x=x^{(k)}}\right).$$

4. Mise à jour :

$$x^{(k+1)} = x^{(k)} - \nu_{opt} \left. \frac{\partial J}{\partial x} \right|_{x=x^{(k)}}$$

5. Itérer les points 2,3 et 4 jusque convergence de l'algorithme (i.e. norme des dérivées en $x^{(k)}$ inférieure à un seuil).

3.1 Application à l'algorithme précédent

Pour déterminer le filtre $A(z)$, on minimise grâce à un algorithme du gradient à pas optimal la fonction $J(A) = \mathbb{E}[|u(k)|^2]$. L'ensemble du bloc est utilisé pour estimer cette fonction :

$$\hat{J}_1(\mathbf{A}) = \frac{1}{L} \sum_{k=1}^L \left| s(k) - \sum_{l=1}^{L_A} a(l)u(k-l) \right|^2.$$

Pour le filtre $B(z)$ nous minimisons la fonction $J_2(\mathbf{B}) = \mathbb{E}[(|v_1(k)|^2 - 1)^2]$. Pour θ nous minimisons la fonction $J_3(\theta) = \mathbb{E}[|v_1(k)e^{-j\theta} - \hat{d}_1(k)|^2]$. Une fois ces paramètres déterminés, nous changeons de structure et passons à une structure d'ERD conventionnel. Cette fois nous allons minimiser la fonction $J(A, B, \theta) = \mathbb{E}[|w_2(k) - \hat{d}_2(k)|^2]$ par un algorithme du gradient initialisé avec les valeurs calculées à l'étape précédente.

Si l'EQM en sortie de l'ERD est meilleure que celle en sortie de l'égalisation linéaire, alors on conserve les symboles décidés en sortie de l'ERD, sinon ce sont ceux en sortie de l'égaliseur linéaire qui sont conservés.

3.2 Evaluation du coût de calcul

Nous avons là encore compté le nombre d'opérations réalisées par ce nouvel algorithme. Pour un bloc de taille L , $L_A = 5$ et $L_B = 19$, $27000T$ opérations sont nécessaires. Soit de l'ordre de 45 fois plus que l'algorithme itératif 'burst'.

4 Simulations

Nous considérons un émetteur utilisant une modulation MDP4. Le modèle de propagation considéré est le suivant. Le facteur d'excès de bande du filtre d'émission est fixé à 0.2, le canal de propagation résulte de la superposition de 3 trajets. Le retard du premier trajet est arbitrairement fixé à zéros tandis que les deux autres sont tirés aléatoirement entre $[0; 3T]$ suivant une loi uniforme avec $1/T$ le rythme d'émission. Le module de l'atténuation du premier trajet est fixé à un, les autres sont tirés aléatoirement entre $[0; 1]$ suivant une loi uniforme. L'argument de l'atténuation est tiré aléatoirement entre $[0; 2\pi]$ suivant une loi uniforme. Enfin le filtre est normalisé. L'observation est acquise au moyen d'une seule antenne. Le bruit est supposé complexe, blanc et gaussien. Le rapport signal à bruit est fixé à 15dB.

Le filtre $A(z)$ possède $L_A = N + 1$ coefficients avec N le nombre de coefficients du canal. Le filtre B en possède $L_B = 19$. Les résultats ont été obtenus en moyennant 3000 réalisations où pour chaque réalisation le canal et les symboles ont été tirés aléatoirement. Le critère de performance retenu est l'erreur quadratique estimée (i.e. $|w_2(n) - \hat{d}_2(n)|^2$) moyennée (EQEM) sur le nombre de réalisations. De cette manière, on peut apprécier l'évolution de la qualité de l'égalisation en fonction du temps.

La figure 3 compare les vitesses de convergence entre l'algorithme itératif classique et l'algorithme itératif 'burst'. On considère des blocs de $L = 1000$ symboles. La courbe en trait plein représente l'EQEM obtenue avec l'algorithme itératif classique tandis que celle en pointillés représente l'EQEM obtenue avec la technique 'burst'. A la vue des résultats de cette simulation, nous comparerons dans la suite des simulations l'algorithme bloc avec uniquement l'algorithme itératif 'burst'.

La figure 4 compare l'EQEM obtenue avec la technique itérative 'burst' avec celle obtenue par l'algorithme bloc. La taille du bloc considérée est égale à $L = 1000$ symboles. Une fois que l'algorithme itératif a convergé, on constate qu'avec un rapport signal à bruit de 15dB, on obtient des performances comparables avec les deux méthodes. Par contre, avec l'algorithme bloc, seuls les L_A premiers symboles du bloc sont mal

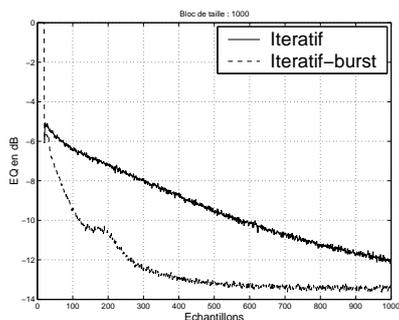


FIG. 3: Vitesse de convergence des algorithmes itératifs

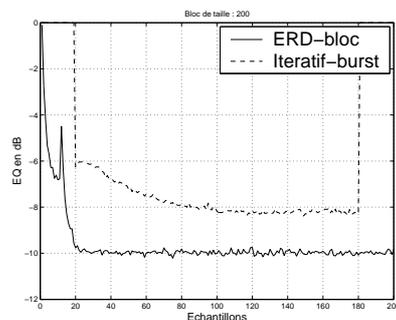


FIG. 6: Erreur quadratique estimée moyennée - $L = 200$

estimés.

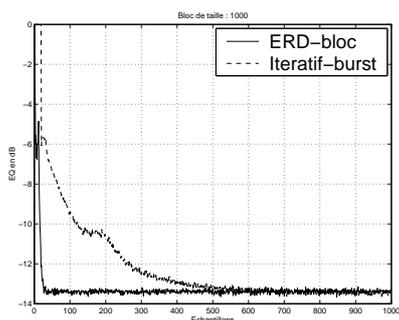


FIG. 4: Erreur quadratique estimée moyennée - $L = 1000$

Il est possible de réduire la taille des blocs, la figure 5, est obtenue avec une taille de bloc égale à $L = 500$ symboles. Les performances obtenues en fin de bloc avec l'algorithme itératif sont comparables avec celle obtenues avec l'algorithme bloc. Mais, encore une fois, avant d'obtenir ces performances, il faut attendre environ les trois quart de la durée du bloc alors qu'avec l'algorithme blocs seuls les L_A premiers symboles sont mal estimés.

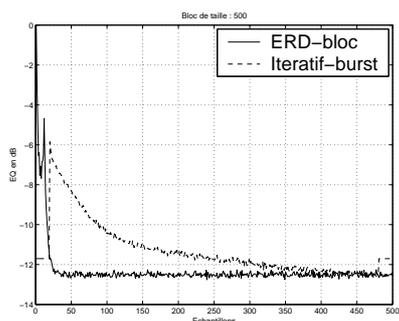


FIG. 5: Erreur quadratique estimée moyennée - $L = 500$

Enfin la figure 6 est obtenue avec des blocs de taille $L = 200$. Cette taille peut être considérée comme la taille critique nécessaire au bon fonctionnement des algorithmes.

Sur des blocs de très petite taille, l'algorithme bloc est nettement meilleur que l'algorithme itératif 'burst'.

La figure 7 illustre un autre avantage de l'algorithme bloc sur son homologue itératif, sa robustesse face au bruit.

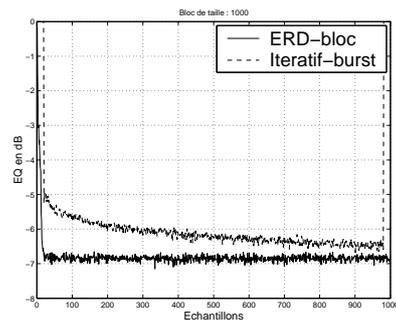


FIG. 7: Influence du Bruit - $L = 1000$, SNR= 7dB

5 Conclusion

La mise en oeuvre de l'égaliseur autodidacte à retour de décision à configuration variable au moyen d'un algorithme traitant les données par bloc permet de s'affranchir de la période de convergence des algorithmes itératifs tout en garantissant la même performance asymptotique. Elle permet aussi de traiter des données sur des blocs très courts (200 symboles). Enfin l'algorithme bloc est plus robuste au bruit que son homologue itératif. Le prix de ces améliorations est un surcoût en terme de temps de calcul (d'un facteur 45). Surcoût qui peut être prohibitif suivant l'application considérée.

References

- [1] J. Labat, O. Macchi, and C. Laot, "Adaptive decision feedback equalization : can you skip the training period ?," *IEEE Trans. on Communication*, vol. 46, no. 7, pp. 921–930, July 1998.
- [2] D.N. Godard, "Self recovering equalization and carrier tracking in two dimensional data communications systems," *IEEE Trans. on Communications*, vol. 28, no. 11, pp. 1867–1875, November 1980.
- [3] O. Shalvi and E. Weinstein, "New criteria for blind deconvolution of nonminimum phase systems (channels)," *IEEE Trans. on Information theory*, vol. 36, no. 2, pp. 312–321, March 1990.
- [4] D.D Falconer, "Jointly adaptive equalization and carrier recovery in two-dimensional data communication systems," *Bell Syst. Tech. J.*, vol. 55, pp. 317–334, March 1976.