



**ÉGALISATION ADAPTATIVE AUTODIDACTE PAR  
RÉTROPRÉDICTION ET PRÉDICTION**

**O. MACCHI<sup>1</sup>, C. A. FARIA DA ROCHA<sup>1,2</sup>, J. M. TRAVASSOS-ROMANO<sup>3</sup>**

<sup>1</sup>Laboratoire des Signaux et Systèmes et GDR Traitement du Signal et Images,  
CNRS-ESE, Plateau de Moulon, 91192 Gif-sur-Yvette

<sup>2</sup>Universidade Federal de Santa Catarina, Brésil, <sup>3</sup>Universidade Estadual de Campinas, Brésil

**RÉSUMÉ** - Cet article apporte deux contributions originales. D'une part une nouvelle structure d'égaliseur autodidacte avec un rétroprédicteur, un prédicteur et un gain qui s'affranchit de la multiplicité des points stationnaires et corrige les canaux les plus sévères. D'autre part, un nouveau critère d'optimisation qui permet le passage continu du blanchiment à la restitution des niveaux. L'erreur de contrôle utilise une non linéarité de sortie qui est réglable. Ce système a une vitesse bien supérieure à celle des autres égaliseurs autodidactes.

**ABSTRACT** - This paper brings two original contributions. First a novel structure of self-learning equalizer which puts in cascade a backward predictor, a forward predictor and a gain. This suppresses the problem of multiple stationary points and allows equalization of very severe channels. Then a novel adaptation using a control error with a variable nonlinearity which switches smoothly from a whiteness objective to a true data level recovery criterion. This system has lower arithmetical complexity and better speed, than other self learning equalizers.

**I. INTRODUCTION**

L'égalisation adaptative  $\mathfrak{H}$  d'un canal  $\mathfrak{F}$  transmettant une suite inconnue de données  $a_k$  indépendantes est faite classiquement avec l'algorithme auto adaptatif du gradient, en prenant pour signal de référence les données  $\hat{y}_k$  restituées à la sortie du système de décision selon

$$H_k = H_{k-1} + \mu \varepsilon(y_k) X_k, \quad \mu > 0 \quad (1.1)$$

où  $H_{k-1}$  est le vecteur d'égalisation transverse à l'instant  $k$ ,  $X_k$  le vecteur des échantillons  $x_k$  reçus; où

$$y_k = H_{k-1}^T X_k \quad (1.2)$$

$$\varepsilon(y_k) = \hat{y}_k - y_k \quad (1.3)$$

sont respectivement la sortie de l'égaliseur et la classique erreur de décision, fonction non linéaire de  $y_k$ . Une telle autoadaptation requiert une bonne initialisation  $H_k$ , ce qui est usuellement obtenu par une phase préalable dans laquelle les  $a_k$  sont connus. Dans les réseaux multipoint, et aussi en communications radio-mobiles, il faudrait s'affranchir de cette contrainte. Alors l'égaliseur devient autodidacte. Divers algorithmes adaptatifs ont entrepris cette tâche en modifiant la fonction d'erreur  $\varepsilon$ . Ainsi, les erreurs respectives de Sato [1], Godard [2], Benvéniste [3]

$$\varepsilon^S(y) = \left( E(a_k^2) / E(|a_k|) \right) \text{sgn}(y) - y; \quad (1.4)$$

$$\varepsilon^G(y) = \left( E(a_k^4) / E(a_k^2) \right) y - y^3; \quad (1.5)$$

$$\varepsilon^B(y) = \alpha_1 |\varepsilon(y)| \varepsilon^S(y) + \alpha_2 \varepsilon(y), \quad \alpha_1, \alpha_2 > 0, \quad (1.6)$$

ainsi que l'algorithme Stop-Go de Picchi et Prati [4].

Mais ces algorithmes ont deux inconvénients :

1 - Leurs performances sont très sensibles à la proximité vers le cercle unité (U) des zéros de la fonction de transfert  $F(z)$  de  $\mathfrak{F}$ .

2 - Ils admettent un grand nombre de points stationnaires et  $H_k$  peut sauter de l'un à l'autre, en même temps que changent de manière inconnue le retard  $\delta$  de restitution de données (entier  $\delta$  pour lequel  $\hat{y}_k = sa_{k-\delta}$ ) ainsi que la polarité de restitution (valeur  $\pm 1$  pour  $s$ ).

Le nouvel égaliseur autodidacte que nous proposons atténue beaucoup le premier inconvénient et s'affranchit totalement du second, ceci grâce à sa structure nouvelle, non plus transverse, mais récursive. Enfin il introduit un nouveau critère d'adaptation, qui repose au départ sur le blanchiment des  $y_k$  successifs, avant d'utiliser la fonction d'erreur  $\varepsilon$ . Ceci augmente considérablement la vitesse de convergence.

**II. TYPES DE CANAUX**

On dit que le canal  $\mathfrak{F}$  est de type "I" (pour intérieur) s'il est à minimum de phase. Sa fonction de transfert

$$F_I(z) = \prod_{i=1}^N (1 - z_i^I z^{-1}) \quad , \quad |z_i^I| < 1 \quad (2.1)$$

à tous ses ( $N$ ) zéros  $z_i^I$  à l'intérieur de U. De même  $\mathfrak{F}$  est de type "E" (pour extérieur) s'il est à maximum de phase, sa fonction de transfert

$$F_E(z) = \prod_{j=1}^Q \left( z^{-1} - (z_j^E)^{-1} \right) \quad , \quad |z_j^E| > 1 \quad (2.2)$$

ayant tous ses ( $Q$ ) zéros  $z_j^E$  à l'extérieur de U. Normalement les canaux de transmission  $\mathfrak{F}$  sont de type M (pour mixte) avec une fonction de transfert

$$F_M(z) = f F_I(z) F_E(z) \quad . \quad (2.3)$$



Soit  $\Phi$  la phase du canal ( $F(z) = |F(z)| e^{j\Phi(z)}$ ). Elle est d'autant plus non linéaire que  $F(z)$  a des zéros plus proches de  $U$  ce qui nécessite plus de coefficients dans  $\mathfrak{H}$ . Pour apprécier les performances des algorithmes, nous avons adopté trois canaux types.

**Canal "I" :**

$$F_I(z) = (1 - \rho_I e^{j\phi_I} z^{-1}) (1 - \rho_I e^{-j\phi_I} z^{-1}). \quad (2.4)$$

On a choisi,  $\phi_I = 120^\circ$ ,  $\rho_I = 0.9$ .

**Canal "E" :**

$$F_E(z) = (z^{-1} - 1/\rho_E e^{j\phi_E}) (z^{-1} - 1/\rho_E e^{-j\phi_E}) \quad (2.5)$$

$\phi_E = 60^\circ$ ,  $\rho_E = 1.1$

**Canal "M" :** "I" et "E" en cascade.

Pour ces trois canaux, les figures 1 et 2 illustrent respectivement l'atténuation et le déphasage. Le canal M est particulièrement sévère, avec deux évanouissements et un déphasage très non linéaire.

### III. ÉGALISATION PAR RÉTROPRÉDICTION ET PRÉDICTION

En l'absence de bruit,  $\mathfrak{H}$  a pour fonction de transfert  $H(z) = F^{-1}(z)$ . Mais il est instable dès que  $F(z)$  possède un zéro extérieur à  $U$ . On résoud ce problème en acceptant un retard  $\delta$  pour restituer les  $a_k$ . D'après (2.3)

$$H(z) = f^{-1} P(z) R(z), \quad (3.1)$$

où

$$R(z) = z^{-\delta} F_E^{-1}(z), \quad P(z) = F_I^{-1}(z). \quad (3.2)$$

L'expression (3.1) montre que  $\mathfrak{H}$  admet une représentation canonique comme la cascade des deux filtres  $\mathfrak{R}$  et  $\mathfrak{P}$  et d'un simple gain  $\mathfrak{G}$ , de fonctions de transfert respectives  $R(z)$ ,  $P(z)$  et  $g = f^{-1}$ . Pour  $\mathfrak{R}$ , d'après (2.2) et (3.2)

$$R(z) = z^{-\delta+Q} \prod_{j=1}^Q \left( 1 - \left( z_j^E \right)^{-1} z \right)^{-1}, \quad (3.3)$$

$$R(z) = z^{-\delta+Q+L} \left[ z^{-L} - \sum_{j=1}^L b^j z^{-L+j} \right]. \quad (3.4)$$

Le passage de (3.3) à (3.4) comporte un développement en série des puissances de  $z^{-1}$  sur le cercle  $U$  (grâce à la condition  $|z_j^E| < 1$ ). Ce développement est tronqué à un ordre  $L$  qui doit être d'autant plus élevé que les  $|z_j^E|$  sont proches de 1. En choisissant le retard  $\delta = Q + L$ . On voit sur (3.4) que le filtre  $\mathfrak{R}$  est un rétroprédicteur qui fabrique, sur la base du futur de  $x_{k-L}$ , une estimation  $\tilde{x}_{k-L}$  de  $x_{k-L}$  et une erreur associée  $p_k$  selon

$$\tilde{x}_{k-L} = \sum_{j=1}^L b^j x_{k-L+j}, \quad p_k = x_{k-L} - \tilde{x}_{k-L} \quad (3.5)$$

Pour  $\mathfrak{P}$ , d'après (2.1) et (3.2), la mise en oeuvre peut être récursive : elle est stable puisque  $|z_i^I| < 1$ . Pour la fonction de transfert associée

$$P(z) = \left[ 1 + \sum_{i=1}^N c^i z^{-i} \right]^{-1} \quad (3.6)$$

les  $c^i$  s'obtiennent en développant le produit (2.1). On voit que, appliqué à la sortie  $p_k$  de  $\mathfrak{R}$ , le filtre  $\mathfrak{P}$  fabrique la grandeur  $e_k$  selon

$$e_k = p_k - \hat{p}_k, \quad \hat{p}_k = \sum_{i=1}^N c^i e_{k-i}. \quad (3.7)$$

Ainsi  $e_k$  est une erreur de prédiction sur la grandeur  $p_k$  :  $\mathfrak{P}$  est un prédicteur. Finalement, après ajustement de niveau par le contrôle de gain  $\mathfrak{G}$ , la sortie de l'égaliseur est

$$y_k = g e_k \quad (3.8)$$

La structure cascade proposée pour l'égalisation est illustrée par la figure 3. Elle permet que les zéros intérieurs  $z^I$  soient très proches de  $U$ . Quant aux zéros extérieurs  $z^E$ , s'ils sont proches de  $U$ , il faudra allonger en conséquence le vecteur  $B$ . Par ailleurs le retard  $\delta = Q + L$  de restitution est fixé sans ambiguïté. Nous avons déjà proposé antérieurement [5] cette structure d'égalisation mais avec une adaptation qui ne corrigeait pas un canal mixte sévère. L'adaptation qui suit réussit dans tous les cas.

### IV- ALGORITHME AUTODIDACTE

L'algorithme présenté ici a pour objectif initial de blanchir les  $y_k$ . Ce n'est qu'ultérieurement que les niveaux  $a_k$  interviennent. Pour cela on adapte le vecteur  $H = (B^T, C^T)^T$  pour minimiser la fonction coût

$$J(H) = \alpha g^2 J_p(H) + (1 - \alpha) J_E(H), \quad \alpha \in [0,1] \quad (4.1)$$

où la composante

$$J_p(H) = E(e_k^2) \quad (4.2)$$

est minimisée lorsque la suite  $e_k$  est blanche (donc blanche aussi la sortie  $y_k = g e_k$ ) et où la composante

$$J_E(H) = E(\varepsilon_k^2) \quad (4.3)$$

est minimisée lorsque la suite  $y_k$  s'identifie avec les niveaux que peuvent prendre les données  $a_k$ . De la sorte, le critère (4.1) bascule graduellement d'un critère de blanchiment au classique critère d'égalisation autoadaptative. Il est à noter qu'une idée semblable sous-tend l'erreur (1.6) pour un égaliseur transverse [3].

Finalement le gain  $g$  de l'égaliseur est contrôlé pour que  $y_k$  ait la puissance de  $a_k$ , selon l'algorithme adaptatif

$$G_k = G_{k-1} + \lambda (y_k^2 - E(a_k^2)), \quad g_k = \sqrt{|G_k|}, \quad \lambda > 0 \quad (4.4)$$

En imposant le signe + à  $g$ , cet algorithme empêche la polarité  $s$  de restitution des données de commuter aléatoirement. Pour la minimisation de (4.1) les formules de filtrage (3.5), (3.7) et (3.8) sont respectivement mises en oeuvre avec  $B = B_{k-1}$ ,  $C = C_{k-1}$  et  $g = g_{k-1}$ , et les adaptations de  $B$  et  $C$  suivent le gradient du nouveau critère (4.1). Le gradient déterministe s'écrit pour  $B$  :



$$B_k = B_{k-1} - \frac{\nu}{2} \nabla_B J(H) \Big|_{B=B_{k-1}} \quad (4.5)$$

Le gradient stochastique, d'après (4.2) et (4.3), s'écrit:

$$B_k = B_{k-1} - \nu \left\{ \alpha g_{k-1}^2 e_k \nabla_B e_k + (1-\alpha) \varepsilon_k \nabla_B \varepsilon_k \right\} \quad (4.6)$$

Or  $e_k = \mathfrak{P}(\mathfrak{P}(x_k)) = \mathfrak{P}(\mathfrak{P}(x_k))$ , tandis que le signal

$$\zeta_k \stackrel{\Delta}{=} \mathfrak{P}(x_k) = x_k - \sum_{i=1}^N c_{k-1}^i \zeta_{k-i} \quad (4.7)$$

est indépendant du vecteur  $B$ . De plus

$$e_k = \mathfrak{P}(\zeta_k) = \zeta_{k-L} - \sum_{j=1}^L b_{k-1}^j \zeta_{k-L+j} \quad (4.8)$$

de sorte que

$$\nabla_B e_k = Z_k = -(\zeta_{k-L+1}, \zeta_{k-L+2}, \dots, \zeta_k)^T \quad (4.9)$$

L'écriture du gradient de  $\varepsilon_k$  nécessite que  $y_k$  ne soit pas exactement égal à un seuil de décision, ce qui a lieu avec probabilité 1. Ainsi, d'après (1.3),

$$\nabla_B \varepsilon_k = -\nabla_B y_k = -g_k \nabla_B e_k \quad (4.10)$$

Il découle de (4.6), (4.9) et (4.10), la règle d'adaptation

$$B_k = B_{k-1} + \nu \varepsilon^\alpha(y_k) g_k Z_k \quad (4.11)$$

avec la fonction  $\varepsilon^\alpha(y)$  ainsi définie

$$\varepsilon^\alpha(y) = \alpha y - (1-\alpha) \varepsilon(y) = y - (1-\alpha) \hat{y} \quad (4.12)$$

De même on trouve à partir du filtrage (3.7) :

$$\nabla_C e_k = -(e_{k-1}, \dots, e_{k-N})^T = -Y_k / g_k \quad (4.13)$$

$$Y_k \stackrel{\Delta}{=} (y_{k-1}, \dots, y_{k-N})^T \quad (4.14)$$

Pour le filtre récursif  $\mathfrak{P}$ , ceci n'est qu'une approximation, mais elle est justifiée [6] par la robustesse de l'algorithme adaptatif associé. Une formule analogue à (4.10) demeure; d'où la règle d'adaptation pour  $C$

$$C_k = C_{k-1} + \mu \varepsilon^\alpha(y_k) Y_k \quad (4.15)$$

Reste à faire varier le coefficient  $\alpha$  avec l'itération  $k$  pour qu'il soit proche de 1 au début, l'erreur classique  $\varepsilon(y_k)$  n'étant pas capable d'initialiser correctement l'égaliseur, et qu'il s'approche de 0 quand l'égaliseur commence d'être bon. Pour que  $\alpha$  soit dans  $[0,1]$ , nous proposons

$$\alpha = \alpha_k = (\tanh(\varepsilon_k))^2 \quad (4.16)$$

Notre algorithme utilise donc la nouvelle fonction d'erreur  $\varepsilon^\alpha(y)$ , donnée en (4.12). Comme toutes les autres erreurs autodidactes,  $\varepsilon^\alpha(y)$  est une fonction non-linéaire de  $y$ , mais sa non-linéarité passe continûment

- d'une erreur linéaire:  $\alpha=1$ ,  $\varepsilon^1(y) = y$ , qui est associée au simple critère de blanchiment/décorrélation. Les niveaux pris par les données  $\alpha_k$  ne jouent pas encore de rôle ;

- à l'erreur la plus non linéaire possible:  $\alpha = 0$ ,  $\varepsilon^0(y) = -\varepsilon(y)$ , qui est l'erreur classique d'égalisation. Les niveaux jouent leur rôle, ce qui est l'objectif final.

## V. SIMULATIONS - CONCLUSION

Les simulations présentées sur les figures 4 à 6 concernent les 3 canaux types "I", "E" et "M". Les données prennent 4 niveaux équiprobables  $\pm 1, \pm 3$ . La grandeur représentée est la puissance de l'erreur classique  $\varepsilon_k$  moyennée sur cent répétitions de l'expérience. Pour tous les algorithmes simulés et pour chaque canal les valeurs des divers paramètres (pas d'adaptation, etc) ont été optimisées.

Les simulations montrent que le nouvel égaliseur converge toujours beaucoup plus vite que les autres égaliseurs. Il a encore deux autres avantages. D'une part sa complexité arithmétique est nettement plus faible parce que la partie prédictive est réalisée sous forme récursive et peut compenser des zéros intérieurs très proches de U. Elle ne nécessite donc pas plus de prises ( $N$ ) qu'il n'y a de zéros intérieurs à U. Nous avons pris  $N=2$  ou  $=4$  et  $L+N=30$ . D'autre part  $H_k$  n'admet qu'un seul point de convergence de sorte que  $\delta$  et  $s$  (respectivement retard et polarité de restitution) sont fixes. Il n'en est pas ainsi avec les autres égaliseurs. Par exemple pour un canal mixte à 4 zéros  $\rho_I = 0.6$ ,  $\phi_I = 60^\circ$ ,  $\rho_E = 1.1$ ,  $\phi_E = 120^\circ$ , l'algorithme de Benvéniste à 31 coefficients, initialisé avec 2 séquences aléatoires, fournit successivement  $\delta = 18$ ,  $s = -1$ ;  $\delta = 16$ ,  $s = 1$ .

Pour certains canaux très sévères — non illustrés — notre égaliseur est le seul à pouvoir fonctionner ; les autres égaliseurs ne convergent pas.

## RÉFÉRENCES

- [1] Y. SATO "A method of self-recovering equalization for multilevel amplitude modulation systems", IEEE Trans. COM-23, 1975, pp. 679-682.
- [2] D.N. GODARD "Self-recovering equalization and carrier tracking in two dimensional data communication systems", IEEE Trans. COM-28, n° 11, 1980, pp. 1867.
- [3] A. BENVENISTE, M. GOURSAT "IEEE Trans. on Communications, vol. COM 32, 1984, pp. 871-883.
- [4] G. PICCHI, G. PRATI "Stop-and-go decision directed algorithm for joint blind equalization and carrier recovery", Proc. GLOBECOM 1986, Houston.
- [5] J.O. MACCHI, Y. GU "Déconvolution d'un canal à non-minimum de phase par prédiction", Actes du GRETSI-89, p. 367-370.
- [6] M. JAÏDANE, O. MACCHI "Quasic-periodic selfstabilization of adaptive ARMA predictors" International J. on Adaptive Control and Signal Processing, vol. 2, pp. 1-31, 1988.

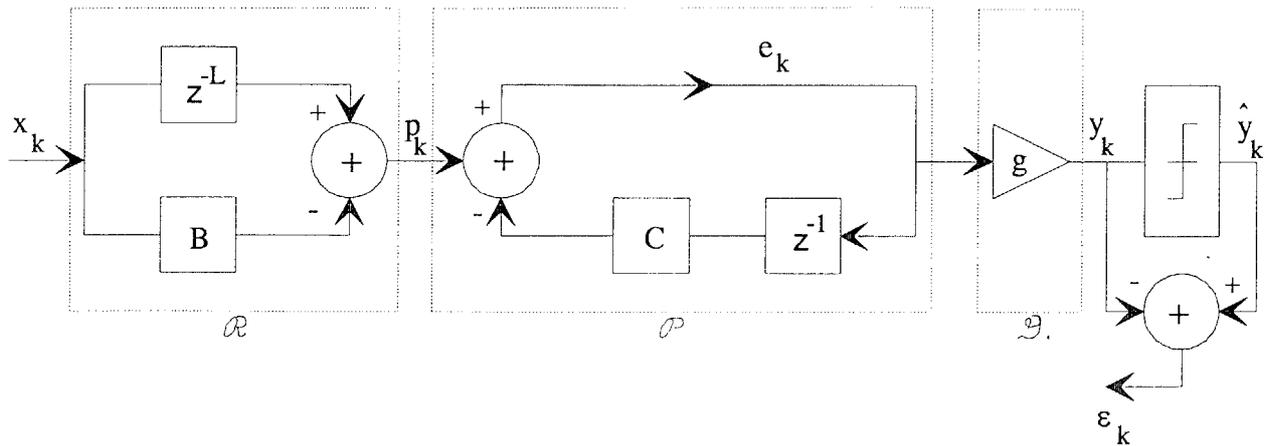


Fig.3 - Egalisation par rétroprédiction, prédiction et contrôle de gain.

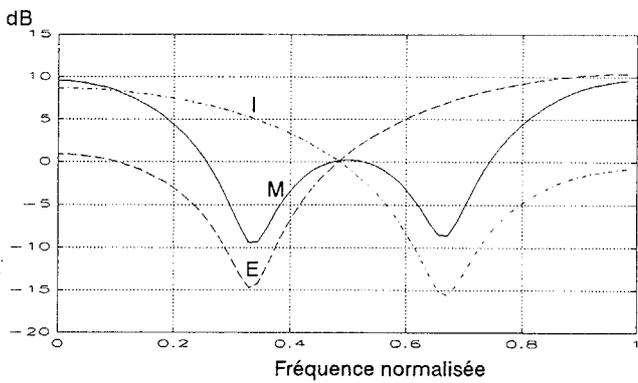


Fig. 1 - L'atténuation des 3 canaux type.

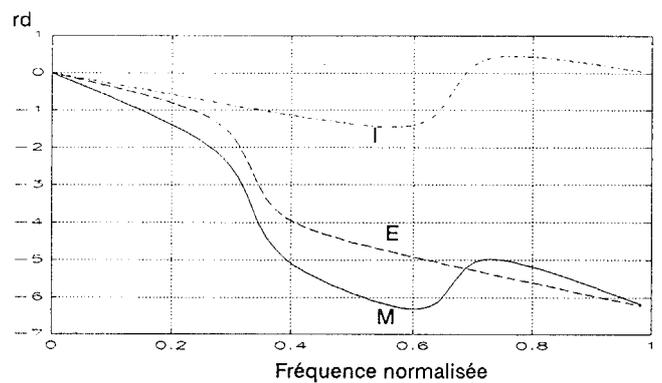


Fig.2 - Le déphasage des 3 canaux type.

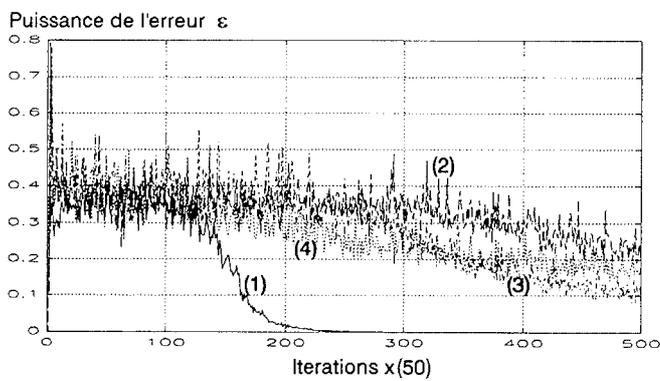


Fig.4 - Convergence des divers algorithmes pour le canal "I".

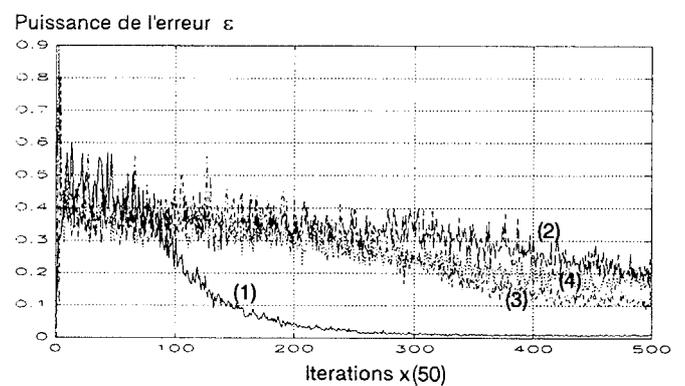


Fig.5 - Convergence des divers algorithmes pour le canal "E".

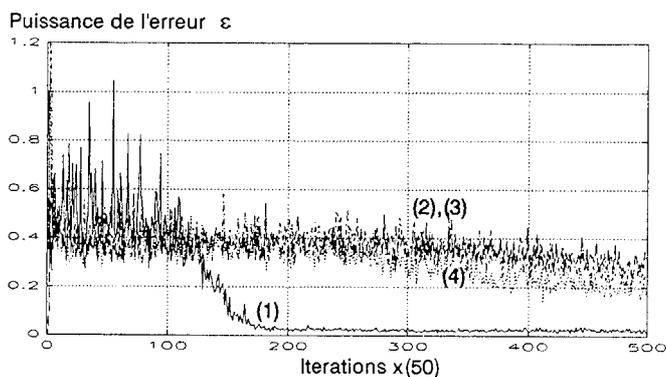


Fig.6 - Convergence des divers algorithmes pour le canal "M".

- (1) - Nouvel algorithme.
- (2) - Algorithme de Benveniste.
- (3) - Algorithme de Picchi-Prati.
- (4) - Algorithme de Godard.