

**STABILISATION DE L'ALGORITHME FTF  
MULTICANAL  
APPLICATION A LA MODELISATION ARMA**

A.AMMOUMOU(\*), D.ABOUTAJDINE(\*)  
M.NAJIM(\*\*)

(\*) LEESA, Faculté des Sciences de Rabat, BP.1014, Rabat

(\*\*) Equipe Signal et Image, ENSERB, 351, cours de la Libération, 33405 Talence Cedex

RÉSUMÉ

ABSTRACT

Dans le présent article nous présentons la version FTF stabilisé dans le cas du filtrage multicanal. On étudiera le cas où il n'y a pas supposition d'égalité de la longueur des canaux.

L'extension de la méthode de stabilisation de BENALLAL-GILLOIRE au cas multicanal restait insuffisante. En effet nous avons constaté que même stabilisé l'algorithme diverge après quelques milliers d'itérations. On propose dans cet article deux méthodes de stabilisation. Elles combinent la méthode de stabilisation de Benallal-Gilloire et les méthodes assurant la définie-positivité des matrices des énergies de prédiction directe et rétrograde.

L'algorithme FTF multicanal stabilisé sera utilisé pour la mise en oeuvre d'algorithmes pour l'identification ARMA rapide.

### I. INTRODUCTION :

Le filtrage adaptatif multidimensionnel a suscité toujours un intérêt particulier à cause des nombreux domaines d'application. La recherche d'algorithmes rapides pour le filtrage adaptatif multivoies, constitue un axe de recherche privilégié /9//14/.

Le filtrage multidimensionnel est la forme que prend le filtrage adaptatif lorsque les signaux d'entrée et/ou de référence sont des vecteurs. Un cas particulier du filtrage multidimensionnel est le cas dit multicanal. Dans ce cas nous disposons d'une seule mesure qui est fonction de plusieurs entrées (canaux). La figure (1) illustre un exemple de filtrage multicanal avec sortie scalaire.

Le filtre programmable dont la sortie est  $y(t)$  est l'ensemble des  $K$  filtres de vecteurs de coefficients respectifs  $H_i(t)$  pour  $1 < i < K$ . Ces coefficients sont estimés par minimisation de l'erreur quadratique moyenne donnée par :

$$J(t) = \sum_{p=1}^t \lambda^{t-p} e_N(t)^2$$

où: 
$$e_N(t) = y(t) - \hat{y}(t) \quad (1)$$

This paper deals with the multichannel adaptive filtering. The general case of multichannel filtering with a different number of parameters per input channel will be presented. We propose to extend the Benallal-Gilloire stabilizing method to a multichannel case. It is shown that this method exhibits a very unstable behavior due to the ill-conditioning of the forward prediction error covariance. We propose two modifications of the original algorithm. The first one is based on the modification of the up-date equation of some variables in the algorithm. The last one uses the LDU decomposition which corresponds geometrically to a modified Gram-Schmidt orthogonalization of multichannel error vectors. Finally, we apply these algorithms to derive a fast ARMA identification procedure.

Avec

$$\hat{y}(t) = \sum_{i=1}^K \hat{y}_i(t) \quad \text{Et} \quad \hat{y}_i(t) = -H_i^T(t) \cdot X_i(t) \quad (2)$$

$$(N_i \text{ longueur du filtre } H_i(t)), \quad N = \sum_{i=1}^K N_i \quad (2')$$

$$\text{Et:} \quad X_i(t) = [x_i(t) \ x_i(t-1) \ \dots \ x_i(t-N_i+1)] \quad (2'')$$

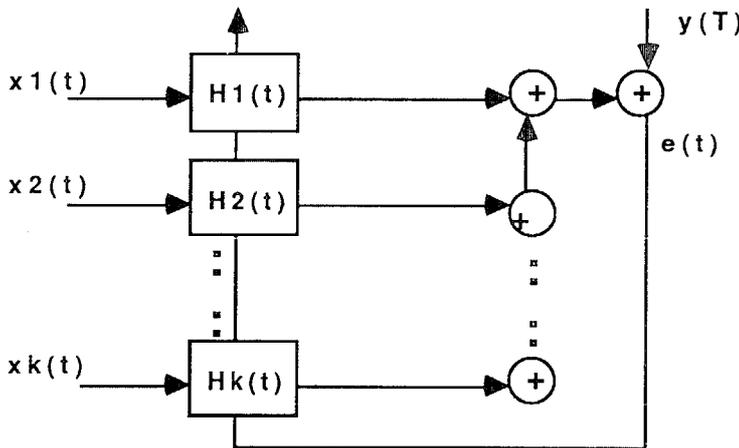
Pour aboutir à des versions rapides, on a toujours supposé que les  $N_i$  étaient égaux. Ceci permet d'obtenir facilement les algorithmes multicanaux à partir des versions monodimensionnelles. Toutefois dans certaines applications telles que l'égalisation dans la transmission multivoies /10/(sous marines par exemple), supposer que les différentes voies sont de même longueur est une condition très contraignante qui conduit à une limitation des performances.

On présentera dans cet article la version FTF stabilisée pour le filtrage multicanal dans le cas général où on ne suppose pas l'égalité de la longueur des mémoires dans chaque voie. Ce problème a été abordé dans /9//10/. Ling et Proakis /9/ ont présenté



un algorithme en treillis basé sur un traitement séquentiel des données d'entrée. leur algorithme a été appliqué pour l'égalisation de canal /14/.

Des réalisations transversales ont été présentées plus tard /11/13/16/17/. Elles sont basées sur l'introduction des permutations successives pour la réorganisation du vecteur d'observation. C'est cette dernière technique que nous allons utiliser dans ce papier. On a constaté que tous ces algorithmes (à part ceux en treillis) présentent des instabilités numériques comme dans le cas monodimensionnel.



L'article est construit comme suit: dans la section 2 nous présentons l'algorithme FTF multicanal (MFTF), dans la section 3 on analysera son instabilité numérique en proposant deux méthodes de stabilisation; enfin dans la section 4 on appliquera l'algorithme (MFTF) pour la mise en oeuvre d'un algorithme de modélisation ARMA rapide.

**II. ALGORITHME FTF MULTICANAL:**

Pour aboutir à une version rapide de l'algorithme transversal, on peut exploiter la propriété de décalage dans le vecteur d'observation. Le vecteur d'observation doit être réorganisé de telle sorte qu'à chaque itération on soit capable d'exploiter cette propriété.

On définit alors le vecteur d'observation global par:

$$\bar{X}_N(t) = [X_1^T(t) X_2^T(t) \dots X_K^T(t)]^T \quad (3)$$

Le vecteur des nouvelles données est donné par :

$$X(t) = [x_1(t) x_2(t) \dots x_K(t)]^T \quad (4)$$

On définit aussi deux permutations T et S par:

$$(i) T \bar{X}_{N+K}(t+1) = [X^T(t+1) \bar{X}_N^T(t)]^T = \tilde{X}_{N+K}^{(1)}(t+1)$$

$$(ii) S \bar{X}_{N+K}(t+1) = [\bar{X}_N^T(t+1) \hat{X}^T(t-N)]^T = \tilde{X}_{N+K}^{(2)}(t+1)$$

Avec:

$$\hat{X}^T(t-N) = [x_1(t-N+1) \dots x_K(t-N+1)] \quad (5)$$

Avec ces définitions, la matrice d'autocorrélation est exprimée par la relation:

$$R_{N+K}(t) = T^T \tilde{R}_{N+K}^{(1)}(t) T = S^T \tilde{R}_{N+K}^{(2)}(t) S \quad (6)$$

$$Et: \tilde{R}_{N+K}^{(\cdot)}(t) = \sum_{p=0}^L \lambda^{t-p} \tilde{X}_{N+K}^{(\cdot)}(p) \tilde{X}_{N+K}^{(\cdot)T}(p)$$

D'où:

$$R_{N+K}^{-1}(t) = T^T [\tilde{R}_{N+K}^{(1)}(t)]^{-1} T = S^T [\tilde{R}_{N+K}^{(2)}(t)]^{-1} S \quad (7)$$

En utilisant le lemme d'inversion d'une matrice par bloc, on obtient aisément les différentes équations de l'algorithme. qui est résumé dans le tableau 1.

**III. ANALYSE DES PERFORMANCES DE L'ALGORITHME FTF MULTICANAL:**

Bien que l'analyse des performances englobe l'étude de plusieurs caractéristiques souhaitables lors de l'utilisation des algorithmes adaptatifs, on s'intéressera ici surtout à l'analyse de la stabilité numérique de l'algorithme FTF multicanal. Pour la stabilisation de cet algorithme D. T. M Slock avait proposé une version stabilisée /16//13/. Ici on propose une extension de la méthode de stabilisation de Benallal /5/ au cas multicanal, on notera surtout que des modifications s'avèrent nécessaires au niveau du calcul de  $\gamma_N(t)$  et  $\alpha_N^f(t)$ .

**a/Extension de la méthode de stabilisation au cas multicanal:**

On définit la variable critère de divergence  $\xi_N(t)$  comme étant la différence des deux expressions de l'erreur de prédiction retour a priori données par:

$$e_N^{b(1)}(t) = \hat{X}(t-N) + B_N^T(t-1) \bar{X}(t) \quad (8)$$

$$e_N^{b(2)}(t) = -\lambda \alpha_N^b(t-1) \tilde{\delta}_{N+K}^{(N+K)}(t) \quad (9)$$

où  $\tilde{\delta}_{N+K}^{(N+K)}(t)$  sont les K dernières composantes du gain de KALMAN.

$\xi_N(t)$  est un vecteur à K composantes, et la méthode de stabilisation consiste à l'introduire dans le calcul de l'erreur de prédiction rétrograde:

$$\xi_N(t) = e_N^{b(1)}(t) - e_N^{b(2)}(t) \quad (10)$$

$$e_N^b(t) = e_N^{b(1)}(t) - \xi_N(t) \quad (11)$$

La figure (2) représente  $\gamma_N(t)$  obtenue dans le cas d'un système à trois canaux , on remarque le mauvais comportement de la variable de vraisemblance avec la version de Benallal/Gilloire. En analysant les différentes équations de l'algorithme, on remarque que la seule variable susceptible d'être à l'origine de cette nouvelle divergence, est l'énergie de prédiction directe  $\alpha_N^f(t)$  (en particulier le calcul de son inverse  $\alpha_N^{-f}(t)$ ). En effet,  $\alpha_N^f(t)$  est une matrice théoriquement symétrique définie positive, qui, à cause des erreurs numériques, est susceptible d'être mal conditionnée (perte de symétrie ou/et de définie-positivité); On a remarqué lors des nombreuses simulations qu'on a

effectuées que  $\alpha_N^{-f}(t)$  commence par perdre d'abord sa symétrie, ensuite sa positivité, et l'algorithme diverge au bout de quelques milliers d'itérations. Pour remédier à ce problème on propose deux méthodes de stabilisation.

#### b/Première méthode de stabilisation:

La mise à jour du gain utilise la variable inverse de  $\alpha_N^f(t)$ , le calcul direct par inversion matricielle en plus de son coût en nombre d'opération (surtout pour  $K$  grand), est la source d'autres problèmes d'instabilité numérique.

L'équation de mise à jour de  $\alpha_N^f(t)$  et l'utilisation du lemme d'inversion matricielle permettent un calcul récursif de  $\alpha_N^{-f}(t)$  suivant:

$$\alpha_N^f(t) = \lambda^{-1} \cdot \alpha_N^f(t-1) - \tilde{W}_{N+K}^{(1)}(t) \cdot \tilde{W}_{N+K}^{(1)T}(t) \cdot \gamma_{N+K}(t) \quad (12)$$

On propose aussi de propager  $\gamma_N(t)$  suivant:

$$\gamma_{N+K}^1(t) = \gamma_N^1(t-1) - \tilde{W}_{N+K}^{(1)T}(t) \cdot e_N^f(t) \quad (13)$$

$$\gamma_N^1(t) = \gamma_{N+K}^1(t) + \tilde{\delta}_{N+K}^{(N+K)T}(t) \cdot e_N^b(t) \quad (14)$$

où  $\tilde{W}_{N+K}^{(1)}(t)$  sont les  $K$  premières composantes du gain de Kalman.

#### c/ Deuxième méthode de stabilisation:

Partant du fait qu'on est en présence d'un mauvais conditionnement de la matrice  $\alpha_N^{-f}(t)$ , on propose le recours aux algorithmes de factorisation, qui sont en fait destinés à garantir la symétrie et la défini-positivité de la matrice de covariance dans l'algorithme RLS /15/. Leur introduction ici provient du fait que l'équation de mise à jour de  $\alpha_N^f(t)$  et  $\alpha_N^b(t)$  est équivalente à l'équation de Riccati dans l'algorithme RLS. D'après /15/, la factorisation LDU par le procédé de Gram-Schmidt modifié est très robuste vis à vis des erreurs numériques. De plus les éléments de la diagonale  $D$  constituent un bon indicateur de l'effet des multicollinéarités des canaux. La mise en oeuvre de la factorisation LDU est réalisée de deux manières; d'abord de façon algébrique par mise à jour des facteurs  $U$  et  $D$  de  $\alpha_N^{-f}(t)$  ou de manière géométrique par le procédé de Gram-Schmidt modifié du vecteur des erreurs de prédiction; opération qui correspond à la décomposition:

$$P_{(XY)} = P_X + P_{[P_X]^\perp} \cdot Y_1 + P_{[P_{(XY_1)}]^\perp} \cdot Y_2 + \dots + P_{[P_{(XY_1, \dots, Y_{K-1})}]^\perp} \cdot Y_K \quad (15)$$

Notons que cette décomposition est réalisée de façon implicite par utilisation de l'algorithme FTF multicanal séquentiel /16/17/. La version stabilisée de cet algorithme est présentée dans le tableau 2.

La figure 3 illustre le bon comportement de  $\gamma_N(t)$  dans le cas d'un système à trois canaux (même exemple que précédemment), par l'utilisation des deux algorithmes stabilisés.

#### IV. ALGORITHMES RAPIDES POUR LA MODELISATION POLE-ZERO (ARMA)

Bien que le problème général de la modélisation ARMA ait été abondamment étudié et appliqué surtout en automatique, le filtrage ARMA par un algorithme rapide n'a fait l'objet que de peu de travaux: Ardalan et Faber /12/ et E. Karlson /11/. La modélisation ARMA rapide a été surtout appliquée en annulation d'écho /2/, elle permet d'avoir les mêmes performances que celles obtenues avec une simple modélisation MA, mais avec beaucoup moins de coefficients.

On étudiera ici les performances des deux réalisations les plus utilisées parallèle et série/parallèle.

##### a. Modélisation ARMA rapide avec l'algorithme FTF multicanal stabilisé:

Un filtre adaptatif avec pôle-zéro peut être considéré comme un filtre à deux (ou trois) entrées et à une seule sortie /19//20/.

Alors, avec une réalisation de type série parallèle, le filtre pôle zéro, peut être considéré comme un filtre à deux entrées (2D):

$$\hat{x}(t) = [ x(t) \quad y(t-1) ] \quad (16)$$

Cependant, pour réduire le biais on utilise souvent la réalisation parallèle avec lissage de l'erreur d'estimation, ce qui correspond à un filtre à trois entrées (3D):

$$\hat{x}(t) = [ x(t) \quad \hat{y}(t-1) \quad e_N(t-1) ] \quad (17)$$

Dans la figure (4) est illustré le RSB en sortie d'un filtre pôle/zéro avec bruit en sortie (RSB=30).

Le filtre pôle/zéro à identifier est:

$$H(z) = \frac{0.05 + 0.1z^{-1} + 0.075z^{-2}}{1 - 0.96z^{-1} + 0.94z^{-2}}$$

On note alors l'intérêt d'une réalisation parallèle.

#### CONCLUSION:

Dans cet article nous avons présenté deux modifications de l'algorithme FTF multicanal. Elles ont été utilisées par la suite pour la mise en oeuvre d'algorithmes rapides pour la modélisation ARMA. On a noté pour cette application l'intérêt de la réalisation parallèle par rapport à la réalisation série parallèle.

#### REFERENCES

- /1/ J.M.CIOFFI and T.KAILATH, " Fast RLS transversal filters for adaptive filtering", IEEE TRANS ON ASSP-32, NO.2, APR 1984.
- /2/ J.L.BOTTO, " Etude des algorithmes transversaux rapides: application pour l'audio conférence", Thèse IRISA Rennes 1986.
- /3/ L.LJUNG, M.MORF and D.FALCONER, " Fast calculation of matrices for recursive estimation schemes", INT.J. Control, VOL.27, NO.1, JAN.1978.



- /4/ D.MANOLAKIS, F.LING and J.G.PROAKIS, " Multichannel lattice -lader structures with applications to pole-zero modeling", Proc.Int.Symp.on Circuits and Systems, Montreal, Canada, May 1984.
- /5/ A.BENALLAL and A.GILLOIRE, " A new method to stabilize fast RLS algorithms based on first order model of propagation of numerical errors", Proc.of ICASSP'88, New York.
- /6/ D.T.M.SLOCK and T.KAILATH, " Numerically stable fast RLS transversal filters", Proc. of ICASSP'88, NEW YORK.
- /7/ G.O.GLENTIS and N.KALOUPSIDIS, " Efficient order recursive algorithms for multichannel LS filtering and system identification and block adaptive structures, " Proc.of ICASSP'90.
- /8/ S.KARABOYAS and N.kalouptsidis, " Efficient adaptive algorithms for ARMA identification", Accepted in IEEE trans.on ASSP.
- /9/ F.LING, and J.G.PROAKIS, " A generalized multichannel least squares lattice algorithm based on sequential processing stages, " IEEE. Trans.on ASSP, Vol 32, pp. 381-389, april 1984.
- /10/ F.LING, and J.G.PROAKIS, " Adaptive lattice decision feedback equalizers.Their performance and application to time-variant multipath channels, " IEEE Trans. on Comm., vol.COM-33, pp. 348-356, 1985.
- /11/ E.KARLSSON, and M.H.HAYER, " Least squares ARMA modeling of linear time-varying systems: Lattice filter structures and fast RLS algorithms, " IEEE Trans. on ASSP, Vol. ASSP.35, pp. 994 -1014, July 1987.
- /12/ S.H.ARDALAN and L.J.FABER, " A fast ARMA transversal RLS filter algorithm, " IEEE Trans.on ASSP, Vol.36, pp.349-358.
- /13/ S.T.SLOCK and T.KAILATH, " Modular and numerically stable multichannel FTF algorithms, " Proc.of ICASSP'89, pp.1039-1042.
- /14/ S.KARABOYAS, N.KALOUPSIDIS, and C.CAROUBALOS, " Highly parallel multichannel LS algorithms and application to decision-feedback equalizers", ASSP-Vol.37, NO 9, Sep.89.
- /15/ G.FAVIER, " Filtrage, Modélisation et Identification des systèmes linéaires stochastiques à temps discrets", Edition du CNRS, 1981.
- /16/ D.T.M.SLOCK and T.KAILATH, " Multichannel fast transversal filter algorithms for adaptive broadband beamforming" SPIE vol.1152, Advanced algorithms and signal processing IV (1989).
- /17/ G.GLENTIS, N.KALOUPSIDIS, " Fast adaptive algorithms for multichannel phase LS filtering", Proceedings of EUSIPCO'90, pp.401-404.
- /18/ S.THEODORIDIS, G.MOUSTAKIDES, " A novel class of fast adaptive algorithms for mutichannel filtering", proceedings of EUSIPCO'90, Barcelona, Spain, 1990, pp 409-411.
- /19/ A.AMMOUMOU, " Contributions à l'étude, la stabilisation et l'implantation des algorithmes rapides", Thèse de troisième cycle, LEESA, Faculté des sciences de Rabat, Mai 1991.
- /20/ M.BELLANGER, " Filtrage Adaptatif Numérique et Traitement du Signal", Edition MASSON, 1989.

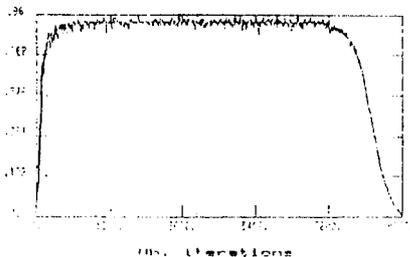


Fig 2

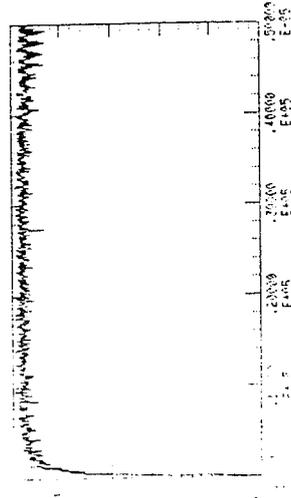


Fig 3

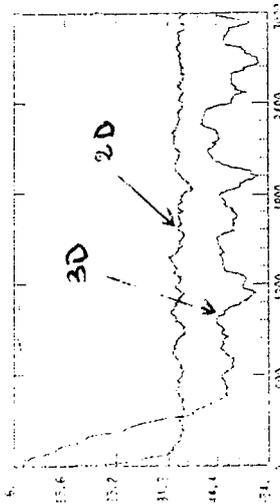


Fig 4

**TABIEAU 1**

$N \sum_{k=1}^K N_k$

Nouvelles données à l'instant t:  
 $x_1(t), x_2(t), \dots, x_p(t)$  et  $d(t)$

Calcul du gain d'adaptation:

$$e_k(t) = x_k(t) + A_{Nk}^T(t-1) \tilde{X}_{Nk}(t-1)$$

$$e_k(t) = e_k(t) - e_k(t-1) \gamma_k(t-1)$$

$$\begin{bmatrix} 1 \\ A_{Nk}(t) \end{bmatrix} = \begin{bmatrix} 1 \\ A_{Nk}(t-1) \end{bmatrix} + \begin{bmatrix} 0 \\ W_{Nk}(t-1) \end{bmatrix} \epsilon_k^T(t)$$

$$\tilde{W}_{Nk,\kappa}^{(1)}(t) = \lambda^{-1} \alpha_k^1(t-1) e_k(t)$$

$$T \tilde{W}_{Nk,\kappa}(t) = \begin{bmatrix} 0 \\ W_{Nk}(t-1) \end{bmatrix} + \begin{bmatrix} I_K \\ A_{Nk}(t-1) \end{bmatrix} \tilde{W}_{Nk,\kappa}^{(1)}(t)$$

$$\tilde{W}_{Nk,\kappa}^{(1)}(t) = \lambda^{-1} \alpha_k^1(t-1) e_k(t)$$

$$T \tilde{W}_{Nk,\kappa}(t) = \begin{bmatrix} 0 \\ W_{Nk}(t-1) \end{bmatrix} + \begin{bmatrix} I_K \\ A_{Nk}(t-1) \end{bmatrix} \tilde{W}_{Nk,\kappa}^{(1)}(t)$$

$$\gamma_k(t) = \gamma_k(t-1) - \tilde{W}_{Nk,\kappa}^{(1)T}(t) e_k(t)$$

$$\alpha_k^1(t) = \lambda^{-1} \alpha_k^1(t-1) \tilde{W}_{Nk,\kappa}^{(1)T}(t) \tilde{W}_{Nk,\kappa}^{(1)}(t) \gamma_k(t)$$
  

$$S \tilde{W}_{Nk,\kappa}(t) = [\tilde{W}_{Nk,\kappa}(t) \delta_{Nk,\kappa}^{(1)T}(t)]^T$$

$$e_k^{(1)}(t) = \tilde{x}(t-N) + B_{Nk}^T(t-1) \tilde{X}_{Nk}(t)$$

$$e_k^{(2)}(t) = \lambda \alpha_k^2(t-1) \delta_{Nk,\kappa}^{(1)T}(t)$$

$$\xi_k(t) = e_k^{(1)}(t) e_k^{(2)T}(t)$$

$$e_k^2(t) = e_k^{(1)}(t) + \xi_k(t)$$

$$W_{Nk}(t) = \tilde{W}_{Nk,\kappa}(t) - B_{Nk}(t-1) \delta_{Nk,\kappa}^{(1)T}(t)$$

$$\gamma_k(t) = \gamma_k(t-1) + \delta_{Nk,\kappa}^{(1)T}(t) e_k^2(t)$$

$$\begin{bmatrix} B_{Nk}(t) \\ 1 \end{bmatrix} = \begin{bmatrix} B_{Nk}(t-1) \\ 1 \end{bmatrix} + \begin{bmatrix} W_{Nk}(t) \\ 0 \end{bmatrix} \epsilon_k^T(t)$$
  

Filtrage:

$$e_k(t) = \lambda \alpha_k(t-1) + e_k(t-1) e_k^T(t)$$

$$e_{Nk}(t) = \gamma(t) + H_{Nk}^T(t-1) \tilde{X}_{Nk}(t)$$

$$e_{Nk}(t) = \gamma_{Nk}(t) e_{Nk}(t)$$

$$H_{Nk}(t) = H_{Nk}(t-1) + W_{Nk}(t) e_{Nk}(t)$$

**TABIEAU 2**

$N \sum_{k=1}^K N_k$

Nouvelles données à l'instant t:  
 $x_1(t), x_2(t), \dots, x_p(t)$  et  $d(t)$

Calcul du gain d'adaptation:

phase 1: do i 0 to K-1

$$e_{k,i}(t) = x_{k,i}(t) + A_{Nk,i}^T(t-1) \tilde{X}_{Nk,i}^{(i)}(t)$$

$$e_{k,i}(t) = e_{k,i}(t) - e_{k,i}(t-1) \gamma_{k,i}(t)$$

$$\tilde{W}_{Nk,i,\kappa}^{(1)}(t) = \lambda^{-1} \alpha_{k,i}^1(t-1) e_{k,i}(t)$$

$$T^{(i)} \tilde{W}_{Nk,i,\kappa}(t) = \begin{bmatrix} 0 \\ W_{Nk,i}(t) \end{bmatrix} + \begin{bmatrix} 1 \\ A_{Nk,i}(t-1) \end{bmatrix} \tilde{W}_{Nk,i,\kappa}^{(1)}(t)$$

$$\alpha_{k,i}^1(t) = \lambda \alpha_{k,i}^1(t-1) + e_{k,i}(t) e_{k,i}^T(t)$$
  

$$\begin{bmatrix} 1 \\ A_{Nk,i}(t) \end{bmatrix} = \begin{bmatrix} 1 \\ A_{Nk,i}(t-1) \end{bmatrix} + \begin{bmatrix} 0 \\ W_{Nk,i}(t) \end{bmatrix} \epsilon_{k,i}^T(t)$$

$$\gamma_{k,i}(t) = \lambda \gamma_{k,i}(t-1) + e_{k,i}(t) e_{k,i}^T(t)$$

phase 2: do i K-1 to 0

$$S^{(i+1)} \tilde{W}_{Nk,i,\kappa}(t) = [\tilde{W}_{Nk,i,\kappa}(t) \delta_{Nk,i,\kappa}^{(i+1)T}(t)]^T$$

$$e_{k,i}^{(1)}(t) = x_{k,i}(t-N) + B_{Nk,i}^T(t-1) \tilde{X}_{Nk,i}^{(i)}(t)$$

$$e_{k,i}^{(2)}(t) = \lambda \alpha_{k,i}^2(t-1) \delta_{Nk,i,\kappa}^{(i+1)T}(t)$$

$$\xi_{k,i}^{(1)}(t) = e_{k,i}^{(1)}(t) e_{k,i}^{(2)T}(t)$$

$$e_{k,i}^2(t) = e_{k,i}^{(1)}(t) + \xi_{k,i}^{(1)}(t)$$

$$\tilde{W}_{Nk,i}(t) = \tilde{W}_{Nk,i,\kappa}(t) - B_{Nk,i}(t-1) \delta_{Nk,i,\kappa}^{(i+1)T}(t)$$

$$\tilde{\gamma}_{k,i}(t) = \gamma_{k,i}(t) + \delta_{Nk,i,\kappa}^{(i+1)T}(t) e_{k,i}^2(t)$$

$$\tilde{W}_{Nk,i}(t) = \frac{(1 + \tilde{\gamma}_{k,i}(t)) e_{k,i}^2(t) \delta_{Nk,i,\kappa}^{(i+1)T}(t)}{\alpha_{k,i}^2(t) - e_{k,i}^2(t) \gamma_{k,i}(t)}$$

$$\begin{bmatrix} B_{Nk,i}(t) \\ 1 \end{bmatrix} = \begin{bmatrix} B_{Nk,i}(t-1) \\ 1 \end{bmatrix} + \begin{bmatrix} \tilde{W}_{Nk,i}(t) \\ 0 \end{bmatrix} \epsilon_{k,i}^T(t)$$
  

Filtrage:

$$e_{k,i}^2(t) = \lambda \alpha_{k,i}^2(t-1) + e_{k,i}^2(t-1) e_{k,i}^2(t)$$

$$e_{Nk}(t) = \gamma(t) + H_{Nk}^T(t-1) \tilde{X}_{Nk}(t)$$

$$e_{Nk}(t) = \gamma_{Nk}(t) e_{Nk}(t)$$

$$H_{Nk}(t) = H_{Nk}(t-1) + W_{Nk}(t) e_{Nk}(t)$$