

## Un logiciel d'aide à la conception des filtres adaptatifs sur PC

### *A PC Software for Adaptive Filter Design*

---

#### **Hassane MIMOUN**



ISEP (Institut Supérieur d'Electronique de Paris), 21, rue d'Assas, 75006 Paris

Ingénieur diplômé de l'Ecole Nationale Polytechnique d'Alger (1985), a obtenu le diplôme de D.E.A. en Automatique et Traitement du Signal à l'Université de Paris VII en 1987. Il prépare actuellement une thèse de doctorat au sein de l'ISEP où il occupe un poste d'enseignant. Domaines de recherche : traitement numérique du signal, filtrage adaptatif.

#### **Michel CIAZYNSKI**



ISEP (Institut Supérieur d'Electronique de Paris), 21, rue d'Assas, 75006 Paris

Ingénieur diplômé de l'Ecole Supérieure d'Electricité (1976), responsable du laboratoire Traitement du Signal et Télécommunications de l'ISEP. Centres d'intérêts : traitement numérique du signal et d'images, télévision.

#### **Maurice BELLANGER**



TRT-Philips, 5, av. Réaumur, 92350 Plessis-Robinson

Ingénieur diplômé de l'ENST (1965) et docteur ès sciences (1981, Université de Paris XI, Orsay), Maurice Bellanger est, depuis 1988, le directeur scientifique de la société TRT-Philips. De 1984 à 1987 il a été professeur associé à l'Université de Paris XI, Orsay, en automatique et traitement du signal. De 1967 à 1984, ses travaux ont porté, au sein de la société TRT, sur les systèmes de transmission pour le réseau de télécommunications.

M. Bellanger est le président de l'EURASIP (European Association for Signal Processing).

---

### RÉSUMÉ

Le logiciel présenté comporte trois modules correspondant à trois grands domaines du filtrage adaptatif : les filtrages monodimensionnels transversaux ou en treillis pour lesquels la mise à jour des coefficients se fait à l'aide d'un algorithme du gradient, ceux pour lesquels elle se fait à l'aide d'un algorithme des moindres carrés et les filtres dont l'entrée est un signal multidimensionnel.

Pour chaque module, la procédure d'utilisation est basée sur une suite de questions-réponses. Les résultats de la simulation sont ensuite visualisés à l'aide d'utilitaires, par exemple du type de ceux qui sont fournis par

l'environnement logiciel du système MONARCH.

L'utilisateur peut ainsi, avec les moyens informatiques les plus simples et un apprentissage rudimentaire, aborder les problèmes essentiels du filtrage adaptatif.

#### MOTS CLÉS

Filtres numériques, Filtres Adaptatifs, logiciel de conception, simulation, algorithmes adaptatifs.

---

### SUMMARY

The software presented consists of three different modules, associated with three major areas in adaptive filtering and their applications. These areas are first the filters for one-dimensional signals using the transversal and lattice structures and the gradient (Least Mean Squares-LMS) algorithm, then, the same filters with the Fast Least Squares (FLS) algorithms and finally the transversal filters for multidimensional inputs.

In each case, the process begins with a question-answer session in which the system and simulation parameters are determined. Then, the results can be visualized, with a set of tools, for example like those provided by the PC

software system MONARCH.

With the software described, a user equipped with basic computer means can, after a simple training, experiment adaptive filters and get acquainted with their most important features.

#### KEY WORDS

Digital Filters, Adaptive Filters, design software, simulation, adaptive algorithms.

## FICHE DE PRÉSENTATION DU LOGICIEL

titre	: Aide à la Conception des Filtres Numériques Adaptatifs
titre abrégé	: ADF (Adaptive Digital Filtering)
type	: programmes exécutables
langage	: C
résumé	: Simulation interactive des filtres adaptatifs en structure transversale et en treillis avec les algorithmes de type gradient et moindres carrés rapides, avec choix des paramètres
matériel	: IBM PC/AT, PS/2 ou 100 % compatible, poste de travail MICROVAX
mémoire vive	: minimum 512 Ko
système	: PC/MS-DOS version 2.1 et plus, UNIX pour poste de travail
graphique PC	: cartes CGA, MCGA, EGA, VGA et HGA sont valables
environnement	: MONARCH sur PC et Xwindow sur poste de travail
entrées	: fichiers de signaux sur disque
sorties	: fichiers de résultats (erreur et coefficients)
paramètres	: paramètres des filtres adaptatifs (ex. : ordre du filtre et durée de fonctionnement)
options	: choix de l'algorithme d'adaptation, structure du filtre, catégorie de filtrage, addition de bruit de mesure et type d'arithmétique
contact	: H. MIMOUN, 21, rue d'Assas 75006 Paris, tél. 49.54.52.62 - fax. 49.54.52.01
documentation	: document de présentation générale, manuel d'utilisation et un jeu d'exemples
état	: logiciel en cours de test et d'intégration sous système MONARCH et sur poste de travail MICROVAX
test	: des messages d'erreurs sont fournis à l'utilisateur en cours de session et pas de procédure de test
maintenance	: envoi des mises à jour

## 1. Introduction

Les logiciels de traitement du signal exigent en général de l'utilisateur une bonne connaissance des systèmes informatiques et un effort d'apprentissage important. La caractéristique du logiciel décrit dans le présent article est au contraire de faire appel aux systèmes les plus simples et de ne nécessiter que des connaissances rudimentaires avec un très faible effort d'apprentissage. L'enseignant, l'étudiant, l'ingénieur ou le chercheur peut ainsi directement se mettre à exploiter le logiciel, en se concentrant sur les problèmes de traitement du signal et en particulier de filtrage numérique adaptatif.

Le filtrage adaptatif est une opération de traitement du signal dont les caractéristiques évoluent dans le temps en fonction de critères définis à l'avance. Comme applications types, on peut citer la correction de canal de transmission et le pilotage automatiques ou encore l'analyse des systèmes en temps réel. Les domaines d'utilisation les plus caractéristiques sont sans doute les télécommunications et l'automatique, mais en fait on peut trouver des applications dans la plupart des secteurs de la technique.

Le principe d'un filtre numérique adaptatif est représenté à la figure 1. Un signal d'entrée  $x(n)$  est appliqué à un filtre programmable dont la sortie est soustraite d'une référence  $y(n)$  pour fournir une erreur  $e(n)$ . La variable  $n$  représente le temps, les signaux  $x(n)$  et  $y(n)$  pouvant par exemple provenir de l'échantillonnage de signaux continus avec la période  $T = 1$ . L'erreur  $e(n)$  est utilisée pour modifier les coefficients du filtre programmable en

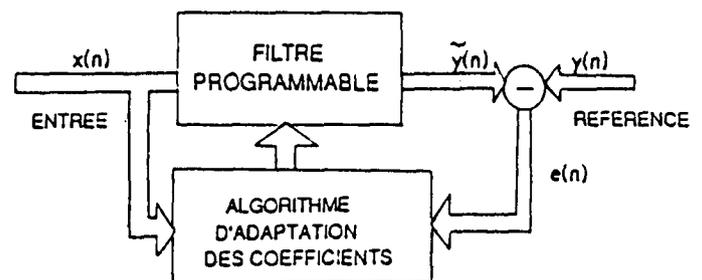


Fig. 1. — Principe d'un filtre adaptatif.

vue de minimiser la puissance de cette erreur ou sa valeur absolue, suivant le critère choisi. Dans les applications de ce principe, on distingue deux catégories suivant le choix de la référence  $y(n)$ . Si la référence est le signal d'entrée lui-même, le filtre est dit de prédiction et c'est le cas de l'analyse de signaux par exemple. Si la référence est un signal différent de l'entrée, le filtre adaptatif effectue une modélisation du système qui a produit cette référence. L'ouvrage [1] présente dans le détail les principes, les structures et les algorithmes du filtrage adaptatif, ainsi qu'un ensemble d'applications caractéristiques.

Un filtre adaptatif étant un système bouclé, l'étude de son comportement n'est pas simple et il est intéressant de disposer d'outils d'aide à l'analyse et à la conception. En particulier, un logiciel de simulation est un outil très utile, d'abord pour se familiariser avec cette technique, ensuite pour analyser les filtres adaptatifs et déterminer leurs

principaux paramètres. Cependant pour être facilement utilisable, ce logiciel doit s'intégrer dans un environnement approprié. Un tel environnement est d'abord présenté ; ensuite les options qui ont été prises sont exposées et justifiées puis, les 3 modules qui constituent ce logiciel sont décrits et enfin quelques applications typiques sont présentées.

## 2. L'environnement logiciel

Un logiciel de simulation de filtres adaptatifs a besoin d'un environnement qui lui fournisse les signaux d'entrée et de référence et auquel il puisse renvoyer ses résultats, pour présentation à l'utilisateur.

Le développement d'un environnement à la fois riche et convivial représente un effort considérable, comme le montre clairement l'observation de logiciels existants [2, 3, 4]. Pour les calculateurs de type PC, le système MONARCH [5] constitue un bon compromis entre la simplicité et la souplesse d'une part et la richesse de l'ensemble des possibilités d'autre part. Le logiciel d'aide à la conception des filtres adaptatifs présenté ci-dessous peut s'intégrer dans le système MONARCH. Il peut aussi s'intégrer dans l'environnement d'un poste de travail.

Les signaux sont contenus dans des fichiers sur disque qui sont appelés par l'utilisateur. Les signaux synthétiques peuvent être constitués à l'aide du module SIGLAB du système MONARCH qui permet de créer toute une gamme de signaux réels ou complexes à une ou plusieurs dimensions. Voici quelques exemples :

— fichier appelé « NOISE1 » contenant 1 026 échantillons d'un bruit blanc gaussien centré de puissance unité :

```
x = gn (1 026)
wsf («noise1», x) [wsf = write a signal file]
```

— fichier appelé « MA2 » contenant 1 024 échantillons de bruit après filtrage à réponse impulsionnelle finie (RIF) d'ordre 2 ; ce fichier est constitué à partir du fichier bruit précédent comme suit :

```
y = zeros (1 024)
for (i = 1 : 1 024)
y[i] = x[i + 2] + 2* x[i + 1] - 0,5* x[i]
end
wsf («MA2», y)
```

— fichier appelé « SIN1 » contenant 1 024 échantillons d'une sinusoïde d'amplitude 1,414, de fréquence 1/14 et de phase nulle :

```
z = mksin (1,414, 1/14, 0, 1 024)
wsf («SIN1», z)
```

Bien entendu il est possible d'ajouter ces fichiers les uns aux autres, de les multiplier terme à terme, d'en faire la convolution, d'en calculer la transformée de Fourier et de les visualiser.

Le module VIEW permet de visualiser les résultats de la simulation. Il est constitué de sous-modules dont le plus

utile est « 2-D », qui permet de faire apparaître les signaux et les fichiers sous la forme de courbes à l'écran. Le nombre de courbes qui peuvent apparaître séparément ou superposées est de 6 au maximum. Le choix est donné entre 3 couleurs. Le sous-module FFT calcule la transformée de Fourier du signal et fournit, pour la visualisation, 3 fichiers réels correspondant à l'amplitude, la phase et sa dérivée, le temps de propagation de groupe. Il est ainsi aisé de visualiser des signaux ou des réponses de filtres, à la fois dans le temps et dans le domaine des fréquences.

Au cours d'une simulation il peut être utile de faire certaines opérations sur des fichiers, sans sortir du système MONARCH. Le module « OS SHELL » offre un ensemble de fonctions de gestion de fichiers, lecture, impression, recopie, effacement, ainsi que l'information de capacité disponible.

Enfin, le module « DESIGN » permet de calculer les coefficients de filtres numériques de type à réponse impulsionnelle infinie (RII) ou finie (RIF), à partir de spécifications données dans le domaine temporel ou en fréquence. La réponse impulsionnelle du filtre ainsi obtenu peut ensuite être utilisée par le module « SIGLAB », par exemple pour opérer par convolution un filtrage sur un fichier signal.

Parmi les quatre modules offerts par le système MONARCH, ce sont surtout « SIGLAB » et « VIEW » qui sont utilisés par les modules de filtrage adaptatif.

## 3. Paramètres et options

L'opération de base du logiciel de filtrage adaptatif consiste à prendre des fichiers de données, entrée et référence, et à produire des fichiers de résultats. Dans la conception de ce logiciel, il faut donc commencer par définir des paramètres et prendre un ensemble d'options, d'une part pour limiter la généralité de l'approche et d'autre part pour guider l'utilisateur dans la simulation.

Les options qui sont communes aux différents algorithmes de mise à jour des coefficients sont d'abord passées en revue. Ce sont principalement :

- le nombre de coefficients
- le nombre d'itérations
- le choix entre prédiction et modélisation
- l'addition d'un bruit au signal d'entrée ou de référence
- la valeur du pas d'adaptation ou de la largeur de la fenêtre temporelle d'observation
- la visualisation des coefficients
- la trajectoire des zéros des fonctions de transfert des filtres
- la simulation à précision limitée.

Le nombre de coefficients et le nombre d'itérations, qui réagissent directement sur le nombre de fichiers à manipuler et leur longueur et déterminent la quantité d'opérations arithmétiques à effectuer, sont principalement limités par les capacités du calculateur. Les valeurs maximales de 100 et 1 024 respectivement ont été choisies pour que les temps de réponse du calculateur PC et les capacités de mémoire nécessaires restent raisonnables dans toutes les configura-

tions. Le bruit peut être ajouté à l'entrée dans le cas de la prédiction et à la référence en modélisation. Bien entendu, il faut veiller à ce que ce bruit additif soit décorrélié des signaux auxquels il s'ajoute. Son amplitude est donnée par une valeur de rapport signal à bruit, ce qui implique le calcul de la puissance du signal utile, obtenue en faisant la somme des carrés des valeurs contenues dans le fichier et en divisant par la longueur de ce fichier.

La valeur du pas d'adaptation ou du facteur de pondération  $W$ , suivant que l'algorithme est de type gradient ou moindres carrés, est un point critique puisque la stabilité du filtre et ses performances, compromis vitesse-précision, en dépendent.

Dans le cas du gradient, une borne supérieure est fournie par l'inégalité :

$$0 < \delta < \frac{2}{NP_x} \quad (1)$$

où  $N$  est le nombre de coefficients et  $P_x$  la puissance du signal d'entrée. Il est utile d'indiquer cette borne à l'utilisateur au moment où il choisit la valeur du pas  $\delta$ .

Les coefficients à visualiser doivent être rangés dans des fichiers, qui sont ensuite appelés par le module « VIEW ». Pour limiter l'encombrement en mémoire, le nombre de ces coefficients a été limité à 6, en laissant à l'utilisateur le choix de ceux qu'il souhaite observer parmi les  $N$  coefficients que compte le filtre. Pour en voir davantage, il est toujours possible de faire un second passage.

Pour avoir une vue globale du fonctionnement du filtre, il est parfois utile, notamment en prédiction linéaire, de visualiser les zéros ou les pôles des filtres dans le plan complexe, généralement dans le cercle de rayon unité. Cette option est offerte à l'utilisateur. Un fichier contenant les coordonnées des racines de la fonction de transfert en  $Z$  du filtre est créé, et appelé ensuite par le sous-module de visualisation « CIRCLE » situé dans « VIEW ».

La précision des calculs est limitée par le nombre de bits des coefficients et des données internes à la machine. Après avoir fait fonctionner le filtre sans limitation en précision, le programme enregistre la plus grande valeur absolue  $A_{\max}$  rencontrée parmi les variables internes au calcul. Dans la simulation en précision limitée, l'utilisateur donne une valeur  $b$  à la longueur totale du mot en nombre de bits et le programme refait un passage en arrondissant les coefficients et les données internes avec le pas de quantification :

$$q = A_{\max} 2^{1-b} \quad (2)$$

Une possibilité de cadrage est offerte avec le nombre de bits fractionnaire. Les fichiers de résultats avec limitation de précision portent le même nom que ceux qui sont sans limitation, la lettre  $Q$  étant ajoutée à la fin pour les différencier. La différence des erreurs est calculée dans le cas du module ADFMD, les lettres  $QE$  s'ajoutant à la fin du nom du fichier.

L'option la plus importante en filtrage adaptatif étant l'algorithme de mise à jour des coefficients, il a été décidé de créer des modules différents, pour les deux principaux

types d'algorithmes. L'algorithme du gradient est le plus utilisé et un module lui est consacré, ADFLMS (Adaptive Digital Filter with Least Mean Squares algorithm), pour le cas où le signal d'entrée est à une dimension.

Les algorithmes des moindres carrés rapides connaissent un développement important et ils peuvent présenter des avantages considérables en rapidité et précision d'adaptation. On les trouve dans le module ADFFLS (Adaptive Digital Filter with Fast Least Squares algorithm).

Un module est également consacré au cas de signaux d'entrée à plusieurs dimensions : ADFMD (Adaptive Digital Filter with Multi Dimensional input signals) ce cas se rencontre fréquemment dans les applications et notamment pour modéliser un système par filtrage RIF/RII.

La séparation en plusieurs modules se justifie, d'une part par le fait que certains paramètres fondamentaux sont différents et d'autre part par l'allègement de la procédure d'utilisation. Ces modules sont examinés successivement.

## 4. Le module ADFLMS

Une partie de la saisie des données nécessaire à la simulation est donnée à la figure 2 pour le module ADFLMS. Le choix est donné entre deux structures de filtre, transversale et en treillis et 5 variantes de l'algorithme du gradient. Dans la variante de base, en structure transversale, les équations du système sont les suivantes :

$$e(n+1) = y(n+1) - H'(n) X(n+1) \quad (3)$$

$$H(n+1) = H(n) + \delta \cdot X(n+1) e(n+1) \quad (4)$$

Le vecteur des  $N$  coefficients du filtre est désigné par  $H(n)$  et  $X(n+1)$  est le vecteur des  $N$  données d'entrée les plus récentes.

### ADAPTIVE DIGITAL FILTERS

```
Filter File : NAME.LMS
Structure : TRANSVERSAL      Plot Zeros in U.Circle : YES
Category : PREDICTION       Add Noise to In. File : YES
Algorithm : STANDARD        Fixed Point Simulation : NO
Filter Order (1..100) : 2   Number Of Iterations : 501
Input File : SIN1

Signal to Noise Ratio : 30.00000
Noise File : NOISE1
```

Fig. 2. — Procédure de saisie des données du module ADFLMS.

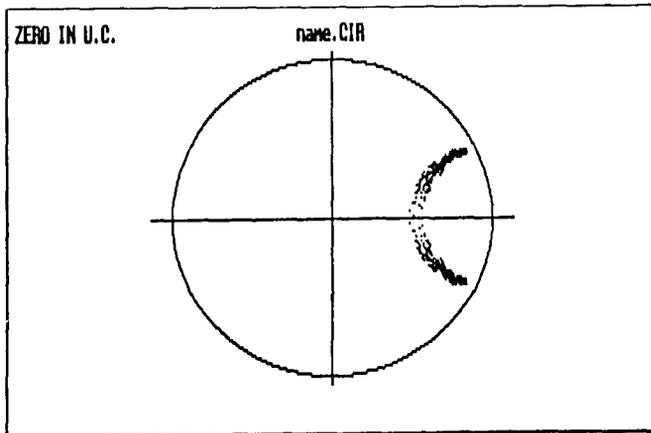
La mise à jour des coefficients est modifiée comme suit dans les autres variantes :

Version normalisée :

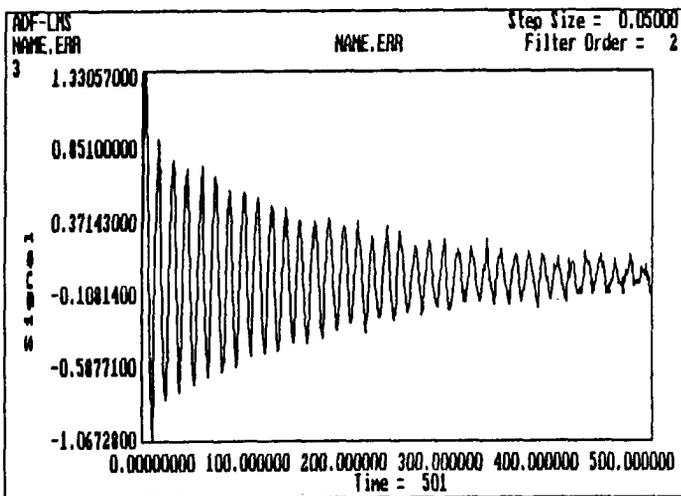
$$H(n+1) = H(n) + \frac{\delta}{0,0001 + X'(n+1) e(n+1)} X(n+1) e(n+1) \quad (5)$$

signe de l'erreur :

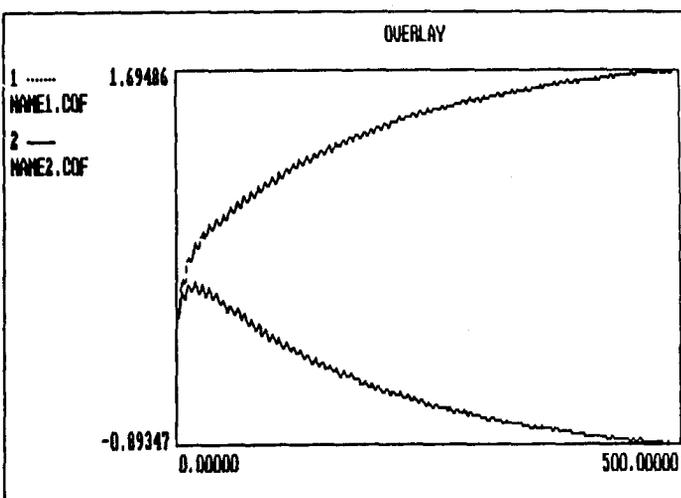
$$H(n+1) = H(n) + \delta \cdot X(n+1) \text{signe}[e(n+1)] \quad (6)$$



(a) Zéros du filtre



(b) Erreur de prédiction



(c) Coefficients du filtre

Fig. 3. — Filtre de prédiction calculé par ADFLMS.

signe des données :

$$H(n+1) = H(n) + \delta \cdot e(n+1) \text{ signe } [X(n+1)] \quad (7)$$

algorithme du signe :

$$H(n+1) = H(n) + \delta \cdot \text{signe } [e(n+1) X(n+1)]. \quad (8)$$

L'utilisateur peut ainsi comparer les résultats obtenus avec les versions simplifiées de l'algorithme de base et vérifier un ensemble de résultats théoriques [1-chapitre 4]. La possibilité de superposer les courbes d'erreur ou de coefficients à l'aide du module « VIEW » s'avère très commode pour cette comparaison.

Les résultats correspondant à l'exemple de la figure 2 avec un pas d'adaptation égal à 0,05 sont donnés à la figure 3. L'évolution des zéros de la fonction de transfert du filtre est représentée à la figure 3 a, l'erreur de prédiction à la figure 3 b et les coefficients à la figure 3 c. Les courbes peuvent apparaître une par une en écran plein, toutes sur un même écran ou superposées avec des couleurs différentes. Une fonction zoom est disponible pour l'agrandissement d'une partie quelconque de la courbe et les coordonnées d'un point marqué sur l'écran par une croix déplaçable peuvent être obtenues, accompagnées par les coordonnées du point le plus proche dans le fichier.

La structure en treillis est représentée à la figure 4. Les coefficients de la partie treillis sont mis à jour par l'équation suivante :

$$k_i(n+1) = k_i(n) + \delta \cdot [e_{a_i}(n+1) e_{b(i-1)}(n) + e_{b_i}(n+1) e_{a(i-1)}(n+1)]/2 \quad (9)$$

et ceux de la partie échelle par :

$$k_{f_i}(n+1) = k_{f_i}(n) + \delta \cdot e_i(n+1) e_{b_i}(n+1). \quad (10)$$

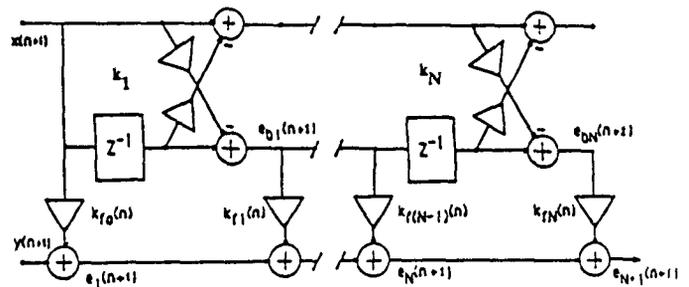


Fig. 4. — Filtre en treillis-échelle.

La borne supérieure du pas d'adaptation  $\delta$  a été prise à  $2/P_x$ , en appliquant la relation (1) pour  $N = 1$  et en considérant que les erreurs de prédiction dans les sous-ensembles sont inférieures en puissance au signal d'entrée.

A partir des coefficients  $k_i$  ( $1 \leq i \leq N$ ) du filtre de prédiction en treillis il est bien entendu possible de calculer les coefficients du filtre transversal par itération sur l'ordre, et de calculer les zéros de la fonction de transfert comme précédemment.

Pour la simulation avec précision limitée, la même approche que dans le cas transversal a été retenue. Toutefois, les coefficients du filtre prédictif en treillis étant en principe limités en valeur absolue par l'unité, la valeur maximale 2 a été retenue, pour tenir compte de possibilité de dépassement temporaire de l'unité.

Les algorithmes du gradient offrent l'avantage de la simplicité et de la robustesse au prix d'une limitation des performances. Dans certains cas, les algorithmes de moindres carrés peuvent apporter une amélioration importante en temps de convergence initiale et en erreur résiduelle après convergence.

## 5. Le module ADFFLS

La procédure de saisie des données est la même que celle donnée à la figure 2 sur un exemple. Comme précédemment le choix est offert entre les structures transversale et en treillis. Deux paramètres interviennent, le facteur de pondération et l'énergie d'erreur de prédiction initiale.

Dans les algorithmes de moindres carrés rapides, la mise à jour des coefficients se fait par l'équation suivante, en structure transversale :

$$H(n+1) = H(n) + R_N^{-1}(n+1) X(n+1) e(n+1) \quad (11)$$

où  $R_N(n+1)$  est la matrice d'autocorrélation estimée du signal d'entrée. En fait, toute manipulation de matrices est évitée en considérant le gain d'adaptation  $G(n+1)$ , défini par :

$$G(n+1) = R_N^{-1}(n+1) X(n+1) \quad (12)$$

et en le calculant à chaque itération.

Le facteur de pondération s'introduit dans la fonction coût à minimiser, donnée, à l'indice  $n$ , par l'expression :

$$J(n) = \sum_{p=1}^n W^{n-p} [y(p) - H^t(n) X(p)]^2. \quad (13)$$

Dans le cas où le signal d'entrée  $x(n)$  est assimilable à un bruit blanc, la matrice d'autocorrélation estimée, définie par :

$$R_N(n) = \sum_{p=1}^n W^{n-p} X(p) X^t(p) \quad (14)$$

peut être approchée par la matrice diagonale suivante :

$$R_N(n) \approx \frac{P_x}{1-w} I_N \quad (15)$$

c'est-à-dire qu'alors l'algorithme devient comparable au gradient avec le pas  $\delta = (1-W)/P_x$ . La valeur  $(1-W)$  peut être interprétée comme l'inverse de la fenêtre temporelle d'observation de l'algorithme.

Les conditions de stabilité imposent une borne supérieure à  $1-W$ . La valeur de cette borne dépend du spectre du signal d'entrée. Pour un bruit blanc gaussien, cette borne est donnée par :  $(1-W) < (1/(N+4(2N)^{1/2}))$  valeur retenue dans le programme. Pour les autres signaux, une marge est à prévoir.

L'énergie d'erreur de prédiction initiale,  $E_0$ , influe sur la rapidité d'adaptation à la mise en route de l'algorithme. Ce paramètre est limité par la précision des calculs et son choix dépend de la connaissance « a priori » des signaux traités. Une valeur typique pour  $E_0$  est la puissance du signal d'entrée  $P_x$ . C'est le choix retenu, sauf indication contraire de l'utilisateur.

En structure transversale, le choix est offert entre deux types d'algorithmes, MCR1 et MCR2, qui diffèrent légèrement en nombre d'opérations par itération, MCR2 étant le plus économique [1 - chapitre 6]. A noter également que MCR2 comprend un contrôle simplifié de l'accumulation des erreurs d'arrondi, ce qui le rend plus robuste.

Les évolutions des zéros de la fonction de transfert du filtre, de l'erreur en sortie du filtre et des coefficients données à la figure 5, sont obtenues avec une énergie de prédiction initiale identique à la puissance du signal d'entrée et un facteur de pondération de 0,95.

L'algorithme du treillis présente l'avantage de n'être pas sujet à l'accumulation d'erreur d'arrondi. Il fait intervenir les mêmes paramètres, facteur de pondération  $W$  et énergie d'erreur de prédiction initiale  $E_0$ . Pour faire apparaître les zéros des filtres de prédiction, il faut revenir aux coefficients transversaux, ce qui est obtenu par les équations suivantes, pour  $0 \leq i \leq N-1$

$$A_{i+1}(n+1) = \begin{bmatrix} A_i(n+1) \\ 0 \end{bmatrix} - k_{b(i+1)}(n+1) \begin{bmatrix} B_i(n+1) \\ -1 \end{bmatrix} + k_{b(i+1)}(n+1) e_{bi}(n+1) \begin{bmatrix} G_i(n+1) \\ 0 \end{bmatrix} \quad (16)$$

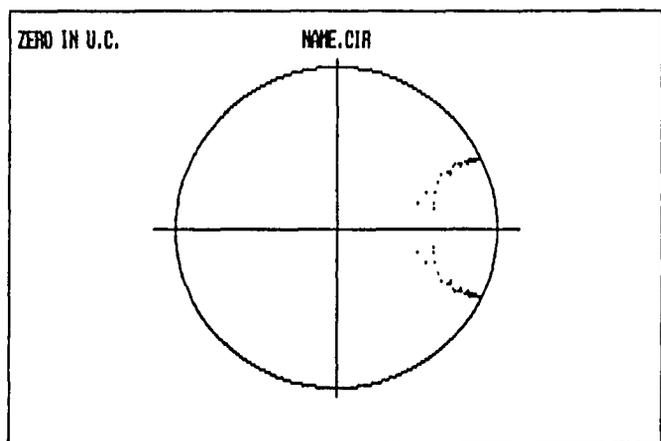
$$B_{i+1}(n+1) = \begin{bmatrix} 0 \\ B_i(n+1) \end{bmatrix} - e_{bi}(n+1) \begin{bmatrix} 0 \\ G_i(n+1) \end{bmatrix} - k_{a(i+1)}(n+1) \begin{bmatrix} -1 \\ A_i(n+1) \end{bmatrix} \quad (17)$$

$$G_{i+1}(n+1) = \begin{bmatrix} G_i(n+1) \\ 0 \end{bmatrix} + \frac{\epsilon_{bi}(n+1)}{E_{bi}(n+1)} \begin{bmatrix} -B_i(n+1) \\ 1 \end{bmatrix} \quad (18)$$

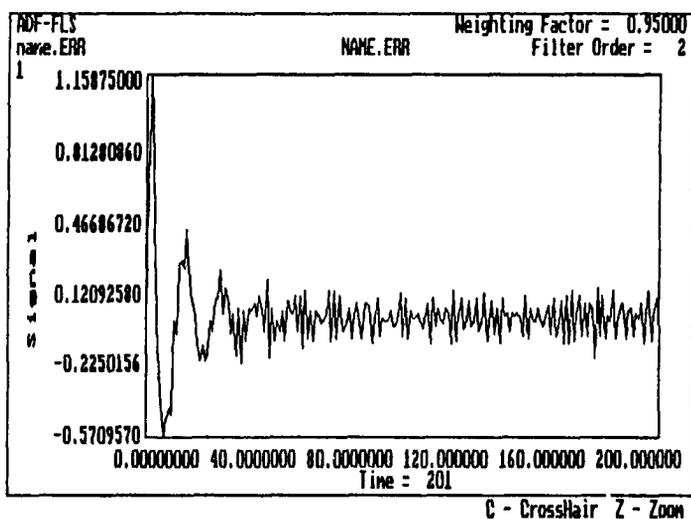
où  $A_i(n)$  et  $B_i(n)$  sont les coefficients de prédiction avant et arrière à l'ordre  $i$  [1].

Des exemples de simulation en modélisation de la structure treillis montrent clairement l'amélioration de performance apportée par l'algorithme des moindres carrés par rapport à l'algorithme du gradient.

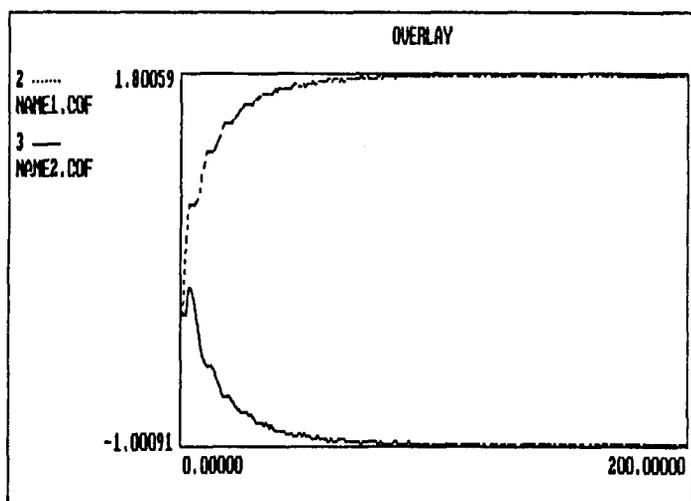
Le signal d'entrée  $x(n)$  a été supposé jusqu'à maintenant être un scalaire ; or, dans une partie des applications c'est en fait un vecteur, il est alors dit multidimensionnel. Les algorithmes peuvent s'adapter à ce cas.



(a) Zéros du filtre



(b) Erreur de prédiction



(c) Coefficients du filtre

Fig. 5. — Filtre de prédiction calculé par ADFFLS.

## 6. Le module ADFMD

Le schéma de principe du filtre adaptatif à entrée multidimensionnelle est représenté à la figure 6. Dans le cas de l'algorithme du gradient, le système correspond en fait à un ensemble de  $K$  filtres différents où les coefficients sont mis à jour par l'équation (4), dans laquelle l'erreur est commune à tous les filtres. Avec l'algorithme des moindres carrés l'optimisation est globale et faite suivant une extension de l'équation (11) comme indiqué au chapitre 7 de la référence [1].

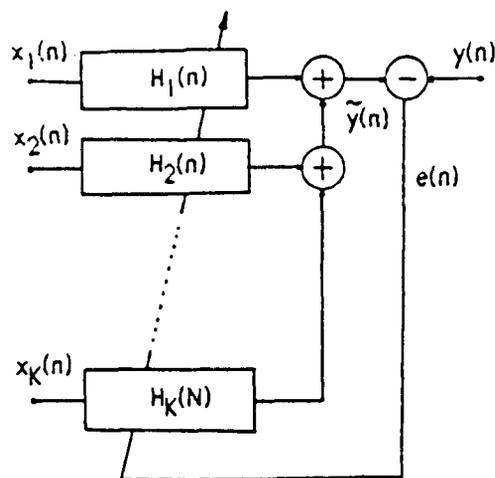


Fig. 6. — Filtre adaptatif à entrée multidimensionnelle et référence scalaire.

La figure 7 représente une partie de la saisie des données nécessaire à une simulation dans le module ADFMD. Le nombre d'entrées a été limité à 10 pour des raisons de complexité et de possibilités de visualisation. Les paramètres doivent être donnés pour chaque entrée. La borne supérieure du pas d'adaptation est calculée pour chaque signal d'entrée. Pour la mise en fichiers sur disque, les coefficients sont numérotés comme suit : les chiffres de zéro à neuf correspondent à l'entrée 1, les chiffres de 10 à 19 à l'entrée 2 et ainsi de suite. L'utilisateur choisit ensuite les coefficients qu'il souhaite visualiser dans cet ensemble. Dans ce module, seule la structure transversale a été considérée.

### MULTIDIMENSIONAL ADAPTIVE DIGITAL FILTERS

Filter File : C.MD  
 Algorithm Type : LEAST-MEAN      Fixed Point Simulation : NO  
 Number Of Input (1.10) : 2      Add Noise to Ref. File : NO  
 Number Of Iterations : 501

Ref. File : DAT2

Input Files

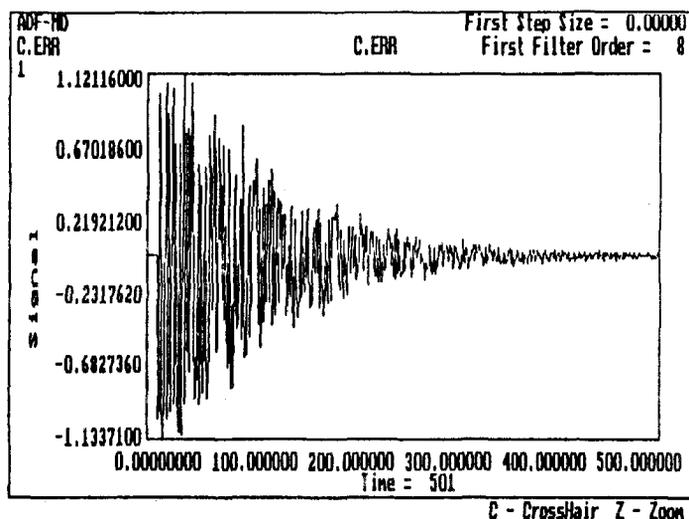
# 1 : DAT3

# 2 : DAT2

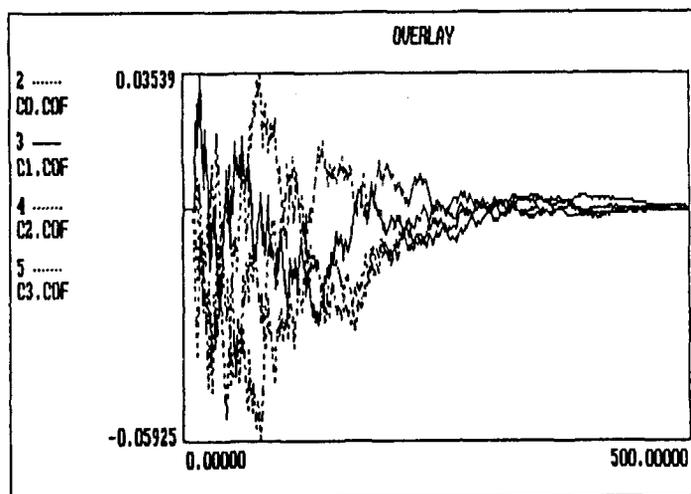
Filter Order  
 8  
 8

Filter	Max. Step	Adaptation Step
1	< 0.25560	0.01000
2	< 0.25508	0.01000
CONTINUE		

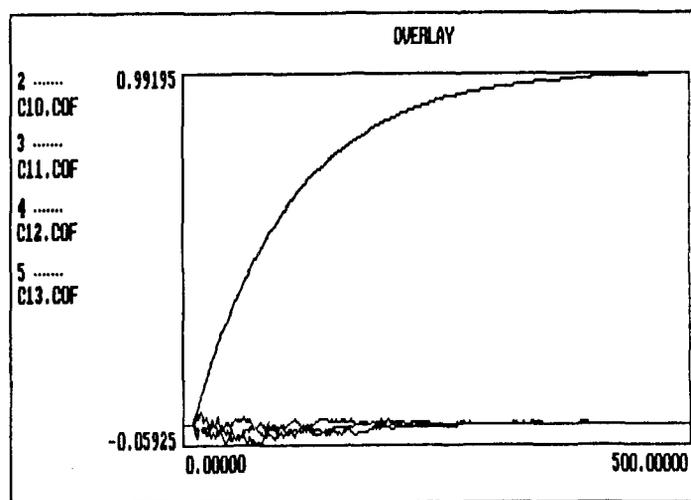
Fig. 7. — Procédure de saisie des données du module ADFMD.



(a) Erreur de sortie



(b) Coefficients de la branche 1



(c) Coefficients de la branche 2

Fig. 8. — Modélisation par filtre transversal à deux branches par ADFMD.

L'application la plus directe de ce module correspond au cas où les signaux d'entrée sont fournis par un ensemble de capteurs, comme dans les antennes adaptatives par exemple. Mais il peut aussi être utilisé en filtrage multiscadence, dans le cas de la réduction de fréquence d'échantillonnage, ainsi que pour l'identification d'un système à zéros et pôles pour la méthode dite de type série/parallèle ou de l'équation d'erreur. Dans ce dernier cas, le système a deux entrées qui sont  $x(n)$  et la référence retardée d'une période,  $y(n-1)$ . Une vérification élémentaire du fonctionnement du filtre dans ce cas est présentée à la figure 8 ; une suite aléatoire est appliquée à l'entrée 2 et à la référence, ainsi qu'à l'entrée 1 avec un retard d'un temps élémentaire ; tous les coefficients reviennent à zéro, sauf le premier de la branche 2, qui tend vers l'unité.

## 7. Conclusion

Les 3 modules qui forment le logiciel de simulation des filtres adaptatifs sur PC présenté, offrent à l'utilisateur la possibilité de se familiariser facilement avec les principales techniques de ce domaine. C'est donc tout naturellement qu'ils peuvent s'introduire dans l'enseignement, par exemple dans le cadre d'ateliers de traitement du signal. Ils peuvent être utilisés notamment pour la vérification de certains résultats théoriques et leur justification, ce qui n'est pas superflu, compte tenu de ce que ces résultats ont été obtenus parfois grâce à des hypothèses très simplificatrices.

Mais ce logiciel peut aussi être utilisé avec profit pour la conception de systèmes adaptatifs. En effet, il permet rapidement de prendre des options et de déterminer les principaux paramètres en s'appuyant sur une vérification expérimentale. Ainsi, il permet de faire la première approche dans un projet, éventuellement complété ensuite par des moyens plus puissants [6].

Le filtrage adaptatif est un domaine difficile à aborder et le logiciel qui a été présenté, conçu pour être simple à manipuler et ne nécessitant que des moyens informatiques rudimentaires, devrait contribuer à le rendre plus largement accessible et à répandre son utilisation dans tous les domaines de la technique.

## BIBLIOGRAPHIE

- [1] Maurice BELLANGER, « Analyse des Signaux et Filtrage Numérique Adaptatif », Ed. Masson Paris, 1989.
- [2] Signal Technology, Inc., « ILS (Interactive Laboratory System) Technical Manuel », Goleta, CA 93117, USA, 1983.
- [3] M. KUNT, « EAO : Atelier de Traitement Numérique des Signaux », Presses Polytechniques Romandes, Lausanne, Suisse, 1984.
- [4] F. TAYLOR and T. STOURAITIS, « Digital Filter Design Software for the IBM-PC », Marcel DEKKER, New York, 1987.
- [5] The Athena Group, Inc., « MONARCH-User's Manual, Gainesville, FL. 32605, USA, 1988.
- [6] B. MONDERER, « Computer Aided Signal Processing Tools for Research », *IEEE ASSP-Magazine*, Vol. 7, N° 2, April 1990.

Manuscrit reçu le 24 août 1989.