

Une présentation unifiée du filtrage rapide fournissant tous les intermédiaires entre traitements temporels et fréquentiels

P. Duhamel, Z.J. Mou, J. Benesty
CNET/PAB/RPE
38-40, rue du General Leclerc
ISSY les MOULINEAUX FRANCE

Résumé

Nous montrons que l'ensemble des algorithmes rapides utilisés pour accélérer les implantations des filtres non-récurrents(RIF) est composé des opérations suivantes : décomposition des signaux (soit polyphase, soit par blocs) et du filtre, "diagonalisation" de la matrice pseudocirculante ainsi obtenue, et reconstruction de la sortie à l'aide du théorème des restes chinois. Nous déduisons de cette présentation de nouvelles possibilités, et montrons que cette technique s'applique également aux filtres adaptatifs.

Summary

In this paper, we show that all the main types of fast FIR filtering algorithms can be derived as follows: decomposition of the signals and of the filter (either polyphase, or blockwisely), "diagonalisation" of the resulting pseudocirculant matrix, and reconstruction of the output by use of the Chinese Remainder Theorem. This presentation is shown to provide new possibilities. An application to adaptive filtering is also given.

1. Introduction

Historiquement, les premiers algorithmes de filtrage rapide ont suivi de très près [1] la redécouverte de la FFT par Cooley et Tukey en 1965, et utilisaient cette dernière comme intermédiaire de calcul. Peu après, ont suivi des techniques utilisant la convolution aperiodique (mono ou multidimensionnelle) [2] comme intermédiaire de calcul.

Ces deux méthodes sont caractérisées par la nécessité de travailler sur de grands blocs de données, et nécessitent des échanges globaux de variables dans ce grand vecteur. De plus, elles font intervenir explicitement (à l'extérieur du processus de calcul) des techniques de recouvrement ("overlap")

Il fallut attendre le travail de Winograd[3] pour qu'apparaissent des algorithmes rapides s'appliquant directement aux equations de filtrage. Ces techniques, basées sur la théorie de la complexité de calcul, font appel à l'imbrication, par produit tensoriel, d'algorithmes de filtrage de petite longueur à complexité de calcul réduite. Leurs inconvénients sont cependant de même nature: les techniques de recouvrement sont ici implicites, mais ces algorithmes, dans leur présentation initiale, requièrent toujours un traitement par grands blocs, et ont même le désavantage d'une structure de programmation moins régulière que les deux familles précédentes.

Ces inconvénients, inhérents aux traitements par grands blocs, ont empêché l'application du filtrage rapide pour de nombreux types d'applications (DSP, essentiellement, mais aussi VLSI). C'est ce qui a amené différents auteurs [4,5,6] à proposer indépendamment des algorithmes offrant une liberté beaucoup plus grande au niveau de la taille des blocs, tout en offrant un potentiel équivalent aux précédentes au niveau de l'amélioration de la charge de calcul.

Ces algorithmes sont très proches de ceux de Winograd, mais en diffèrent essentiellement sur les points suivants:

---possibilité de réduction de la charge de calcul dès que l'on calcule deux sorties successives d'un filtre, avec des réductions de plus en plus importantes quand la taille du bloc augmente.

---possibilité de dérivation à l'aide directement d'algorithmes de produits de polynômes, extensivement étudiés dans la littérature récente.

---localité des échanges de données.

A première vue, tous ces algorithmes semblent complètement différents. Par contre, aux yeux d'un mathématicien, ils possèdent tous la même structure. Notre objectif ici est de montrer non seulement que ces algorithmes possèdent la même structure, mais que, de plus, ce sont de simples variantes d'un même schéma général, où différents choix de paramètres fournissent l'ensemble des algorithmes connus. Puis, nous montrons que la connaissance de l'ensemble des algorithmes appartenant à cette classe permet d'obtenir un certain nombre de possibilités nouvelles, plus efficaces sous certains aspects que celles précédemment connues. Enfin, nous montrons que ces techniques sont d'application plus générale que le simple filtrage à coefficient fixe, en donnant leur application aux filtres adaptatifs (algorithme LMS).

2. Les étapes essentielles de ces algorithmes

2.1 formulation à l'aide d'une matrice pseudocirculante

Un filtre non récursif vérifie l'équation :

$$(1) \quad y_n = \sum_{i=0}^{L-1} x_{n-i} h_i \quad n = 0, 1, 2, \dots, \infty$$

2.1.1 Formulation à "décimation temporelle"

Bien que l'existence de ces algorithmes ne soit pas liée à celle de la transformée en z des signaux considérés, puisque nous verrons qu'ils s'appliquent aux filtres adaptatifs, leur présentation en est facilitée.

La transformée en z de l'eq.(1) est :

$$(2) \quad Y(z^{-1}) = H(z^{-1})X(z^{-1})$$

et, par superposition de versions sous échantillonnées des signaux en cause (sous échantillonnage de N sur X et Y, de M sur H), on obtient

$$(3) \quad \sum_{i=0}^{N-1} Y_i(z^{-N}) z^{-i} = \sum_{j=0}^{M-1} H_j(z^{-M}) z^{-j} \sum_{k=0}^{N-1} X_k(z^{-N}) z^{-k}$$

Cette equation est maintenant sous la forme d'un produit de deux polynômes, de degrés respectifs N-1 et M-1, dont tous les coefficients sont eux mêmes des polynômes. Quand N=M, eq.(3) peut se représenter par le produit d'une matrice pseudocirculante et d'un vecteur:

$$(4) \quad \begin{bmatrix} Y_{N-1} \\ Y_{N-2} \\ \vdots \\ Y_0 \end{bmatrix} = \begin{bmatrix} H_0 & H_1 & \dots & H_{N-1} \\ z^{-N}H_{N-1} & H_0 & \dots & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & H_1 \\ z^{-N}H_1 & \cdot & \cdot & z^{-N}H_{N-1}H_0 \end{bmatrix} \begin{bmatrix} X_{N-1} \\ X_{N-2} \\ \cdot \\ \cdot \\ X_0 \end{bmatrix}$$

Pour L pair, et N=M=2, par exemple, on peut aussi écrire:

$$(5) \quad \begin{bmatrix} y_n \\ y_{n-1} \end{bmatrix} = \begin{bmatrix} x_{n-L+1} \dots x_{n-3} x_{n-1} & x_{n-L+2} \dots x_{n-2} x_n \\ x_{n-L} \dots x_{n-4} x_{n-2} & x_{n-L+1} \dots x_{n-3} x_{n-1} \end{bmatrix} \begin{bmatrix} h_{L-1} \\ \cdot \\ \cdot \\ h_3 \\ h_1 \\ \cdot \\ \cdot \\ h_2 \\ h_0 \end{bmatrix}$$



qui est la restriction des polynômes infinis intervenant dans (4) au seul domaine dans lequel ils sont utilisés pour le calcul de N=2 sorties successives. En effet, une réduction de la charge de calcul ne nécessite l'invariance de H que pendant la durée d'un bloc (ici N=2), et il est donc préférable dans certains cas d'utiliser (5) plutôt que (4). De toutes façons, au prix d'un léger abus de notation, (5) peut s'écrire sous une forme pseudocirculante par blocs:

$$\begin{bmatrix} Y_1 \\ Y_0 \end{bmatrix} = \begin{bmatrix} X_0 & X_1 \\ z^{-2}X_1 & X_0 \end{bmatrix} \begin{bmatrix} H_1 \\ H_0 \end{bmatrix}$$

et on voit que tout algorithme accélérant le calcul de (4) accélérera aussi le calcul de (6), pourvu qu'il ne fasse pas appel à la commutativité des produits.

On voit que la matrice pseudocirculante (4) est obtenue par séparation des termes polyphases, comme dans une FFT à décimation temporelle, d'où le nom donné à cette section

2.1.2 Formulation à "décimation fréquentielle"

Récrivons maintenant l'équation (5) pour N=M=L=6

$$\begin{bmatrix} Y_n \\ Y_{n-1} \\ Y_{n-2} \\ Y_{n-3} \\ Y_{n-4} \\ Y_{n-5} \end{bmatrix} = \begin{bmatrix} X_{n-5} & X_{n-4} & X_{n-3} & X_{n-2} & X_{n-1} & X_n \\ X_{n-6} & X_{n-5} & X_{n-4} & X_{n-3} & X_{n-2} & X_{n-1} \\ X_{n-7} & X_{n-6} & X_{n-5} & X_{n-4} & X_{n-3} & X_{n-2} \\ X_{n-8} & X_{n-7} & X_{n-6} & X_{n-5} & X_{n-4} & X_{n-3} \\ X_{n-9} & X_{n-8} & X_{n-7} & X_{n-6} & X_{n-5} & X_{n-4} \\ X_{n-10} & X_{n-9} & X_{n-8} & X_{n-7} & X_{n-6} & X_{n-5} \end{bmatrix} \begin{bmatrix} h_5 \\ h_4 \\ h_3 \\ h_2 \\ h_1 \\ h_0 \end{bmatrix}$$

On voit aisément que, si l'on partitionne cette matrice par blocs de taille 3x3, on obtient:

$$\begin{bmatrix} Y_2 \\ Y_1 \\ Y_0 \end{bmatrix} = \begin{bmatrix} X_0 & X_1 & X_2 \\ z^{-6}X_2 & X_0 & X_1 \\ z^{-6}X_1 & z^{-6}X_2 & X_0 \end{bmatrix} \begin{bmatrix} H_2 \\ H_1 \\ H_0 \end{bmatrix}$$

ou, en partitionnant en blocs de taille 2x2:

$$\begin{bmatrix} Y_1 \\ Y_0 \end{bmatrix} = \begin{bmatrix} X_0 & X_1 \\ z^{-6}X_1 & X_0 \end{bmatrix} \begin{bmatrix} H_1 \\ H_0 \end{bmatrix}$$

et que dans tous les cas, on a une structure pseudocirculante. Cette structure existe d'ailleurs également dans l'équation (7), en remplaçant X_{n-k} par $z^{-6}X_{n-k+6}$ pour tous les éléments de la moitié inférieure gauche de la matrice

2.2 Calcul par le théorème des restes chinois

Il est bien connu, depuis le travail de Winograd sur la complexité des calculs, qu'un produit de polynômes comme (3) peut se calculer à l'aide du théorème des restes chinois (TRC). Un tel produit de deux polynômes à N et M coefficients se calcule alors avec N+M-1 multiplications. Soit donc à calculer:

$$\begin{aligned} Q(z) &= P_1(z)P_2(z) \\ \text{avec } P_1(z) &= H_0 + H_1z + \dots + H_{M-1}z^{M-1} \\ P_2(z) &= X_0 + X_1z + \dots + X_{N-1}z^{N-1} \\ Q(z) &= C_0 + C_1z + \dots + C_{N-1}z^{N-1} + \dots + C_{N+M-2}z^{N+M-2} \end{aligned}$$

dont on sait que le degré de Q(z) est N+M-2. On voit aisément que le calcul du résultat modulo un polynôme P(z) fournira le bon résultat pourvu que $\deg(P(z)) \geq N+M-1$. Choisissons un polynôme P(z) factorisable de la manière suivante :

$$P(z) = \prod_{i=1}^s p_i(z)$$

où $\{p_i(z)\}$ sont relativement premiers entre eux. Supposons d'autre part que l'on connaisse:

$$q_i(z) \equiv Q(z) \pmod{p_i(z)}$$

alors on a:

$$Q(z) \equiv \sum_{i=1}^s T_i(z)q_i(z) \pmod{P(z)}$$

avec

$$\begin{aligned} T_i(z) &\equiv 1 \pmod{p_i(z)} \\ &\equiv 0 \pmod{p_j(z)} \text{ pour } j \neq i. \end{aligned}$$

L'utilisation de ce théorème pour le calcul du produit de

polynômes s'effectue le plus souvent en choisissant:

$$p_i(z) = z - a_i; i=0,1,\dots,N+M-2 \text{ et } a_i \neq a_j \text{ si } i \neq j$$

Il est alors facile d'évaluer:

$$\begin{aligned} P_1(a_i) &\equiv P_1(z) \pmod{z - a_i} = \sum_{k=0}^{M-1} H_k a_i^k \\ P_2(a_i) &\equiv P_2(z) \pmod{z - a_i} = \sum_{j=0}^{N-1} X_j a_i^j \end{aligned}$$

donc

$$q_i(z) = Q(a_i) = P_1(a_i) P_2(a_i)$$

et de déduire

$$T_i = \prod_{\substack{j=0 \\ j \neq i}}^{N+M-2} \frac{z - a_j}{a_i - a_j} = \frac{P(z)}{P'(a_i)(z - a_i)}$$

et l'application de la formule(13) fournit Q(z):

$$Q(z) = \sum_{i=0}^{N+M-2} Q(a_i) \prod_{\substack{j=0 \\ j \neq i}}^{N+M-2} \frac{z - a_j}{a_i - a_j} = P(z) \sum_{i=0}^{N+M-2} \frac{Q(a_i)}{P'(a_i)(z - a_i)}$$

qui est similaire à la formule de l'interpolation de Lagrange.

2.3 Recouvrement

Dans le cas où N=M, on a l'égalité suivante, à partir des eq. (3) et (10):

$$Y_i = \begin{cases} C_{N-1}; & i = N - 1 \\ C_i + z^{-N}C_{i+N}; & i = 0, 1, \dots, N - 2 \end{cases}$$

qui est la reconstruction du filtre à partir du produit de polynômes. Sous cette présentation, on obtient la technique habituellement connue sous le nom de recouvrement-addition (overlap-add).

En général, quand N≠M, le recouvrement ne peut pas s'établir facilement, parce que $\{C_i\}$ sont en fonction non seulement de z^{-N} mais de z^{-M} , alors que $\{Y_i\}$ sont uniquement en fonction de z^{-N} . Une exception est le cas où M=L et les termes z^{-M} disparaissent. On a donc les recouvrements comme suivant:

$$Y_i = C_i + \sum_{k=1}^r C_{kN+i} z^{-kN}; \quad i = 0, 1, \dots, N - 1$$

Notons que $C_i = 0$ si $i > N+L-2$ et 'r' est l'entier le plus petit qui n'est inférieur à L/N.

$$Y_i = \begin{cases} C_i; & i = L - 1, L, \dots, N - 1 \\ C_i + z^{-N}C_{i+N}; & i = 0, 1, \dots, L - 2 \end{cases}$$

3. Les algorithmes connus

3.1 Les algorithmes par FFT

Choisissons un facteur de sous-échantillonnage maximal sur H (chaque sous-filtre ne comporte plus qu'un seul coefficient) M=L, et choisissons N tel que $2^n = N+M-1$. Posons de plus $\{a_k = \exp(-j2\pi k/2^n); k=0,1,\dots, 2^n-1\}$. Les equations dans (16) deviennent:

$$\begin{aligned} P_1(a_i) &= \sum_{k=0}^{L-1} h_k W^{ik} \\ P_2(a_i) &= \sum_{k=0}^{N-1} X_k W^{ik} \end{aligned}$$

qui sont clairement les FFT d'une part du filtre, d'autre part de portions du signal d'entrée. Après produit terme à terme de ces FFT dans l'eq.(17), il reste à évaluer l'eq.(18), et on peut montrer que

$$Q(z) = \sum_{i=0}^{N+L-2} Q(a_i) \sum_{j=0}^{N+L-2} \frac{1}{N} W^{-ij} z^j$$

et les coefficients de Q(z)

(25)

$$C_j = \frac{1}{N} \sum_{i=0}^{N+L-2} Q(a_i) W^{-ij}$$

qui est bien de la forme d'une FFT inverse. L'eq.(21) ou (22) constitue le recouvrement. Le graphe correspondant est montré en Fig.1. Le graphe transposé aura exactement la même fonction de transfert, et correspondra aux techniques dites "overlap-save".

3.2 Les algorithmes de Winograd [3]

Ces algorithmes utilisent l'approche à décimation fréquentielle pour la formulation en matrice pseudocirculante. Ils nécessitent donc l'attente de L échantillons avant de pouvoir réduire la charge de calcul à l'intérieur de ce bloc de taille L.

En utilisant $\{a_k\} = \{0,1,\infty\}$ pour l'application du TRC, et le partitionnement indiqué en (9), on trouve comme algorithme à charge de calcul réduite:

(26)

$$\begin{bmatrix} Y_1 \\ Y_0 \end{bmatrix} = \begin{bmatrix} X_0(H_0 + H_1) - (X_0 - X_1)H_0 \\ X_0(H_0 + H_1) + (X_1 z^{-6} - X_0)H_1 \end{bmatrix}$$

L'évaluation de la charge de calcul montre une réduction d'environ 25%. Il faut remarquer que chacun des produits dans l'eq.(26) est encore basé sur une matrice pseudocirculante, et peut donc être l'objet d'une réduction supplémentaire de la charge de calcul. Une autre remarque est que chacun de ces opérateurs, bien qu'étant une série de produits scalaires ressemblant à un filtrage sur le bloc considéré, n'est pas à proprement parler un filtre si l'on raisonne sur plusieurs blocs consécutifs, et ceci complique le séquençement des opérations.

3.3 Les algorithmes de petite longueur

Ce sont essentiellement des algorithmes du type de ceux de Winograd, où la formulation pseudocirculante est obtenue par une approche à décimation temporelle. Prenons $N=M=2$ dans (3), et $\{a_k\} = \{0,1,\infty\}$ pour le TRC, et l'on obtient l'algorithme donné en Fig.2.

Les avantages de cette structure sont nombreux : possibilité de travailler par petits blocs, de garder la structure de filtre comme partie travaillante, et localité du flot de données. Ces algorithmes sont détaillés en ref.[11].

4. Nouvelles possibilités

L'ensemble de ces techniques peut s'interpréter comme une représentation multidimensionnelle du filtrage, où un algorithme rapide est appliqué suivant une dimension, et l'algorithme classique, ou du moins une variante, suivant l'autre.

Si l'on fait varier convenablement les paramètres à notre disposition, on peut obtenir un très grand nombre de variantes des schémas connus de filtrage

Il est ainsi possible d'obtenir d'une manière très simple un algorithme demandant moins d'opérations arithmétiques que les meilleurs algorithmes classiques par FFT, quelle que soit la longueur du filtre L, au prix d'une structure à peine plus compliquée que le schéma de départ. Ces possibilités seront décrites en détail par ailleurs.

5. Filtrage adaptatif rapide

Nous proposons dans ce paragraphe d'étendre les techniques de filtrage rapide fixe au filtrage adaptatif.

On utilisera l'algorithme du gradient stochastique (LMS) [12] pour sa simplicité, associé à un filtre RIF.

5.1 Etude d'un exemple simple : N=2.

Le vecteur signal d'entrée à l'instant n est défini par :

$$X^T(n) = [x_n \ x_{n-1} \ \dots \ x_{n-L+1}]$$

où L est la longueur du filtre que l'on choisira paire.

Le vecteur poids à l'instant n est :

$$H^T(n) = [h_0 \ h_1 \ \dots \ h_{L-1}](n)$$

Le n^e signal de sortie est donné par (1).

L'algorithme LMS de Widrow et Hopf est décrit par les équations:

(27)

$$\begin{aligned} e_n &= y_n - X^T(n) H(n) \\ H(n+1) &= H(n) + \mu e_n X(n) \end{aligned}$$

où μ est un paramètre qui contrôle la stabilité et la vitesse de convergence et e_n l'erreur du système (cf. fig.3).

En écrivant ces mêmes équations à l'instant n-1 et en remplaçant l'expression de H(n) dans l'eq. (27) on obtient :

$$\begin{aligned} e_n &= y_n - X^T(n) H(n-1) - e_{n-1} s(n) \\ \text{avec } s(n) &= \mu X^T(n) X(n-1) \end{aligned}$$

Et sous forme matricielle :

(28)

$$\begin{bmatrix} e_{n-1} \\ e_n \end{bmatrix} = \begin{bmatrix} y_{n-1} \\ y_n \end{bmatrix} - \begin{bmatrix} X^T(n-1) \\ X^T(n) \end{bmatrix} H(n-1) - \begin{bmatrix} 0 & 0 \\ s(n) & 0 \end{bmatrix} \begin{bmatrix} e_{n-1} \\ e_n \end{bmatrix}$$

Le second terme de l'expression de droite correspond à un filtrage fixe, qui s'accélère à l'aide de l'algorithme donné en fig.2. On voit que le terme d'erreur (e_n, e_{n-1}) n'est pas obtenu directement à partir

des sorties de ce filtre à coefficients fixes, mais subit une correction (dûe à s(n)) destinée à imposer à l'algorithme de suivre le LMS initial. On pourrait craindre que ce terme n'augmente la charge de calcul, mais on peut remarquer que le scalaire s(n) se calcule comme suit :

(29)

$$s(n) = s(n-2) + \mu [x_{n-1}(x_n + x_{n-2}) - x_{n-L-1}(x_{n-L} + x_{n-L-2})]$$

De la même manière :

(30)

$$H(n+1) = H(n-1) + \mu [X(n-1) \ X(n)] \begin{bmatrix} e_{n-1} \\ e_n \end{bmatrix}$$

Complexité arithmétique:

μ est en général choisi comme une puissance négative de 2 et est donc négligé dans les évaluations du nombre de multiplications.

L'algorithme LMS nécessite pour deux itérations successives 4L multiplications et 4L additions, contre 3L+2 multiplications et 4L+8 additions pour l'algorithme ci-dessus. Le nombre de multiplications a donc été réduit d'environ 25 % sans aucune conséquence sur le comportement du filtre.

5.2 Généralisation

Supposons que la longueur du filtre est factorisable $L=NL_1$, et évaluons le vecteur erreur pour N sorties successives:

(31)

$$\begin{bmatrix} e_{n-N+1} \\ e_{n-N+2} \\ \vdots \\ e_{n-1} \\ e_n \end{bmatrix} = \begin{bmatrix} y_{n-N+1} \\ y_{n-N+2} \\ \vdots \\ y_{n-1} \\ y_n \end{bmatrix} - \begin{bmatrix} X^T(n-N+1) \\ X^T(n-N+2) \\ \vdots \\ X^T(n-1) \\ X^T(n) \end{bmatrix} H(n-N+1) - S(n) \begin{bmatrix} e_{n-N+1} \\ e_{n-N+2} \\ \vdots \\ e_{n-1} \\ e_n \end{bmatrix}$$

$N \times 1$ $N \times 1$ $N \times L$ $L \times 1$ $N \times N$ $N \times 1$

avec :

$$S(n) = \begin{bmatrix} 0 & 0 & \dots & \dots & \dots & 0 \\ s_1(n-N+2) & 0 & & & & \\ s_2(n-N+3) & s_1(n-N+3) & & & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ s_{N-1}(n) & s_{N-2}(n) & \dots & \dots & s_2(n) & s_1(n) & 0 \end{bmatrix}$$

où :

$$s_i(n) = \mu X^T(n) X(n-i) ; \text{ pour } i=1,2,\dots,N-1$$

Encore une fois, le terme d'erreur fait intervenir la somme d'un filtre à coefficients fixes (dont la réduction de charge de calcul se traite avec les techniques décrites en 3.3) et d'une correction nécessitant l'évaluation d'une matrice S(n). Cette évaluation fait intervenir des récurrences entre les différents termes de la matrice.

Pour une taille de bloc $N = 2^n$, les nombres d'opérations en jeu sont les suivants:

L'algorithme LMS nécessite $2L = 2 \cdot 2^n M$ multiplications et 2L additions, contre $2 \cdot (3/2)^n L/2^n + (3 \cdot 2^n - 5)/2 + 2^{-n}$ multiplications et $2 \cdot (2(3/2)^n - 1)L/2^n + 4 \cdot (3/2)^n + 2 \cdot (2^n - 3)$ additions pour l'algorithme proposé.

Le nombre de multiplications diminue comme dans le cas fixe, le nombre d'additions diminue, lui, pour une taille de bloc strictement supérieure à 2.

Une première remarque est de constater l'intérêt de pouvoir travailler avec des blocs de taille raisonnable, puisque la charge de calcul est proportionnelle à N^2 (pour un bloc de taille N). Cette approche n'est donc véritablement intéressante que couplée aux techniques de la section 3.3. Une deuxième remarque est que cette charge de calcul réduite correspond à un algorithme au comportement



inchangé par rapport au LMS. Enfin, il est possible de voir que les termes de S s'éloignant de la diagonale ont une influence de plus en plus faible sur le comportement du filtre, et qu'une approximation raisonnable est de les négliger. Ceci sera traité dans des articles ultérieurs.

6. Conclusion

Nous avons dérivé de manière homogène les différentes classes d'algorithmes de filtrage rapide, en montrant l'intérêt d'une famille récemment proposée. Après avoir cité de nouvelles possibilités apportées par cette présentation générale, nous avons montré que ces techniques s'appliquaient également au filtrage adaptatif, sans en affecter le comportement. De plus, des approximations simples devraient alléger de beaucoup la charge de calcul.

References

[1] T.G.Stockham, Jr., "High speed correlation and convolution", AFIPS Conf. Proc., vol. 28, 1966, pp. 229-233.
 [2] R.C.Agarwal, C.S.Burrus, "Fast one-dimensional digital convolution by multidimensional techniques", IEEE Trans. Acoust. Speech, Signal Processing, Vol.22, Feb. 1974, pp.1-10.
 [3] S.Winograd, "Arithmetic complexity of computations", CBMS-NSF Reginal Conf. Series in Applied Mathematics, SIAM Publications No. 33, 1980.
 [4] Z.J.Mou, P.Duhamel, "Fast FIR filtering: algorithms and implementations", Signal Processing, Dec. 1987, pp.377-384.

[5] M.Vetterli, "Running FIR and IIR filtering using multirate filter banks", IEEE Trans. Acoust. Speech, Signal Processing, Vol.36, May 1988, pp.730-738.
 [6] H.K.Kwan, M.T.Tsim, "High speed 1-D FIR digital filtering architecture using polynomial convolution", in Proc. ICASSP, Dallas, USA, pp.1863-1866.
 [7] P.P.Vandyanathan, S.K.Mitra, "Polyphase networks, block digital filtering, LPTV systems and alias-free QMF banks: A unified approach based on pseudocirculants.", IEEE Trans. Acoust. Speech, Signal Processing, Vol.36, March 1988, pp.381-391.
 [8] P.Duhamel, M.Vetterli, "Improved Fourier and Hartley transform algorithms: application to cyclic convolution of real data", IEEE Trans. Acoust. Speech, Signal Processing, Vol.35, June 1988, pp.818-824.
 [9] H.V.Sorensen, D.L.Jones, M.T.Heideman, C.S.Burrus, "Real-valued fast Fourier transform algorithms", IEEE Trans. Acoust. Speech, Signal Processing, Vol.35, June 1988, pp.849-863.
 [10] Z.J.Mou, P.Duhamel, "A unified approach to the fast FIR filtering algorithms", in Proc. ICASSP, New-York, USA, April 1988, pp.1914-1917.
 [11] P.Duhamel, Z.J.Mou, "Algorithmes de filtrage de petite longueur et leur utilisation en filtrage non-récurrent", Actes de cette conférence.
 [12] B.Widrow et al., "A comparison of adaptive algorithms based on the methods of steepest descent and random search", IEEE Trans. Antennas Propagation, Vol.24, Sept. 1976, pp.615-637.

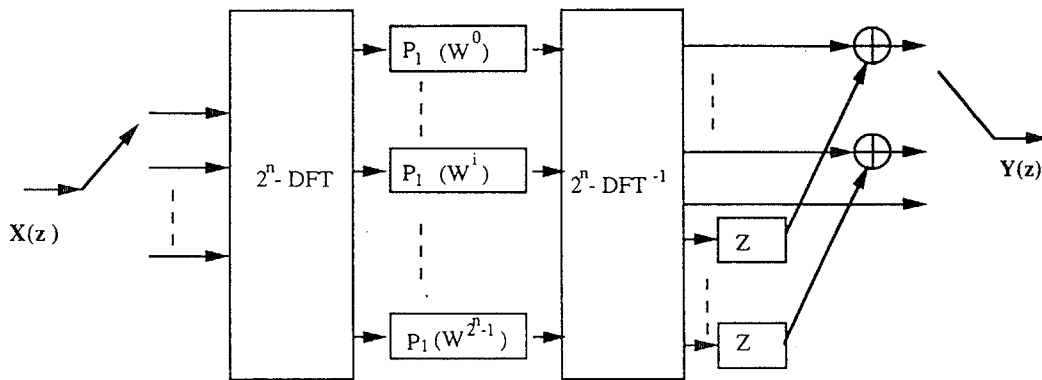


Fig.1 Filtrage RIF par FFT

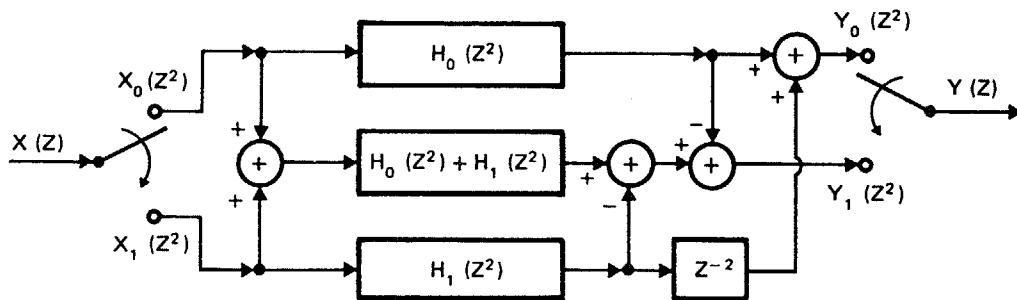


Fig.2 Filtrage de petite longueur (N=2)

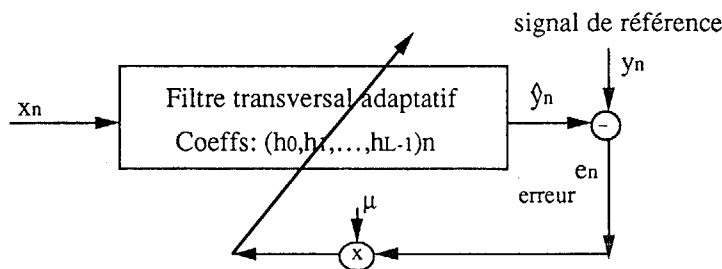


Fig.3 Filtrage adaptatif LMS.