Algorithmes rapides de filtrage non récursifs par blocs: Compromis temps d'exécution - précision numérique des calculs

A.Zergaïnoh, P.Duhamel*, J.P.Vidal

INT/SIM/TNS, 9 rue Charles Fourier 91011 Evry France
*ENST/SIG, 46 rue Barrault 75013 Paris France.
e-mail: zergaino@galaxie.int-evry.fr

RÉSUMÉ

Dans cet article, nous analysons et estimons la puissance du bruit de calcul de la nouvelle classe d'algorithmes rapides de filtrage non récursif (RIF) adaptée à une implantation, en virgule fixe, efficace en temps d'exécution sur processeur de traitement du signal. Nous étudions et évaluons en conséquence l'impact du coût de l'amélioration de la précision numérique des calculs sur le temps d'exécution évalué en nombre de cycles machine par point.

ABSTRACT

In this paper, we analyze and estimate the power of roundoff noise in fixed point of an efficient Digital Signal Processor implementation in execution time of a recent class of fast FIR filtering. We study and evaluate consequently the impact of the cost of arithmetic precision on the execution time evaluated in number of machine cycles per point.

1. Introduction

Une classe d'algorithmes rapides de filtrage à Réponse Impulsionnelle Finie (RIF) par blocs de petite taille a été proposée en [1]. Ces algorithmes de filtrage réduisent la complexité arithmétique de plus de 50% par rapport à l'algorithme de la convolution directe. Une étude de ces algorithmes de filtrage a permis de les adapter à une implantation efficace en temps d'exécution sur un processeur spécialisé en traitement du signal (DSP) (voir [2],[3]). L'adéquation algorithme architecture consiste à conserver la structure Multiplieur-Accumulateur (MAC) présente sur tous les DSP, à se contenter des ressources matérielles disponibles, sans aucune répercussion sur le nombre moyen de cycles machine nécessaires pour le traitement. Ces algorithmes de filtrage présentent des gains appréciables pour des filtres de grande longueur, ce qui convient parfaitement au filtrage adaptatif dont l'une des applications importantes est l'annulation d'écho acoustique. Nous nous intéresserons donc à l'aspect précision numérique des calculs de la partie filtrage. L'implantation en virgule fixe, des algorithmes sur DSP nécessite un codage de variables sur des mots machine de longueur limitée. Pour certaines applications il est important de se préoccuper des erreurs de quantification aussi faibles que soient elles, car elles peuvent conduire à des solutions finales inacceptables, voire même instables. Nous analyserons les effets du nombre de bits de codage, le choix du type de quantification des variables pour l'implantation des algorithmes de filtrage. Nous effectuerons un compromis entre le temps d'exécution et l'imprécision de calculs.

2. Description et organisation des algorithmes rapides de filtrage non récursifs par blocs

Les algorithmes rapides de filtrage notés $F(N,N)$ sont basés sur le principe de la décimation de la réponse impulsionnelle du filtre et des signaux d'entrée et de sortie par un facteur N. Le filtre de longueur initiale L est alors remplacé par un ensemble de sous-filtres RIF de longueur plus petite L/N disposés en parallèle et fonctionnant à une cadence réduite dans un même rapport relativement au filtre initial (voir [1]). La charge de calcul initiale exigée par le filtre de longueur L est répartie sur chacun des sous-filtres en parallèle. La technique d'imbrication des sous-filtres (voir [1]) consiste à itérer le processus sur chacun des sous-filtres de longueur L/N, de façon à obtenir des sous-filtres RIF en parallèle, de longueur très petite et à réduire la complexité arithmétique. Ces algorithmes notés $F(N_1,N_1),\ldots,F(N_n,N_n)$ traitent un bloc de données consécutives de façon à mettre à profit les redondances de calcul, réduisant en conséquence la charge de traitement. Pour assurer une implantation simple et générique à cette classe d'algorithmes de filtrage, dans le cas où plusieurs décimations successives des filtres sont opérées, une organisation générale de l'algorithme a été adoptée (voir [3]) quelque soit les facteurs de décimation N_i. L'algorithme est organisé en trois modules auxquels est attribué un traitement particulier (voir [3]). Le premier module effectue des opérations arithmétiques sur les échantillons acquis. Le deuxième module regroupe tous les sous-filtrages. Le dernier combine les produits de convolution partiels calculés par le deuxième module pour la reconstitution du signal filtré.

3. Implantation d'algorithmes de filtrage sur DSP

3.1 Contraintes architecturales

La représentation en virgule fixe par complément à 2 d'un nombre réel normalisé à 1, en base deux, consiste à réserver un bit pour le signe et b bits pour coder la partie fractionnaire. La virgule est placée à droite du bit de signe. Le nombre à coder est alors compris -1 et $1-2^{-b}$. Une opération de numérisation des signaux d'entrée en valeurs représentables est nécessaire. Cette opération inévitable est liée à la résolution du convertisseur analogique-numérique produisant une erreur de quantification d'entrée qui a fait l'objet de nombreuses discutions (voir [6,7]). Le deuxième type d'erreur est introduit par la quantification des coefficients du filtre. Cette erreur a été calculée et a fait l'objet de nombreuses études (voir [6,7]). Elle introduit des lobes secondaires parasites dans le spectre du signal traité. Le troisième type d'erreur; plus important; est l'erreur de quantification qui est liée aux structures

arithmétiques du processeur (multiplieurs, additionneurs, décaleurs) et qui dépend de l'approximation choisie: arrondi ou troncature. Dans ce qui suit nous nous intéresserons à ce dernier type d'erreur connu par bruit de calcul. Nous analyserons les effets de dégradations indésirables et inévitables sur les algorithmes rapides de filtrage non récursifs.

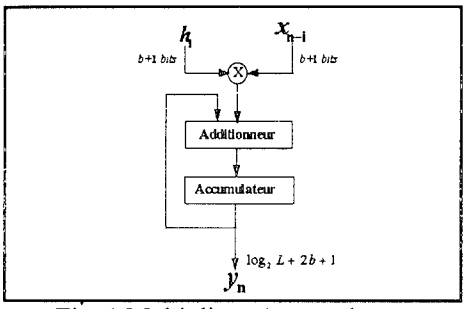

Fig. 1 Multiplieur-Accumulateur

Rappelons que l'équation de filtrage non récursif est donnée par le produit de convolution de la séquence d'entrée x_i et de la réponse impulsionnelle du filtre h_i de longueur L.

$$y_n = \sum_{i=0}^{L-1} x_{n-i} h_i \text{ avec } n = 0,1,\ldots,\infty \qquad (1)$$

L'implantation sur DSP en virgule fixe de cette équation de filtrage est exécutée par la structure MAC (voir Fig. 1). Celle ci opère en entrée sur des mots signés codés sur $b+1$ bits. La multiplication de deux nombres réels fournit un résultat codé sur $2b+1$ bits. L'accumulation de deux produits peut dans certains cas fournir un résultat ne respectant pas la dynamique autorisée par la représentation en virgule fixe, nous sommes alors confrontés au problème de dépassement de capacité. Ce problème de dynamique des variables internes est une étape importante, fastidieuse et parfois même difficile à contourner. Le nombre de bits de débordement est évalué à $\log_2 L$. Certains constructeurs de DSP prévoit des bits de débordement (voir tableau 1). Parfois ce nombre s'avère insuffisant pour des filtres de grande longueur.

Processeurs		Multiplieur (bits)	Accumulateur (bits)	Débordement (bits)
Analog Devices	ADSP-2100	16x16	40	8
Texas Instruments	TMS32010 TMSC50	16x16	32	0
Motorola	56000	24x24	56	8

Tableau 1

Les méthodes adoptées pour le codage des variables dépendent du critère de performance sélectionné:

- *temps d'exécution* privilégié si nous réduisons le nombre de bits du résultat final du produit scalaire,ou introduisons un facteur d'échelle qui limite la dynamique du signal ou des coefficients,
- *précision numérique des calculs* privilégiée si nous travaillons en pleine précision.

3.2 Implantation efficace en temps d'exécution

L'objectif à atteindre consiste à implanter le plus efficacement possible en temps d'exécution ces algorithmes de façon à ce que la réduction de la complexité arithmétique se traduise par une réduction de la charge totale en nombre de cycles machine par point. Pour la sauvegarde d'un produit de convolution partiel en mémoire ou dans un registre, nous réduisons le nombre de bits de chaque résultat final codé sur $2b+1$ bits par une opération d'arrondi pour ne garder que les $b+1$ bits de poids fort. Le troisième module (voir Fig. 2) se charge d'effectuer les opérations arithmétiques adéquates (voir [3]) sur les convolutions partielles codées sur $b+1$ bits. Des opérations de transfert de résultats s'effectueront de UAL vers la mémoire ou registre.

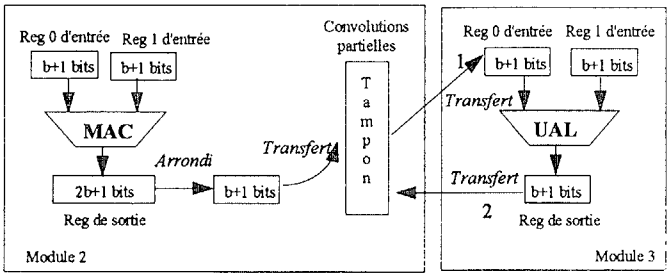

Fig. 2 Technique d'implantation efficace

3.2.1 Evaluation théorique de la puissance du bruit de calcul

L'opération d'arrondi opérée sur le résultat final du produit de convolution introduit une erreur dont la puissance du bruit de calcul donnée par l'équation (2) est indépendante de la longueur du filtre. Afin d'évaluer le bruit de calcul engendré par l'exécution des algorithmes de filtrage $F(N_1,N_1),\ldots,F(N_n,N_n)$ sur DSP analysons tout d'abord la puissance du bruit introduite par l'exécution des algorithmes de base $F(N_i,N_i)$.

$$\sigma_{\varepsilon_0}^2 = \frac{2^{-2b}}{12} = \frac{q^2}{12} \qquad (2)$$

3.2.1.1 Bruit de calcul dans $F(2,2)$

Soit l'architecture du filtre donnée par la Figure ci-dessous. Suivons l'algorithme pas à pas et déterminons la puissance de l'erreur globale. L'exécution de trois produits de convolution partiels engendre trois erreurs de calcul notées ε_0, ε_1 et ε_2.

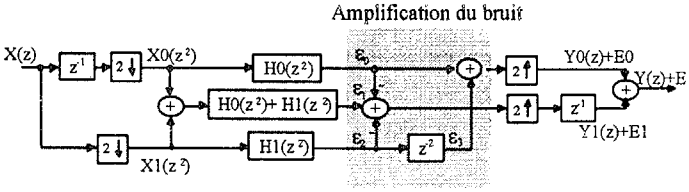

Fig. 3 Algorithme de base $F(2,2)$

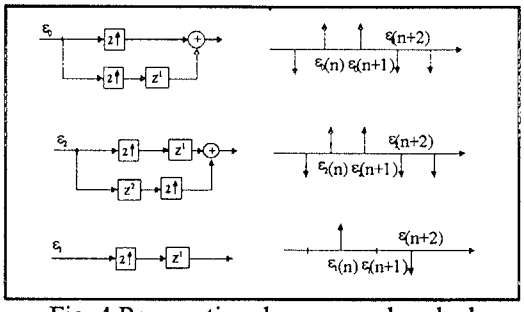

Fig. 4 Propagation des erreurs de calcul

Analysons la propagation de ces trois erreurs dans chaque branche (voir Fig. 4) et déduisons les fonctions d'autocorrélations (voir Fig. 5):

$$R_{\varepsilon_0\varepsilon_0}(0) = \mathrm{E}\left[\varepsilon_0^2(n)\right] = \sigma_{\varepsilon_0}^2 \qquad (3)$$

$$R_{\varepsilon_0\varepsilon_0}(1) = R_{\varepsilon_0\varepsilon_0}(-1) = \begin{cases} \sigma_{\varepsilon_0}^2 & \text{échantillons corrélés} \\ 0 & \text{échantillons indépendant} \end{cases}$$

$$R_{\varepsilon_0\varepsilon_0}(1) = R_{\varepsilon_0\varepsilon_0}(-1) = \mathrm{E}\left[\varepsilon_0(n)\varepsilon_0(n+1)\right] = \sigma_{\varepsilon_0}^2/2 \qquad (4)$$

$$R_{\varepsilon_0\varepsilon_0}(k) = \mathrm{E}\left[\varepsilon_0(n)\varepsilon_0(n+k)\right] = 0 \text{ pour } |k| \geq 2 \qquad (5)$$

$$R_{\varepsilon_2\varepsilon_2}(0) = \mathrm{E}\left[\varepsilon_2^2(n)\right] = \sigma^2_{\varepsilon_2} \quad (5)$$

$$R_{\varepsilon_2\varepsilon_2}(1) = R_{\varepsilon_2\varepsilon_2}(-1) = \mathrm{E}\left[\varepsilon_2(n)\varepsilon_2(n+1)\right] = \sigma^2_{\varepsilon_0}/2 \quad (6)$$

$$R_{\varepsilon_2\varepsilon_2}(k) = \mathrm{E}\left[\varepsilon_2(n)\varepsilon_2(n+k)\right] = 0 \text{ pour } |k| \geq 2 \quad (7)$$

$$R_{\varepsilon_1\varepsilon_1}(0) = \mathrm{E}\left[\varepsilon_1^2(n)\right] = \sigma^2_{\varepsilon_1}/2 \quad (8)$$

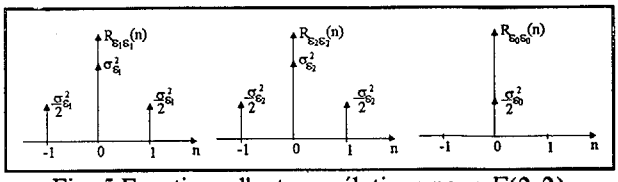

Fig. 5 Fonctions d'autocorrélations pour $F(2,2)$

Déduisons la densité spectrale:

$$\Phi_{F(2,2)}(f) = \sum_{-1}^{1}\left(R_{\varepsilon_0\varepsilon_0}(n) + R_{\varepsilon_1\varepsilon_1}(n) + R_{\varepsilon_2\varepsilon_2}(n)\right)\mathrm{e}^{-j2\pi nf}$$

$$\Phi_{F(2,2)}(f) = \frac{5}{2}\sigma^2_{\varepsilon_0} + 2\sigma^2_{\varepsilon_0}\cos 2\pi f \quad (9)$$

Notons qu'à la sortie du filtre, le bruit de calcul n'est pas à spectre plat, en conséquence il n'est pas blanc. La puissance du bruit de calcul est évaluée à:

$$\sigma^2_{F(2,2)} = \frac{5}{2}\sigma^2_{\varepsilon_0} = \frac{5}{2}\frac{q^2}{12} = F_{2,2}\frac{q^2}{12} \quad (10)$$

3.2.1.2 Bruit de calcul dans $F(3,3)$

Après avoir mentionné les erreurs de calcul sur la structure du filtre (voir Fig. 6) de l'algorithme $F(3,3)$,
nous adoptons la même méthode décrite précédemment pour l'algorithme $F(2,2)$.

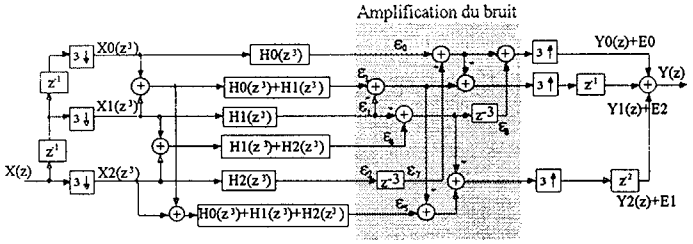

Fig. 6 Algorithme de base $F(3,3)$

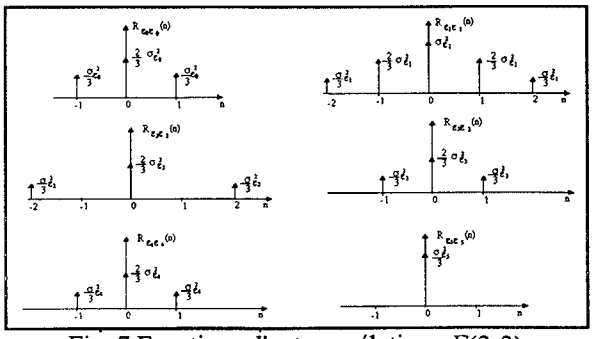

Fig. 7 Fonctions d'autocorrélations $F(3,3)$

Le calcul des fonctions d'autocorrélations (voir Fig. 7), nous permet d'évaluer la puissance du bruit de calcul:

$$\sigma^2_{F(3,3)} = 4\sigma^2_{\varepsilon_0} = 4\frac{q^2}{12} = F_{3,3}\frac{q^2}{12} \quad (11)$$

3.2.1.3 Bruit de calcul dans $F(N_1,N_1),\ldots,F(N_n,N_n)$

Le raisonnement adopté pour le calcul de la puissance du bruit des algorithmes de base $F(N_i,N_i)$, peut s'étendre aux calculs de la puissance du bruit de $F(N_1,N_1),\ldots,F(N_n,N_n)$. Nous déduisons l'expression mathématique générale de la puissance du bruit de calcul (voir [8]) notée $\sigma^2_{F(N_1,N_1),\ldots,F(N_n,N_n)}$:

$$\sigma^2_{F(N_1,N_1),\ldots,F(N_n,N_n)} = \frac{q^2}{12}\prod_{i=1}^{n} F_{N_i,N_i} \quad (12)$$

Remarquons que la puissance du bruit évolue en fonction du format de codage des données (voir tabeau 2).

Algo de filtrage	$\sigma^2_{F(N_1,N_1),\ldots,F(N_n,N_n)}$	ρ_{16bits}	ρ_{32bits}
Convo directe	$q^2/12$	91.10	187.43
$F(2,2)$	$2.5q^2/12$	87.12	183.45
$F(3,3)$	$4.33q^2/12$	84.73	181.06
$2F(2,2)$	$6.25q^2/12$	83.14	179.47
$F(2,2),F(3,3)$	$10q^2/12$	81.10	177.43
$F(3,3),F(2,2)$	$10q^2/12$	81.10	177.43
$2F(3,3)$	$16q^2/12$	79.05	175.38
$3F(2,2)$	$15.62q^2/12$	79.16	175.49
$2F(2,2),F(3,3)$	$25q^2/12$	77.12	173.45
$F(2,2),F(3,3),F(2,2)$	$25q^2/12$	77.12	173.10
$F(3,3),2F(2,2)$	$25q^2/12$	77.12	173.10
$2F(3,3),F(2,2)$	$40q^2/12$	75.08	171.40
$F(2,2),2F(3,3)$	$40q^2/12$	75.08	171.40
$F(3,3),F(2,2),F(3,3)$	$40q^2/12$	75.08	171.40
$3F(3,3)$	$64q^2/12$	73.03	169.36
$4F(2,2)$	$39.06q^2/12$	75.19	171.51
$\rho_{x\,bits} = 10\log_{10}(\sigma^2_{signal\,entrée}/\sigma^2)$: Rapport Signal sur bruit (en dB). Puissance du signal d'entrée: $\sigma^2_{signal\,entrée} = 0.1$.			

Tableau 2

3.3 Implantation efficace en précision numérique

Dans ce paragraphe, nous nous préoccupons de l'implantation des algorithmes de filtrage sur DSP en ayant pour souci la minimisation des erreurs de calcul. Toutes les opérations arithmétiques sont traitées sur $2b+1$ bits, puisque nous n'avons aucune contrainte temporelle à satisfaire. La première étape consiste à stocker les produits de convolution codés sur $2b+1$ en mémoire (voir Fig. 8).

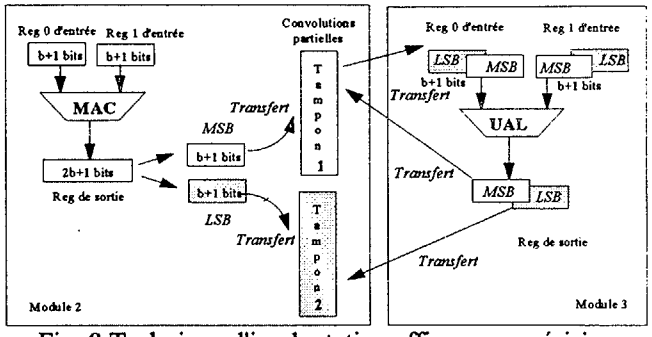

Fig. 8 Technique d'implantation efficace en précision

Nous définissons alors deux tampons: l'un attribué à la sauvegarde des bits de poids faible (LSB), et l'autre au stockage des bits de poids fort (MSB), entraînant un accroissement de l'espace mémoire. Il faut donc compter des opérations supplémentaires de transfert de données qui se traduisent par des cycles machine supplémentaires, et l'attribution d'un pointeur d'adresse additionnel pour accéder aux LSB. Le troisième module en plus des opérations arithmétiques adéquates effectuées sur les MSB, doit également traiter les LSB en tenant compte d'une éventuelle retenue. Les résultats sont ensuite transférés en mémoire. L'implantation efficace en précision numérique de ces algorithmes sur DSP consiste à ajouter des opérations arithmétiques et de transfert dans les macro-instructions établies pour une implantation efficace en temps d'exécution.

4- Compromis temps d'exécution - précision numérique

Les algorithmes de filtrage ont été implantés sur un processeur en virgule fixe de chez Analog Devices "ADSP-2100". Les unités de calcul traitent des données sur 16 bits. La structure MAC réalise les opérations de multiplication-accumulation sur 40 bits. Il est donc possible d'effectuer au minimum 256 accumulations suivies de recadrage. Les résultats de l'implantation des algorithmes sont évalués en nombre de cycles machine en fonction de la longueur du filtre fournis par le tableau 3. La troisième colonne du tableau correspond au gain maximal que peut atteindre l'algorithme de filtrage par rapport à l'algorithme de convolution directe donnée par l'équation (12), sachant que N_{convo} et N_{cp} représentent respectivement le nombre de cycles machine par point de l'algorithme de convolution classique et d'un algorithme de filtrage $F(N_1,N_1),\ldots,F(N_n,N_n)$.

$$Gain(\%) = (1 - \frac{N_{cp}}{N_{convo}}) * 100 \qquad (12)$$

Algorithmes de filtrage	Nombre de cycles machine par point		Gain limite
	Contrainte "temps d'exécution"	Contrainte "précision des calculs"	
Convolution-directe	L+9	L+10	0%
F(2,2)	15.5+3L/4	28+3L/4	25%
F(3,3)	22+2L/3	44.33+2L/3	33%
2 F(2,2)	28.25+9L/16	37.5+9L/16	44%
F(2,2),F(3,3)	38.5+L/2	73.89+L/2	50%
F(3,3),F(2,2)	39+L/2	71.33+L/2	50%
2 F(3,3)	53.11+4L/9	107.77+4L/9	56%
3 F(2,2)	47.75+27L/64	77.25+27L/64	58%
2 F(2,2),F(3,3)	63.16+3L/8	118.07+3L/8	62%
F(2,2), F(3,3), F(2,2)	64.16+3L/8	102.32+3L/8	62%
F(3,3),2 F(2,2)	65+3L/8	116.66+3L/8	62%
2 F(3,3), F(2,2)	87.27+L/3	162.71+L/3	67%
F(2,2), 2 F(3,3)	85.3+L/3	169.76+L/3	67%
F(3,3), F(2,2), F(3,3)	80.38+L/3	161.94+L/3	67%
3 F(3,3)	115+8L/27	236.07+8L/27	70%
4 F(2,2)	76.37+81L/256	105.18+81L/256	68%

Tableau 3

Les représentations graphiques (voir Fig. 9), fournissent les temps d'exécution de l'implantation des algorithmes de filtrage en fonction de la longueur du filtre RIF. Analysons l'effet de la précision numérique sur le temps d'exécution. Nous constatons que la longueur minimale (point de croisement) à partir de laquelle le temps d'exécution des algorithmes rapides est inférieur au temps d'exécution de la convolution classique est décalé par rapport au point de croisement des algorithmes ayant pour contrainte la précision des calculs.

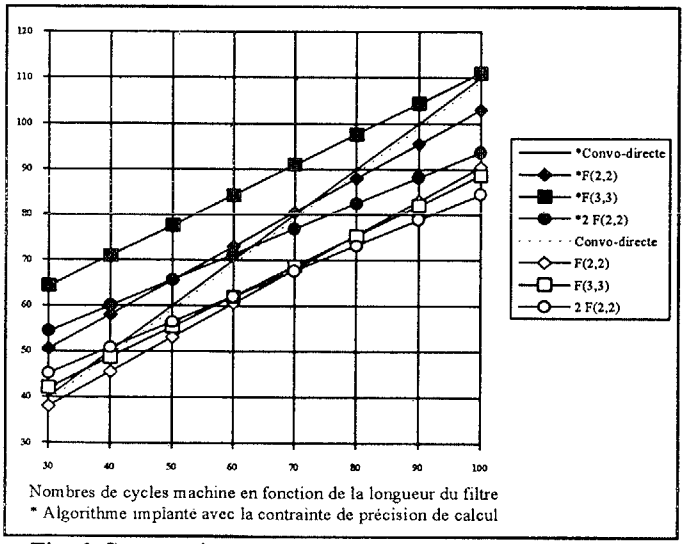

Fig. 9 Comparaison des temps d'exécution des algorithmes

Prenons un exemple: l'algorithme $F(3,3)$ présente des réductions sensibles pour des filtres de longueur supérieure à 40 coefficients, par contre $*F(3,3)$ ne peut offrir des réductions que pour des filtres de longueur supérieur à 100 coefficients. Le décalage du point de croisement s'explique par l'"overhead" qui pour certains algorithmes rapides de filtrage a doublé. Pour des filtres de même longueur, le nombre de cycles machine nécessaire pour l'implantation d'un algorithme de filtrage (précision des calcul) est sensiblement supérieur au nombre de cycles machine exigés pour l'implantation d'algorithmes rapides. Cette différence est directement liée à l'overhead. La réduction du temps d'exécution des algorithmes (précision des calculs) reste d'autant plus importante pour des filtres de grande longueur.

5. Conclusion

Après avoir analysé les différents types de limitations des DSP, nous avons déterminé les erreurs de calcul engendrées par l'exécution des algorithmes rapides de filtrage non récursifs. L'implantation de ces algorithmes sur DSP, ayant pour souci la réduction des erreurs de calculs, nous a permis d'évaluer et de comparer le temps d'exécution par rapport à celui des algorithmes rapides de filtrage. Les contraintes de précision numérique sont satisfaites au détriment du temps d'exécution, bien que ces algorithmes offrent toujours des réductions de temps d'exécution pour des filtres de grande longueur. Ce qui est donc rassurant pour les applications exigeant des contraintes sévères de précision de calcul.

Remerciements: Ce travail a bénéficié de discussions avec M. Eric Martin dans le cadre du GDR 134.

Références

[1] Z.J.Mou, P.Duhamel, "Short length FIR filters and their use in fast non recursive filtering", IEEE Trans. ASSP, 1991.

[2] A.Zergaïnoh, P.Duhamel, J.P.Vidal, "Implantation efficace d'algorithmes de filtrage rapide RIF sur ADSP-2100", Conférence Adéquation Algorithmes Architectures", Greco TDSI, AFCET, SEE, CNET, Grenoble Janvier 94.

[3] A.Zergaïnoh, P.Duhamel, J.P.Vidal, "Efficient implementation of composite length fast FIR filtering on the "ADSP-2100", IEEE proc. ICASSP Adelaïde 94, Australia.

[5] R.E.Blahut, "Fast Algorithms for Signal processing", Addison-Wesley, Reading, MA, 1985.

[6] Alan V.Oppenheim, Ronald W.Schaffer, "Digital Signal Processing", Prentice-Hall International, 1975 Editions.

[7] Ping Wah Wong, "Quantification and Roudoff Noises in fixed Point FIR Digital Filters", IEEE Trans on Signal Processing, vol. 39, no. 7, July 1991.

[8] A.Zergaïnoh, Thèse de doctorat de l'Université de Paris sud Orsay, Décembre 1994.