Signal And System Laboratory
Master Thesis

DIGITAL LINEAR PHASE FIR FILTER DESIGN IN THE DISCRETE SPACE

(CONCEPTION DES FILTRES NUMERIQUES RIF A PHASE LINEAIRE DANS L'ESPACE DISCRET)

Ahmed Nabil Belbachir and Mohamed Faouzi Belbachir

University of Science and Technology of Oran « MOHAMED BOUDIAF »

Department of Electronics
Electrical Engineering Faculty, USTO, Oran, Algeria

Oran 2000

DIGITAL LINEAR PHASE FIR FILTER DESIGN IN THE DISCRETE SPACE

Copyright © 2000 Ahmed Nabil Belbachir

Department of Electronics
Electrical Engineering Faculty,
USTO, Oran,
Algeria

Printed in Austria, Vienna 2000

Biography

Ahmed Nabil BELBACHIR was born in Oran, Algeria in 24 March 1973. He received the engineering degree from the Institute of Electronics, University of Science and Technology of Oran U.S.T.O., Oran, Algeria in 1996. He was also teacher in Computer Science at 'Science Exacte, Informatique et Tronc commun' SETI at the same University. This book represents the Master thesis for the work (DIGITAL LINEAR PHASE FIR FILTER DESIGN IN THE DISCRETE SPACE). Now, is also a PhD Student At Vienna University of Technology. He is also a author and co-author of more than 10 publications.

Abstract

Parks–Mc Clellan method allows the design of Linear Phase FIR filters. The coefficients h(n) which give the best Chebyshev approximation to the desired frequency response '$H_D(e^{jw})$' are obtained. However, this method uses an infinite precision optimisation (computer). When these filters are implemented on Digital Signal Processor with a special purpose-hardware, each filter coefficient has to be represented by a finite number of bits 'b' smaller than that used on a computer. The simplest and the most widely used approach to the problem is the rounding of the optimal infinite precision coefficients to its 'b' bits representation. However, the filters obtained are degraded and in most case there exists another set of finite word length coefficients which gives the best Chebyshev approximation to the desired frequency response '$H_D(e^{jw})$'. To find these coefficients, it is necessary to include the finite word length restriction into the filter design. The main object of this work is to perform strategies which give the best performances to the literature algorithms. Several approaches are proposed. The effect of the binary representation choice is studied. An iterated approach to LSD (DMC) is performed. And, a new filter design method, sequential and progressive is also proposed. The results obtained are very promising.

Résumé

Ce travail a permis de mettre en évidence les problèmes liés au filtrage numérique et spécialement ceux concernant la mise en œuvre des coefficients du filtre sur un processeur de signaux de longueur de mot finie. A ce propos, des approches permettant d'évaluer la méthode adéquate de l'optimisation de la mise en œuvre des filtres sur un processeur sont proposées dans ce travail.

Dans ce contexte, les effets de la quantification des coefficients, du choix de la représentation binaire, du choix de critère d'évaluation et du choix de la méthode de synthèse de filtre sur la qualité de sortie (précision et temps de calcul) du filtre ont été étudiés. Il a été montré que les méthodes existantes dans la littérature scientifique concernant la synthèse de filtres, telles que celles de Parks-McClellan avec quantification et celles du laboratoire Signaux et Systèmes (méthode RA et méthode DMC) donnent soit un temps de calcul prohibitif, soit des résultats peu performants. Le but de ce travail est de mettre en œuvre trois approches permettant d'améliorer la qualité de sortie des filtres mis en œuvre. Le choix adéquat de la représentation binaire, la méthode itérative (DMCI) et une méthode séquentielle (RSP). Tels les résultats montrent, les algorithmes proposés sont performants.

© 2000 Ahmed Nabil Belbachir, ALL RIGHTS RESERVED

Printed in Austria, Vienna 2000

© 2000 Ahmed Nabil Belbachir, ALL RIGHTS RESERVED

DEDICACES

REMERCIMENTS

Remercier M. Mohamed Faouzi Belbachir, mon directeur de recherche, en quelques lignes n'est pas suffisant, ni à la hauteur des efforts qu'il a fait pour l'accomplissement de ce travail, ni pour mettre en évidence sa valeur prodigieuse. J'aurais préféré de rédiger le volume de ce mémoire gratitude pour M. Belbachir, mais la réalité m'oblige à être plus pragmatique. La similitude du nom, n'a influé en aucun cas ces paroles, ce sont des propos du fond du cœur et de l'âme profonde. Je remercie M. Belbachir pour ces conseils qui étaient tous très importants pour le bon déroulement de ce travail, pour son amabilité et sympathie durant cette période de mémoire. Non seulement que M. Belbachir soit un homme de science, au vrai sens de terme, mais aussi soit un homme chaleureux, humain, aphorique et euphorique. Si j'avais à revenir en arrière ' voyager dans le temps' et à rechoisir un encadreur je rechoisirai sûrement M. Belbachir. M. Belbachir n'était pas seulement un encadreur, mais aussi un ami, un frère et un père à la fois. A travers ce mémoire je dis à M. belbachir Merci, Merci beaucoup.

Je tiens aussi à remercier M. Boulerial pour son aide grandiose dans l'accomplissement de ce travail. L'idée de ce thème provient de lui, et grâce à lui notre laboratoire a pu attaquer un sujet de recherche très important, qui est la synthèse de filtres numériques. Je remercie M. Boulerial pour sa contribution scientifique et administrative dans la réalisation de ce travail. Si j'avais à ajouter un nom sur la couverture de la thèse, j'ajouterai son nom. Merci Beaucoup M. Boulerial.

Oh Nawal ! Je ne sais pas si je dois te remercier, ou tirer un chapeau ou quoi faire. Vraiment, ton soutien moral et physique, ton aide et ta contribution a fait que ce travail pourrait ne pas être accompli à temps. Les larmes de l'imprimante sur ces humbles lignes ne pourrait satisfaire le volcan de mot qui frotte mes lèvres, par conséquent, je cède à la technologie.

Je tiens aussi à remercier Hafida qui m'a soutenu le long de mon travail. Sa contribution morale m'a aidé à présenter ce mémoire à temps et à avoir espoir dans des moments de pessimisme. Grâce à ces efforts, j'ai confirmé que nos régions se connaissent, et nos coutumes se ressemblent. Merci beaucoup Hafida.

Je ne peux oublier la Hadjira, la fameuse, la grandiose, et la célèbre par deux communications internationales. Non la modeste Hadjira, qui a vraiment su me maintenir actif pour accomplir ma thèse, parfois par ignorance intelligente, et parfois par paroles aphoriques et consistantes. Je ne te remercie pas, mais je me baisse en reconnaissance de tes actes.

Je remercie aussi tous les membres du laboratoires Signaux et Systèmes pour leur contribution morale dans ce mémoire, Hakim le magnifique, Sid Ahmed l'extraordinaire, Wissam le fantastique et le binôme ingénieur Nacéra la splendide et Djamila la ravissante.

Je remercie aussi tous les membres de l'association 'Baheth' avec Bouchra (pas Rekia), Imane, Mustapha (Courtois et élégant et fond très bon), Della et Safia les superbes et les autres. Merci pour votre soutien. Vous êtes très bien restez comme vous êtes.

Je tiens aussi à remercier Prof. Michèle Marchesi et Alessandra Fanni - Italie pour l'intérêt qu'ils ont portés à mon travail et pour leur conseils très important pour son l'accomplissement parfait. Merci Beaucoup aussi pour les étudiants chercheurs du département de l'Electronique de L'université de Cagliari pour leur précieux soutien.

Je tiens aussi à remercier le Prof. Lars Wanhammer de Suède, Prof. Ioannis Pitas de Grèce, Prof. Josep Sala Alvarez d'Espagne et les Prof. Wolfgang Mecklenbräucker, et Gehrard Döblinger d'Autriche pour la mise en valeur précieuse de mon travail, chose qui m'a incité et encouragé à apprécier ce thème et à mieux le traiter. Merci Beaucoup.

Je tiens aussi à remercier les membres de Jury de mon Magister et l'attention qu'ils ont prêté pour ce sujet. Je remercie M. Belaidi le président, M. Benyettou et M. Djebbari les examinateurs pour leur effort dans la lecture de ce mémoire.

Au fruit de ce travail je me retrouve avec un seul diplôme mais plusieurs frères, sœurs et amis d'une qualité exceptionnelle. J'espère que je n'ai oublié personne.

Merci à tous

Ahmed Nabil BELBACHIR

08/12/1999

SOMMAIRE

Chapitre II.

PERFORMANCES DES ALGORITHMES RA ET DMC EN FONCTION DE LA REPRESENTATION ET DU CRITERE CHOISI

Chapitre III.

METHODE DIRECTE PAR MOINDRE CARRE ITERATIVE 'D.M.C.I.'

Chapitre IV.
METHODE DE RECHERCHE SEQUENTIELLE ET PROGRESSIVE 'R.S.P.'

INTRODUCTION

L'évolution de la technologie électronique, l'intégration à grande échelle de l'outil informatique, amènent une diminution du coût des matériels de traitement numérique de l'information, alors que leur puissance de traitement s'accroît ; il en résulte une augmentation du nombre des applications dans lesquelles intervient le calculateur. On conçoit aisément qu'on puisse utiliser la souplesse du calcul numérique pour réaliser sur ordinateurs munis de cartes d'acquisition - restitution et d'interfaces adaptées (processeurs de signaux), des lois de commande, effectuer des pretraitements sur les données. Ce processus de pretraitement nommé 'Filtrage' est placé après un dispositif de conversion analogique – numérique.
Il existe deux sortes de filtres numériques : les filtres à Réponse Impulsionnelle Finie (R.I.F.) et les filtres à Réponse Impulsionnelle Infinie (R.I.I.). dans notre travail, nous nous intéresserons aux filtres RIF à phase linéaire caractérisés par la réponse en fréquence $H(e^{jw})$ donnée sous la forme :

$$H(e^{jw}) = \sum_{n=0}^{N-1} h(n)\ e^{-jwn}$$

avec h(n) : coefficients du filtre
N : longueur du filtre.

Les filtre RIF sont stables et présentent l'avantage de posséder une phase exactement linéaire. Par contre, ils ont l'inconvénient de nécessiter un grand nombre de multiplications comparés aux filtres RII qui satisfont des spécifications similaires. Mais, l'avance dans la technologie des semi-conducteurs a rendu l'opération d'implantation matérielle directe plus rapide et moins chère. C'est pourquoi, les filtres RIF ont pris de l'importances ces dernières années. Dans l'implantation matérielle directe, la précision dans la représentation des coefficients du filtre est proportionnelle à la taille des registres. Par conséquent, il est indispensable d'effectuer une optimisation discrète des coefficients pour maintenir la longueur du mot la plus petite que possible. Dans le cas de filtres RIF à phase linéaire, cette optimisation discrète qu'on nommera par la suite 'Synthèse discrète', se traduit généralement par la quantification (troncature ou arrondissement) des coefficients optimaux de grande précision de Parks - Mc Clellan [9] à la taille des registres. Cependant, les filtres obtenus ne sont pas optimaux, et dans la plupart des cas, il existe d'autres coefficients sur la même longueur de mot qui présentent une meilleure approximation dans le sens de Chebyshev de la réponse en fréquence désirée. Pour retrouver ces coefficients, il est indispensable d'inclure la restriction de la longueur de mot dans la procédure de synthèse de filtres.
Les auteurs [2]-[4] ont tenté de trouver des algorithmes améliorant les résultats obtenus par simple quantification. Ils utilisent la méthode du gradient simulé (Simulated Annealing). Ces méthodes sont statistiquement fructueuses seulement pour des filtres de longueur petite à cause du temps de calcul prohibitif. De plus la solution obtenue n'est pas garantie à être la meilleure dans le sens de Chebyshev. D'autres auteurs [1], [11], [13]-[16] ont élaboré des méthodes qui effectuent la synthèse de filtres directement dans l'espace discret des coefficients. Malgré qu'il soit possible d'obtenir des résultats optimaux, le temps de calcul, même avec les supers calculateurs actuels devient prohibitif quand la longueur du filtre augmente.
Le but de notre travail est d'obtenir des coefficients discrets de filtre qui présente la meilleure approximation dans le sens du critère de l'erreur choisi (critère de

Chebyshev ou critère de l'erreur quadratique moyenne) , dans un temps de calcul acceptable. Notre intérêt pour ce sujet s'appuie sur des travaux effectués dans le laboratoire Signaux et Systèmes [31]-[33]. Il a ouvert une voie à plusieurs axes de recherche dans le domaine du filtrage numérique. Celui-ci a utilisé la représentation binaire de Somme de Deux de Puissance de Deux 'S.D.P.D.' Il a présenté 3 méthodes :

La méthode de Recherche Arborescente 'R.A.' (Branch and Bound Technique).
La méthode Directe par les Moindres Carrés 'D.M.C.'.
La méthode de Séparation et Evaluation Progressive 'S.E.P.'(Breath First Search).

Dans ce mémoire, nous nous sommes basés sur les deux premières méthodes. La méthode RA présente l'avantage de procurer des solutions optimales mais le temps de calcul quand on traite plus de 8 coefficients devient prohibitif. La méthode DMC possède une grande vitesse de convergence, mais les solutions obtenues bien que performantes sont rarement équivalentes à celles de la méthode RA. Parmi les problèmes que nous avons été confrontés durant l'accomplissement de ce travail se trouve la lenteur des algorithmes, spécialement celui de la méthode RA, sur des ordinateurs de fréquence de travail CPU à 300 Mhz et de 32 Mo de RAM. L'opération effectuée pour obtenir les résultats d'un seul filtre d'ordre 7 peuvent durer plusieurs jours, alors que nous avons été amenés à tester des centaines de cas de filtres pour que l'évaluation de la méthode utilisée soit significative. De plus, le manque de la documentation récente dans ce domaine, pour comparaison, spécialement, la référence de 1ère degré concernant l'utilisation de la méthode du gradient simulé intitulée 'Application of Simulated Annealing for the Design of Special Digital Filter' de N. Benvenuto, M. Marchesi et A. Uncini, publiée dans la revue de IEEE , Signal Processing en février 1992.
L'objet de ce mémoire est de déterminer le processus adéquat de synthèse de filtres dans l'espace discret, et de se prononcer pour la représentation binaire susceptible à optimiser l'implantation du filtre sur le processeur. Ce document est divisé en 4 chapitres :

Dans le premier chapitre, nous donnons des rappels sur les représentations des nombres (en virgule fixe, en virgule flottante et en SDPD), l'effet sur la limitation de la longueur des coefficients et la théorie de filtres RIF à phase linéaire. Nous présentons aussi les méthodes que nous avons jugé les plus intéressants de la synthèse de filtres numériques existantes dans la littérature scientifique, avant de se prononcer pour le choix des méthodes et des représentations binaires usitées dans les chapitres suivants.

Dans le deuxième chapitre, notre propos a été d'étendre les travaux du laboratoire Signaux et Systèmes concernant les méthodes RA et DMC dans les trois représentations binaires en virgule fixe, en virgule flottante et en SDPD. Nous montrerons pour la méthode RA que la représentation en virgule fixe procure le filtre à plus faible erreur quadratique moyenne, mais le temps de calcul est très grand comparé à celui conçu dans la représentation SDPD qui est de performance moindre. Nous allons aussi montrer, qu'il est souvent préférable de concevoir des filtres en virgule fixe sur un mot de longueur plus petite de 4 bits que celle prescrite en utilisant SDPD, afin de gagner en précision et en temps de calcul. Dans le cas de la méthode DMC, nous allons montrer qu'il est déconseillé d'utiliser la représentation SDPD, afin

d’obtenir de bonnes performances, puisque cet algorithme possède une convergence rapide. Notre principale contribution dans ce contexte a été de prouver statistiquement qu’il est possible d’obtenir de meilleures performances que ceux en [31], grâce à un choix adéquat de représentation binaire.

Dans le troisième chapitre, nous présentons une nouvelle méthode nommée méthode Directe par les Moindres Carrés Itérative ‘D.M.C.I.’. Cette méthode améliore itérativement les résultats de DMC. Nous montrons que cette méthode retrouve souvent les performances de la méthode RA. Le temps de calcul est plus grand que celui de la méthode DMC mais nettement plus petit que celui de la méthode RA. Nous montrons que les représentations en virgule fixe et en virgule flottante procurent de meilleures performances que celles de la représentation SDPD.

Dans le quatrième chapitre, nous présentons une nouvelle méthode nommée méthode de Recherche Séquentielle et Progressive ‘R.S.P.’ basée sur la méthode RA. Cette méthode présente une nouvelle stratégie de branchement afin de réduire l’espace discret de recherche utilisé par la méthode RA. Nous montrerons que le temps de calcul sera drastiquement réduit et que nous pouvons effectuer la synthèse pour n’importe quelle longueur de mot. La réduction de l’espace discret ne semble pas dégrader l’optimalité des résultats.

Chapitre I

PRELIMINAIRES ET ETAT DE L'ART 'CONCEPTION DE FILTRES NUMERIQUES NUMERIQUES A PHASE LINEAIRE DANS

XI. L'ESPACE DISCRET DES COEFFICIENTS'

I. INTRODUCTION :

Dans ce chapitre, nous donnons les éléments de base permettants une bonne compréhension de notre travail. Une description des différentes représentations binaires des nombres et des principales méthodes de quantification sont données. Nous présentons la méthode la plus connue et intensivement utilisée pour la conception des filtres 'R.I.F.' à phase linéaire nommée méthode de Parks - Mc Clellan [9]. Cette méthode permet de retrouver les filtres qui possèdent la meilleure approximation au sens de l'erreur de Chebyshev avec des coefficients à précision infinie. Ensuite, nous mettons en relief les conséquences de la limitation du mot machine sur le gabarit du filtre.

Nous nous intéresserons aussi aux méthodes de synthèse existantes dans la littérature scientifique pour des coefficients à précision finie.

II. ESPACE CONTINU ET ESPACE DISCRET :

On appelle ensemble {v}, l'ensemble des valeurs que peuvent prendre les coefficients h(n) de filtres RIF définis par $H(z) = \sum_{n=0}^{N-1} h(n) z^{-n}$. Lorsqu'il n'y a aucune contrainte sur le nombre de chiffres significatifs, les coefficients du filtre appartiennent à un ensemble continu qu'on notera par 'E_c': $\{v\} = E_c$

où $E_c = \{ v / v \in R \}$.

En pratique on utilise des ordinateurs de longueur de mot finie. On peut montrer alors que l'ensemble des valeurs représentables, que nous nommerons par E_{disc}, est discret. Toutefois si ces valeurs sont très proches entre elles (lorsque le nombre de bits est suffisamment grand), on peut admettre moyennant une erreur acceptable que l'ensemble est continu. Dans ce cas précis, dans la suite de notre travail, on confond quelquefois l'ensemble discret, dont les valeurs sont représentées par des mots de longueur supérieure ou égale à 64 bits, avec E_c.(E_c : espace continu)

L'ensemble discret E_{disc} est un sous ensemble de E_c. Les valeurs appartenant à E_{disc} sont telles que la différence entre deux nombres quelconques est toujours égale à un multiple d'une quantité q, appelé pas de quantification.

Si nous considérons un ordinateur de longueur mot machine Nb_{ord} = 64 bits, fonctionnant en virgule fixe (détaillée dans le paragraphe III.1.1.) pour représenter les nombres purement fractionnaires (nombres compris entre -1 et +1), les valeurs représentées par cet ordinateur seront également espacées avec un pas de quantification égal à $q = 2^{-63}$ (un bit étant réservé au signe du nombre).

Avec la même représentation, dans un processeur de signaux où le nombre de bits Nb_{proc}= 16 bits, le pas de quantification est aussi constant et égal à $q=2^{-15}$.

Dans les deux exemples précédents les pas sont constants, ils dépendent uniquement du nombre de bits Nb. Si nous utilisons une autre représentation, par exemple la représentation virgule flottante, le pas devient variable. Ceci explique que le pas de quantification dépend aussi de la représentation utilisée.

Il importe de donner les remarques suivantes :

- Un ensemble discret, dépend principalement du pas de quantification c'est-à-dire du nombre de bits et de la représentation utilisée.
- Le pas de quantification q varie inversement au nombre de bits Nb. Plus les coefficients sont représentés par des mots de longueur plus grande, plus petit est le pas et meilleur est la précision.

Pour distinguer l'espace dans lequel nous allons travailler, et suite à ces deux remarques, nous définissons deux ensembles :

- Ensemble discret ordinateur $E_{disc} = E_{ord}$: ensemble des valeurs décrites suivant une représentation par des mots de longueur Nb_{ord} bits.
- Ensemble discret processeur $E_{disc} = E_{proc}$: ensemble des valeurs décrites suivant une représentation par des mots de longueur Nb_{proc} bits, avec $Nb_{proc} < Nb_{ord}$.

Un ensemble est un espace à une dimension où les coefficients du filtre prennent leur valeur. Les solutions recherchées, se composant de N valeurs, et varient à l'intérieur d'un espace à N dimensions, qu'on appellera espace discret des solutions. Il sera noté E^N. Enfin on définira par:

E^N_{ord} : l'espace discret des solutions, de dimension N, où les valeurs sont représentées par des mots à Nb_{ord} bits.

E^N_{proc} : l'espace discret des solutions, de dimension N, où les valeurs sont représentées par des mots à Nb_{proc} bits [32].

Dans notre travail qui est la 'synthèse des filtres dans l'espace discret des coefficients', nous nous intéresserons à l'espace discret nommé 'E^N_{proc}'. Dans le paragraphe suivant, nous présenterons les éléments intervenants dans notre travail tels que les différentes représentations binaires, les types de quantification, un descriptif théorique sur les filtres RIF à phase linéaire, spécialement les filtres de Parks - Mc Clellan, et les erreurs dues à la limitation de la longueur du mot des coefficients.

III. NOTIONS FONDAMENTALES :

Les bruits intervenants dans le traitement numérique de signal sont les suivants :

- Le bruit de conversion analogique/numérique du signal d'entrée.
- Le bruit de conversion des coefficients du filtre obtenu en coefficients de longueur de mot finie.
- Le bruit relatif à l'exécution des opérations arithmétiques dans un mot machine dont la longueur n'augmente pas.
- Le bruit de conversion numérique/analogique du signal de sortie.

Le calcul des coefficients se fait généralement à l'aide d'ordinateurs de longueur de mot Nb_{ord} donnant une précision satisfaisante. Lors de la mise en œuvre du filtre ainsi obtenu sur un processeur de longueur de mot machine $Nb_{proc} < Nb_{ord}$ il faut obligatoirement quantifier ces valeurs ; ceci pourrait changer la réponse en fréquence du filtre ou – en d'autres termes– la position des zéros du filtre. Ces modifications sont généralement néfastes. Il peut arriver qu'après quantification, le filtre ne

satisfasse plus les spécifications sur lesquelles le calcul des coefficients non quantifiés étaient basés.

Cette sensibilité des coefficients à la quantification varie d'une structure de filtre à une autre. La position des zéros dans le cercle unité relève uniquement de l'effet de quantification.

Nous avons supposé à chaque instant que les coefficients du filtre pouvaient prendre n'importe quelle valeur. Pour la catégorie très importante des signaux et des systèmes numériques, cette supposition ne tient pas parce que chaque quantité est représentée par la combinaison d'un nombre fini de bits (c'est à dire un mot binaire ou simplement un mot). Un bit est un nombre qui peut prendre seulement deux valeurs différentes (habituellement 0 et 1). Avec une longueur de mot Nb bits, nous pouvons distinguer au plus 2^{Nb} valeurs différentes. Quand nous sommes libres de choisir Nb, nous pouvons rendre la représentation numérique aussi précise que nous le voulons et donc approcher tout système numérique ou tout signal numérique avec une précision aussi grande que nous le souhaitons.

Cependant dans la pratique, la réalité est complètement différente. Pour des raisons d'économie, nous sommes souvent intéressés de savoir comment nous pouvons choisir la valeur Nb la plus basse possible sans introduire d'erreurs importantes. Nous sommes alors inévitablement confrontés à un certain nombre d'effets de nature très variée, causés par la longueur de mot finie que nous utilisons. Ces effets sont souvent très compliqués, et difficilement à mettre en équation. Les seules conclusions que nous pouvons tirer à leur sujet sont des conclusions statistiques (reliées aux valeurs quadratiques moyennes, aux valeurs maximales, etc.).

L'exploitation fructueuse du mot à Nb bits, dépend du choix adéquat de la représentation binaire et de l'algorithme à utiliser. Dans ce contexte, nous allons présenter les représentations binaires, les plus utilisées, qui nous intéresseront dans notre travail.

III.1. REPRESENTATIONS DES NOMBRES :

La représentation des nombres par un mot machine fini se fait suivant des normes établies que nous nommons par «code numérique ». Les représentations les plus courantes sont les suivantes :

1) En signe et amplitude (ou valeur absolue).
2) En complément à un.
3) En complément à deux.

Dans les trois représentations (1), (2) et (3), le bit situé le plus à gauche est le bit de signe ; il est égal à « 0 » en présence d'un nombre positif, et à « 1 » dans le cas d'un nombre négatif.

Dans la représentation en signe et amplitude, à part le bit de signe les autres bits représentent la valeur absolue du nombre.

Dans la représentation en complément à un, les nombres négatifs sont obtenus en remplaçant chacun des bits du nombre positif correspondant par le bit opposé (inversion de bits : transformer les ‘0’ en ‘1’ et les ‘1’ en ‘0’).
Les nombres négatifs dans la représentation en complément à deux, sont obtenus en inversant tous les bits du nombre positif correspondant et en ajoutant un « un » à la place correspondante au bit de plus faible poids.
Parmi ces trois représentations, la première et la troisième sont les plus largement utilisées. Leur principal avantage est de réaliser les multiplications, les additions et les soustractions.
Nous nous intéresserons dans notre étude qui suit aux nombres fractionnaires.

Les différentes représentations existantes sont les suivantes :

*) la virgule fixe.
*) la virgule flottante.
*) la représentation SDPD (Somme de Deux de Puissance de Deux).

III.1.1. REPRESENTATION EN VIRGULE FIXE :

Le nombre h(n) est représenté par un mot binaire de (Nb+1) bits.

a(0)	a(1)	a(2)	.....	a(Nb-2)	a(Nb-1)	a(Nb)

représentation d’un nombre en binaire virgule fixe

où les a(i) représentent les valeurs des bits du nombre considéré. (0 ou 1)
Le bit a(0) est le bit de signe.

a(0)=0 si le nombre est positif.
a(0)=1 si le nombre est négatif.

La représentation d’un nombre fractionnaire dans un champ fixe de Nb+1 positions (registre de Nb+1 cellules) ne permet pas de représenter les valeurs inférieures à 2^{-Nb}.

Les valeurs représentées par ce système de numération sont discrètes et espacées d’un pas constant égal à $q=2^{-Nb}$ (q: étant le pas de quantification) .

A titre d’exemple(fig.1), pour un espace discret à 8 bits, nous reservons sept bits pour la valeur absolue du nombre.
A part ‘0’ et en valeur absolue, la valeur la plus faible représentable est: $1.2^{-7} = 0.0078125$, tandis que la valeur la plus élevée est: $1.2^{-7}+ 1.2^{-6}+ 1.2^{-5}+ 1.2^{-4}+ 1.2^{-3}+ 1.2^{-2}+ 1.2^{-1}=0.9921875$

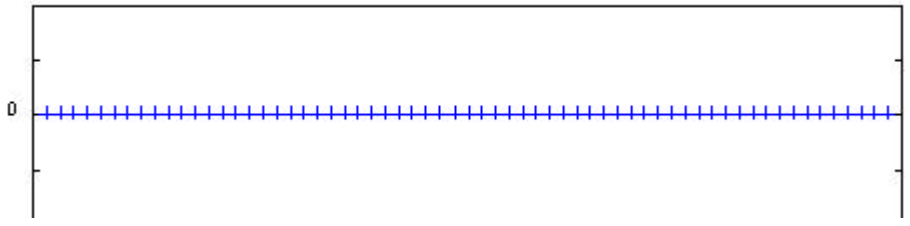


Fig.1. Signe et amplitude des valeurs discrètes dans l'intervalle [-1, 1] suivant la représentation en virgule fixe dans un espace discret de longueur de mot 8 bits.

Nous remarquons sur la figure 1 que la représentation en virgule fixe est une représentation uniforme et le pas est constant. La représentation des coefficients en virgule fixe conduit à une valeur maximale des erreurs de quantification indépendante des valeurs représentées. Dans ce cas la représentation des coefficients du filtre en virgule fixe affecte les coefficients à valeurs absolues faibles, d'une erreur relative importante.

III.1.2. REPRESENTATION EN VIRGULE FLOTTANTE:

Un nombre peut être représenté sous forme d'un nombre réel et d'un facteur multiplicatif qui est une puissance d'une base. Cette représentation est dite flottante [24],[32]. Un nombre h(n) dans la représentation en virgule flottante, peut être écrit comme suit :

$$h(n) = \pm\ M.B^{-E}. \qquad (1)$$

où M : la mantisse, elle est réelle et positive.
B : la base, elle est entière. En binaire elle est choisie égale à 2.
E : est l'exposant, il est entier.

La représentation d'un nombre h(n) en virgule flottante par un mot de (1+e+m) bits est donnée comme suit.

signe |	exposant (e bits)			|	mantisse (m bits)		
a(0)	a (1)	. . .	a (e)	b(1)	. . .	b(m)

représentation d'un nombre en binaire virgule flottante

où a(0) : bit de signe du nombre.
m : nombre de bits réserves à la mantisse.
Avec

$$M = \sum_{i=1}^{m} b(i)\,2^{-i} \qquad (2)$$

e : nombre de bits pour représenter la valeur de l'exposant.

$$E = \sum_{i=1}^{e} a(i)\,2^{e-i} \qquad (3)$$

Cette représentation est choisie pour diminuer au maximum les erreurs d'arrondi dans une représentation en champ fixe. Ceci en lui offrant une flexibilité et une dynamique plus grande. On choisit en général l'exposant de telle façon qu'il n'y ait pas de zéro non significatif au début de la mantisse. Dans ce cas la représentation est dite normalisée. L'utilisation d'une mantisse normalisée permet de conserver la précision la plus grande pour les nombres. L'erreur sur une mantisse fractionnaire de m bits est de 2^{-m} dans le cas d'une quantification par troncature ou de $(1/2)\,2^{-m} = 2^{-m-1}$ dans le cas d'une quantification par arrondi [32].L'erreur relative, dans le cas d'un

champ binaire normalisé de k bits est égale à 2^{-k+1} , elle est indépendante du nombre représenté [25]. La mantisse est, plus fréquemment, considérée comme un nombre purement fractionnaire plutôt que comme un nombre entier.
A titre d'exemple(fig.2), pour un espace discret à 8 bits, à part '0' et en valeur absolue, la valeur la plus faible est : $2^{-4}.2^{-7}$ =0.00048828125,
tandis que la valeur la plus élevée est : $1.2^{-4} + 1.2^{-3} + 1.2^{-2} + 1.2^{-1}$ =0.9375

Conventionnellement dans la suite de notre travail, nous avons choisi un nombre de bits constant pour l'exposant (3 bits), pour chaque longueur de mot supérieure à 4 bits.

En général, pour un mot dont la longueur des coefficients est de Nbproc bits, nous avons pris:

- 3 bits pour l'exposant.
- 1 bits réservé au signe.
- Le nombre de bits restant pour la mantisse:
  Nbmantisse = Nbproc -4.

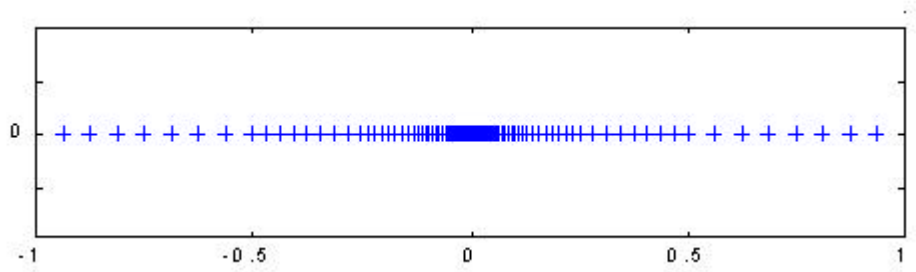


Fig.2. Signe et amplitude des valeurs discrètes dans l'intervalle [-1, 1] suivant la représentation en virgule flottante dans un espace discret de longueur de mot 8 bits.

Nous remarquons sur la figure 2 que le pas entre deux valeurs discrètes adjacentes n'est pas constant. Plus nous nous approchons du '0', plus le pas diminue. Par conséquent, le nombre de valeurs discrètes est plus important autour de '0' que les autres régions. La difficulté réside dans les opérations arithmétiques comparée à la représentation en virgule fixe. De ce fait, nous présenterons dans le paragraphe suivant une troisième représentation nommée SDPD 'représentation en somme de deux de puissance de deux'. Celle ci possède les propriétés des deux précédentes représentations, la virgule fixe et la virgule flottante.

III.1.3. REPRESENTATION BINAIRE EN SOMME DE DEUX DE PUISSANCE DE DEUX (S.D.P.D.) :

La représentation binaire en virgule flottante entraîne une complication des opérations arithmétiques. L'algorithme de l'addition (ou de la soustraction) est plus complexe en virgule flottante qu'en virgule fixe, le plus souvent dans les réalisations spécialisées des filtres numériques, on utilise la représentation binaire en virgule fixe.

Dans cette section nous étudions une autre représentation binaire en ''Somme de Deux Puissances de Deux'' ou S.D.P.D.[14],[17],[19]. Cette représentation permet de lever les inconvénients des deux précédentes représentations binaires en virgule fixe et en virgule flottante. Elle permet de représenter les coefficients à faible valeur avec une erreur plus petite que celle en représentation à virgule fixe. La dynamique

des valeurs est plus grande qu'en virgule flottante pour un ensemble discret processeur 'E^N_{proc}'plus petit.

La valeur du coefficient h(n), dans la représentation S.D.P.D., est décrite par une somme de deux termes, où chacun des deux termes est écrit dans la représentation binaire en virgule flottante.

$$h(n) = \sum_{i=1}^{2} m_i(n).2^{-ei(n)} \qquad (4)$$

où $m_i(n)$: mantisse entière, $m_i(n) \in [-1, 0, 1]$.
$e_i(n)$: exposant, $e_i(n) \in [0, 1, 2, ..., p-2, p-1]$.
p : nombre de valeurs que peut prendre l'exposant (a partir de 0).
dans cette représentation la mantisse ne prend que trois valeurs { -1, 0, 1 }, deux bits sont suffisants pour la représenter. La représentation de la valeur du coefficient h(n) en S.D.P.D. est donnée par:

1° terme		2° terme	
$m_1(n)$	$e_1(n)$	$m_2(n)$	$e_2(n)$
2bits	e bits	2bits	E bits

Représentation d'un nombre en S.D.P.D..

La représentation du nombre h(n) nécessite Nb = 2.(2+e) bits, e étant le nombre de bits de chaque exposant.

A titre d'exemple (fig.3), pour un espace discret à 10 bits, à part '0' et en valeur absolue, la valeur la plus faible est : 2^{-7} = 0.0078125, tandis que la valeur la plus élevée est : 1.

nous donnons dans le tableau ci-dessous, les valeurs discrètes dans la représentation S.D.P.D. pour Nb = 10.

- $1.2^0 = 1.0000$
- $1.2^0 - 1.2^{-7} = 0.9921875$
- $1.2^0 - 1.2^{-6} = 0.984375$

…….

- $1.2^{-1} + 1.2^{-3} = 0.6250$
- $1.2^{-1} = 0.5000$
- $1.2^{-1} - 1.2^{-3} = 0.3750$

……..

- $1.2^{-6} = 0.015625$
- $1.2^{-7} = 0.0078125$
- $0 = 0.0000$

…….

- $-1.2^{-7} = -0.9921875$
- $-1.2^0 = -1.0000$

Tableau de valeurs en S.D.P.D (Nb=10).

Comme nous pouvons le remarquer, les valeurs ne sont pas régulièrement espacées. Il existe une plus grande concentration dans des régions comme autour de 0, de ± 0.5

et ± 0.25. L'écart entre deux valeurs consécutives n'est, par conséquent, pas toujours constant comme en virgule fixe q_{min}=0.0078125 et q_{max} = 0.125. La représentation S.D.P.D. est donc une représentation non uniforme.

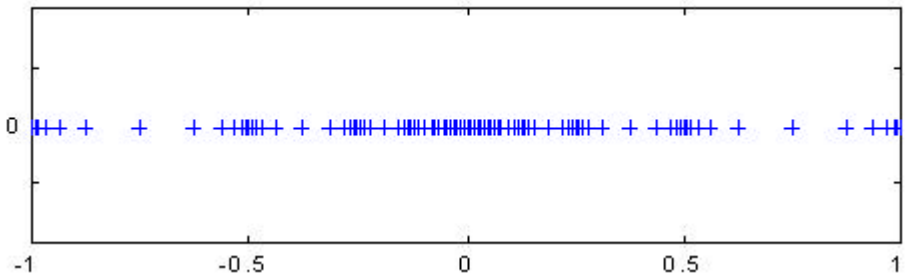


Fig.3. Signe et amplitude des valeurs discrètes dans l'intervalle [-1, 1] suivant la représentation SDPD dans un espace discret de longueur de mot 10 bits.

Nous remarquons (fig.3) que la concentration des valeurs discrètes est autour de quelques principales valeurs telles que : 0, ±1, ± 0.5, ± 0.25, 0.125, due à la nature de la représentation.

III.1.4. ETUDE COMPARATIVE DES REPRESENTATIONS VIRGULE FIXE, VIRGULE FLOTTANTE ET SDPD :

Le but de cette étude comparative est de mettre en relief l'intérêt du choix adéquat de la représentation binaire dans la conception de filtres avec une grande précision et un temps de calcul réduit. Dans le but de trouver la meilleure représentation binaire qui permet une implantation à faible erreur des coefficients du filtre sur un processeur de signaux de longueur de mot finie avec un bruit réduit, nous avons effectué l'étude comparative entre la représentation en virgule fixe, la représentation en virgule flottante, et la représentation SDPD dans ce qui suit.

Considérons F_{fix}, F_{flot} et F_{sdpd} les filtres qui présentent la meilleure approximation dans le sens de Chebyshev respectivement dans les représentations en virgule fixe , en virgule flottante et en SDPD. Les filtres F_{fix} , F_{flot} et F_{sdpd} sont conçus dans un espace discret à Nb bits dans des temps de calcul respectivement t_{fix}, t_{flot}, et t_{sdpd}. Pour un mot machine de longueur Nb bits, nous avons:

- F_{fix} est le filtre qui a la meilleure approximation au sens de Chebyshev en représentation à virgule fixe de temps de calcul de t_{fix}. Nous posons l'erreur minmax correspondante égale à E_{fix}.
- F_{flot} est le filtre qui a la meilleure approximation au sens de Chebyshev en représentation à virgule flottante de temps de calcul de t_{flot}. Nous posons l'erreur minmax correspondante égale à E_{flot}.
- F_{sdpd} est le filtre qui a la meilleure approximation au sens de Chebyshev en représentation SDPD de temps de calcul de t_{sdpd}. Nous posons l'erreur minmax correspondante égale à E_{sdpd}.

L'étude que nous nous proposons d'effectuer est de savoir quelle est la représentation (virgule fixe, virgule flottante ou SDPD) qui présente la plus petite erreur au sens de Chebyshev et le temps de calcul correspondant.

Pour une comparaison significative entre les trois représentations binaires en virgule fixe, en virgule flottante et en SDPD, nous devons prendre en compte :

- de la précision au sens de Chebyshev (E_{fix}, E_{flot} et E_{sdpd})
- du temps de calcul (t_{fix}, t_{flot} et t_{sdpd})

Nous nommons par valeur admissible ou valeur permise la valeur discrète pouvant être représentée sur un mot machine de longueur Nb bits. nous nommons par pas minimal, le plus petit pas entre deux valeurs discrètes adjacentes, et par pas maximal le plus grand pas entre deux valeurs discrètes adjacentes pour une représentation donnée.

- Pour la représentation à virgule fixe nous avons:

- le nombre de valeurs admissibles noté par Nb_{ad} est

$$Nb_{ad} = 2^{Nb-1} \quad (5)$$

avec Nb: longueur de mot machine.

- Le pas minimal noté par p_{min} et le pas maximal noté par p_{max} sont égaux

$$p_{min} = p_{max} = 2^{-Nba+1} \quad (6)$$

• Pour une représentation à virgule flottante nous avons:

- le nombre de valeurs admissibles est

$$Nb_{ad} = 1 + 2.[\ 2^{Nbmantisse} + (2^{Nbmantisse-1}.(\ 2^{Nbexp}-1))] \quad (7)$$

Avec Nbmantisse : Nombre de bits dans la mantisse.
et Nbexp: Nombre de bits réservé à l'exposant (conventionnellement nous l'avons choisi égal à trois bits).
alors en remplaçant Nbexp =3 nous aurons

$$Nb_{ad} = 1 + 2.[\ 2^{Nbmantisse} + 7.(2^{Nbmantisse-1}.)] \quad (8)$$

- le pas minimal est

$$p_{min} = 2^{-Nbmantisse} . 2^{-7} \quad (9)$$

- le pas maximal est

$$p_{max} = 2^{-Nbmantisse} \quad (10)$$

Pour une représentation SDPD (cf II.1.3):

- Le nombre de valeurs admissibles est

$$Nb_{ad} = 1 + 2.[\ 2^{Nbexp} + (\ 2^{Nbexp} - 2)^2\ ] \qquad (11)$$

- Le pas minimal est

$$p_{min} = 2^{-a} \qquad (12)$$

avec:

$$a = 2^{Nbexp} - 1. \qquad (13)$$

- Le pas maximal est

$$p_{max} = 2^{-3} \qquad (14)$$

Ces formules reportées précédemment exprimant le nombre de valeurs admissibles, le pas minimal et le pas maximal ont été élaborées à partir de la définition de chaque représentation. Une comparaison primitive entre ces formules pour chaque représentation n'est pas significatif. Elle dépend de la longueur du mot machine dans lequel le filtre sera conçu. A titre d'exemples, nous avons choisi deux espaces discrets de longueur de mot de 8 bits et 16 bits et nous avons construit les tableaux suivants:

Longueur du mot machine	Virgule fixe	Virgule flottante	S.D.P.D.
8 bits	255	143	17
16 bits	65535	36863	7817

Tableau1 : Nombre de valeurs discrètes admissibles dans l'espace discret [-1, 1] à 8 bits et à 16 bits suivant chaque représentation.

Longueur du mot du processeur	Virgule fixe	Virgule flottante	S.D.P.D.
8 bits	2^{-7}	2^{-11}	2^{-3}
16 bits	2^{-15}	2^{-19}	2^{-63}

Tableau2 : Valeur du plus petit pas dans l'espace discret [-1, 1] sur 8 bits et 16 bits suivant chaque représentation.

Longueur du mot du processeur	Virgule fixe	Virgule flottante	S.D.P.D.
8 bits	2^{-7}	2^{-4}	2^{-3}
16 bits	2^{-15}	2^{-12}	2^{-3}

Tableau3 : Valeur du plus grand pas dans l'espace discret [-1, 1] sur 8 bits et 16 bits suivant chaque représentation.

Dans le tableau 1, nous remarquons que le nombre de valeurs admissibles est plus grand dans la représentation à virgule fixe que celui dans la représentation à

virgule flottante et SDPD. Dans cette dernière représentation, la majeure partie du mot machine a été consommée par la mantisse. Dans la représentation à virgule flottante, pour des raisons pratique, nous avons fixé l'exposant à 3 bits. par conséquent, la dynamique des valeurs serait réduite. Dans les tableaux 2 et 3, nous remarquons que la précision est plus grande dans la représentation à virgule flottante pour Nb = 8 bits et dans SDPD pour Nb=16 bits.

Dans ce paragraphe, nous avons donné les propriétés de chaque représentation binaire tel que le nombre de valeurs admissibles, le pas minimal et le pas maximal. La connaissance de ces propriétés serait prise en considération dans l'étude qui suit pour une bonne évaluation des performances des algorithmes utilisés, pour la précision et le temps de calcul. Le pas minimal et maximal interviennent dans la précision tandis que le nombre de valeurs admissibles influe sur le temps de calcul. Pour la synthèse des filtres dans l'espace discret des coefficients, le choix de la représentation dépend des ressources disponibles telles que la précision, le temps de calcul et la complexité de l'algorithme utilisé. L'évaluation de la meilleure représentation est liée à la méthode de synthèse utilisée.

Dans le paragraphe suivant, nous allons présenter les deux types de quantification troncature et arrondissement ainsi que les erreurs engendrées afin de mieux évaluer la qualité des méthodes de synthèse de filtres numériques utilisant la quantification.

III.2. QUANTIFICATION :

III.2.1. TRONCATURE :

Nous représentons 'x' par la valeur discrète 'x_T', adaptable par troncature à la taille du mot binaire fini. La figure 4. illustre la fonction troncature.

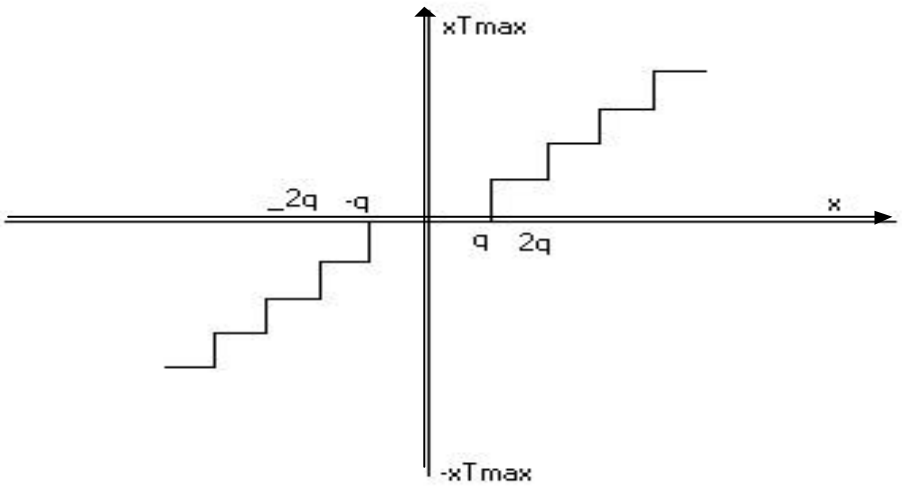


Fig. 4. Représentation de la fonction troncature

q : représente le pas de quantification et

$$x_T = q.Ent[x/q] \qquad (15)$$

Ent : désigne la partie entière.
Sgn(x) : le signe de x.

L'erreur d'arrondi est $e_T = x_T - x$ et on a :
$-q < e_T £ +q$.

III.2.2. ARRONDISSEMENT :

Nous représentons 'x' par la valeur discrète 'x_A' obtenue par l'opération d'arrondissement de 'x'. la figure 5. illustre la fonction arrondi.

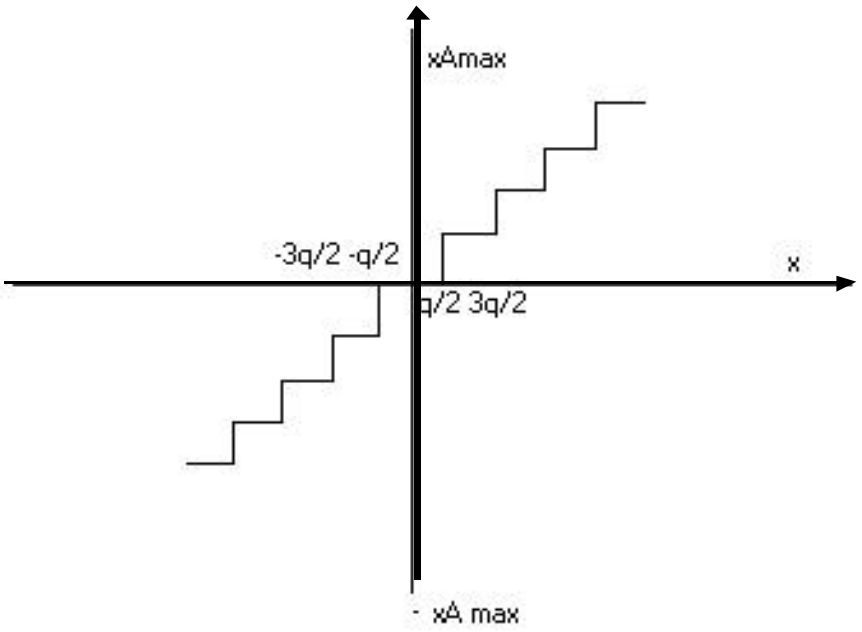


Fig. 5. Représentation de la fonction arrondi

q : représente le pas de quantification et

$$x_A = q.Ent[x/q + 0.5sgn(x)] \quad (16)$$

Ent : désigne la partie entière.
Sgn(x) : le signe de x.

L'erreur d'arrondi est $e_A = x_A - x$ et on a : $-q/2 < e_A £ +q/2$.

III.3. THEORIE DES FILTRES R.I.F. A PHASE LINEAIRE :
III.3.1. INTRODUCTION :

Dans ce paragraphe, nous allons présenter les notions fondamentales et les caractéristiques des filtres RIF à phase linéaire, sujet de notre travail de recherche de base. Dans la conception des filtres, la classe des filtres à durée finie possède la propriété intéressante, telle que la stabilité. De plus, les filtres R.I.F. peuvent être conçus de façon à ce que leur réponse en fréquence possède une phase exactement linéaire.

III.3.2. CARACTERISTIQUES DES FILTRES R.I.F. A PHASE LINEAIRE :

Soit {h(n)} une séquence causale à durée finie définie sur l'intervalle $0 \le n \le N-1$. La transformée de Fourier de {h(n)} est :

$$H(e^{jw}) = \sum_{n=0}^{N-1} h(n)\, e^{-jwn} \qquad (17)$$

posons

$$H(e^{jw}) = \pm \left| H(e^{jw}) \right| e^{-j\theta(w)} \qquad (18)$$

Avec $\left| H(e^{jw}) \right|$ et $\theta(w)$ respectivement l'amplitude et la phase de $H(e^{jw})$.

Pour qu'un filtre R.I.F. ait une phase exactement linéaire, il faut que la phase $\theta(w)$ soit de la forme suivante :

$$\theta(w) = -\alpha.w \quad \text{avec} \quad -\pi < w < \pi \qquad (19)$$

ou α est un retard de groupe constant.

Par égalité de (17) à (18) nous avons

$$H(e^{jw}) = \sum_{n=0}^{N-1} h(n)\, e^{-jwn} = \pm \left| H(e^{jw}) \right| e^{-j\theta(w)} \qquad (20)$$

Nous obtenons

$$\tan(\alpha.w) = \frac{\sum_{n=0}^{N-1} h(n).\sin wn}{h(0) + \sum_{n=1}^{N-1} h(n).\cos wn} \qquad (21)$$

Il existe deux solutions possibles :

- $\alpha = 0$: qui entraîne que h(0) est arbitraire et h(n) =0 pour $n \neq 0$, résultat peu utile.
- $\alpha \neq 0$: qui entraîne que l'équation (21) s'écrit

$$\sum_{n=0}^{N-1} h(n).\sin w(\alpha - n) = 0 \qquad (22)$$

La solution de (22) si elle existe est unique et est de la forme

$\alpha = (N-1)/2$ et
$h(n) = h(N-1-n)$ avec $0 \le n \le N-1$

Nous remarquons que pour chaque cas N, il y a une seule valeur de retard de groupe 'α=(N-1)/2', la réponse impulsionnelle doit être d'une symétrie spéciale.

- Si 'N' est impaire, 'α' est entier, alors, le centre de symétrie de la réponse impulsionnelle du filtre à phase linéaire est l'échantillon du milieu (figure 6.).
- Si 'N' est paire, alors, le centre de symétrie de la réponse impulsionnelle du filtre à phase linéaire est le milieu entre deux échantillons (figure7.).

On dit qu'un filtre est à phase exactement linéaire, s'il possède un retard de groupe constant et un retard de phase constant. Si on désire seulement que le retard de groupe soit constant (comme c'est souvent la cas), alors le filtre à phase linéaire se définit comme suit

$$H(e^{jw}) = \pm \left| H(e^{jw}) \right| e^{-j(\beta-\alpha w)} \qquad (23)$$

Les seules nouvelles solutions pour {h(n)}, α et β sont :

$$\alpha=(N-1)/2, \quad \beta = \pm\pi/2 \qquad (24)$$

et $h(n) = - h(N-1-n)$ avec $0 \le n \le N-1$

Les filtres solutions ont un retard de (N-1)/2 échantillons avec des réponses impulsionnelles antisymétriques autour du centre de la séquence opposée au réelle séquence à phase linéaire.

- Pour N impaire, h(N-1)/2 doit égale à 0. Le centre de symétrie de la réponse impulsionnelle du filtre à phase linéaire est l'échantillon du milieu égale à 0 (figure 8.).
- Pour N paire, le centre de symétrie de la réponse impulsionnelle du filtre à phase linéaire est le milieu entre deux échantillons (figure 9.).

Dépendant de la valeur de N (paire ou impaire) et du type de la symétrie de la séquence à réponse impulsionnelle (symétrique ou antisymétrique), il existe quatre cas possibles des filtres RIF à phase linéaire [5].

III.3.3. REPONSE EN FREQUENCE DES FILTRES R.I.F. A PHASE LINEAIRE :

Soit la réponse impulsionnelle (23), nous posons :

$H^*(e^{jw}) = \pm \left| H(e^{jw}) \right|$. α, β sont définies suivant l'équation (24).

$H(e^{jw})$ peut être exprimée en terme des coefficients de la réponse impulsionnelle pour chacun des quatre cas du filtre à phase linéaire.

- cas 1 (réponse impulsionnelle symétrique et symétrie impaire) :

$$H(e^{jw}) = e^{-jw((N-1)/2)} \sum_{n=0}^{N-1} a(n) \cos wn \qquad (25)$$

avec a(0)=h((N-1)/2) et a(n) =2h((N-1)/2-n) pour n=1……(N-1)/2

- cas 2 (réponse impulsionnelle symétrique et symétrie paire) :

$$H(e^{jw}) = e^{-jw((N-1)/2)} \sum_{n=0}^{N-1} b(n) \cos w(n-1/2) \qquad (26)$$

avec b(n) =2h(N/2-n) pour n=1……(N)/2

à w= π nous avons $H^*(e^{jw}) = 0$, ce qui signifie que les filtres passe-haut ne peuvent pas être évalués avec ce type de filtre.

- cas 3 (réponse impulsionnelle antisymétrique et symétrie impaire) :

$$H(e^{jw}) = e^{-jw((N-1)/2)} e^{j\pi/2} \sum_{n=0}^{N-1} c(n) \sin wn \qquad (27)$$

avec c(n) =2h((N-1)/2-n) pour n=1……(N-1)/2
Ce filtre à phase linéaire possède une réponse en fréquence imaginaire.

- cas 4 (réponse impulsionnelle antisymétrique et symétrie paire) :

$$H(e^{jw}) = e^{-jw((N-1)/2)} e^{j\pi/2} \sum_{n=0}^{N-1} d(n) \sin w(n-1/2) \qquad (28)$$

avec d(n) =2h(N/2-n) pour n=1……N/2[5]

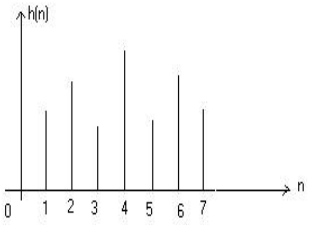


fig.6. cas 1 : réponse impulsionnelle symétrique et symétrie impaire

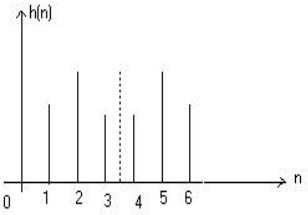
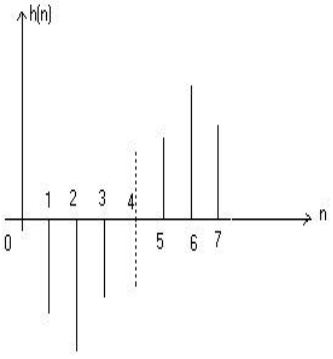
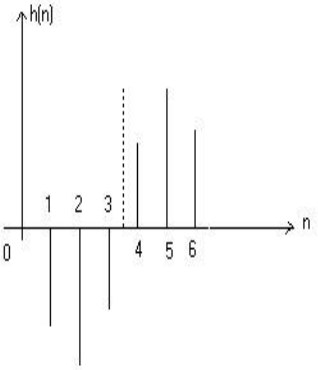


fig.7. cas 2 : réponse impulsionnelle symétrique et symétrie paire





antisymétrique et symétrie impaire antisymétrique et symétrie paire

Les figures 6, 7, 8, et 9 représentent la réponse impulsionnelle de chaque cas étudié pour les filtres RIF à phase linéaire. Le trait en pointillé représente l'axe de symétrie. La répartition des coefficients est relatif au cas considéré. L'approche mathématique la plus utilisée jusqu'à présent dans le cas de coefficients à précision infinie est celle de Parks-McClellan (P.M.C.). Le filtre de P.M.C. possède la meilleure approximation au sens de Chebyshev. Son fondement mathématique et ses propriétés sont développés dans le paragraphe suivant.

III.4. CALCUL DES COEFFICIENTS D'UN FILTRE A PHASE LINEAIRE PAR P.M.C. :

III.4.1. INTRODUCTION :

Le filtre R.I.F. à phase linéaire conçu par PARKS-McCLELLAN utilisant l'algorithme de Remez représente la meilleure approximation au sens de Chebyshev avec des coefficients à précision infinie. Auparavant, plusieurs chercheurs ont étudié les problèmes de conception (R.I.F.) pour certains types de filtre en utilisant des algorithmes différents. L'importance de cette nouvelle approche réside dans la combinaison entre la rapidité de la procédure de Remez avec la capacité de conception d'une grande classe de types de filtre, avec des filtres peu communs tels que les filtres passe bande à bandes multiples, filtre avec la transformée de Hilbert et différentiateurs et des filtres plus communs tels que passe bande, coupe bande, passe bas et passe haut. La réponse

en fréquence peut ainsi être approximée. Dans ce paragraphe, nous allons présenter le fondement mathématique de la méthode P.M.C. pour des coefficients à précision infinie.

III.4.2. FORMULATION DU PROBLEME D'APPROXIMATION :

Soit la réponse en fréquence d'un filtre R.I.F. :

$$H(f) = \sum_{k=0}^{N-1} h(k) \exp(-j2\pi kf) \qquad (29)$$

La réponse en fréquence d'un filtre R.I.F. à phase linéaire s'écrit :

$$H(f) = G(f) \exp(j(L\pi/2-((N-1)/2)2\pi f)) \qquad (30)$$

G(f) : fonction à valeur réelle. L=0 ou 1.

Il existe quatre cas de filtre R.I.F. à phase linéaire, ils dépendent de la parité de la longueur et de la symétrie de la réponse impulsionnelle (paire ou impaire), (positive (L=0) ou négative (L=1)) respectivement.

Par symétrie positive , nous avons h(k) = h(N-1-k).
Par symétrie négative , nous avons h(k) = -h(N-1-k).

Dans tous les cas, on ne s'intéresse qu'à G(f) fonction réelle qui est utilisée pour approximer les spécifications de l'amplitude idéale désirée, puisque le terme de la phase linéaire n'a aucun effet sur la réponse en amplitude.

Auparavant, tous les algorithmes déjà établis, se sont concentrés sur le cas numéro 1, dans cette approche on a pu combiné les quatre cas dans un seul algorithme en notant que G(f) s'écrit comme :

$$G(f) = Q(f)\, P(f) \qquad (31)$$

où P(f) est une combinaison de fonctions cosinus qui dépende de chaque cas.

$$P(f) = \sum_{k=0}^{r-1} \alpha(k) \cos(2\pi kf) \qquad (32)$$

et $\alpha(k)$ est une réponse implusionnelle dépendante du cas considéré.

Les quatre cas ont été écrits sous une forme commune afin que l'algorithme de Remez s'accomplisse convenablement.

Le problème d'approximation d'origine réside dans la minimisation du maximum de l'erreur absolue pondérée définie comme :

$$\|E(f)\| = \max\{W(f)|D(f) - G(f)|\} \text{ avec } f \in F \qquad (33)$$

W(f) : fonction de pondération.
F : sous ensemble de fréquence dans les bandes d'intérêt (bande passante et bande atténuée).
D(f) : réponse en amplitude désirée.

En remplaçant G(f) par sa valeur on aura :

$$\|E(f)\| = \max\{W'(f)|D'(f) - P(f)|\} \text{ avec } f \in F' \qquad (34)$$

avec $W'(f) = W(f)\,Q(f)$. $D'(f) = D(f)/Q(f)$. $F' \subset F$.

III.4.3. THEOREME DE L'ALTERNANCE :

Soit G(f) et P(f) des fonctions définies respectivement suivant Eq. 31 et Eq. 32. Une condition suffisante et nécessaire pour que P(f) soit l'unique et meilleure approximation au sens de Chebyshev à une fonction continue D'(f) dans une gamme de fréquence F' est que :

E(f)= (W'(f)|D'(f) – P(f)|), expose r+1 fréquences extrêmes dans F' notées par 'Fi'. I = 1, 2,…., r+1,

où F1 < F2 < …..< Fr < Fr+1.

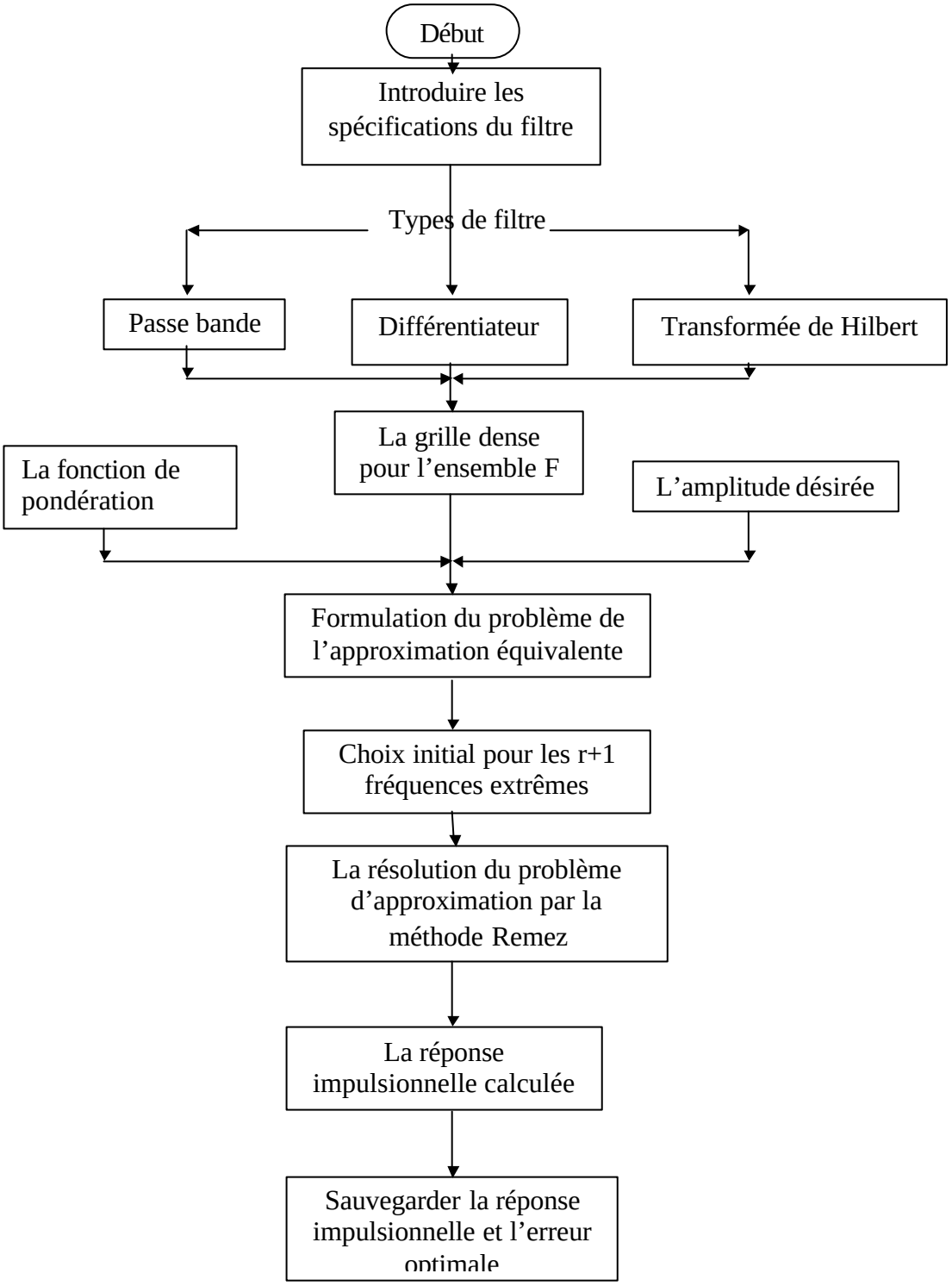
Avec E(Fi) = - E(Fi+1), i= 1,….,r.

Et |E(Fi)|= max (E(f)) pour f ∈ F'.

Un algorithme peut donc être conçu pour satisfaire la condition sur l'erreur du filtre dans le théorème d'alternance [9].

III.4.4. DESCRIPTION DE L'ALGORITHME DE CONCEPTION DE FILTRE DE PARKS-McCLELLAN SOUS L'ALGORITHME D'ECHANGE DE REMEZ:

La conception de cet algorithme consiste en une section de lecture des données, formulation du problème par une approximation appropriée équivalente, solution du problème d'approximation en utilisant la méthode de Remez, et enfin le calcul de la réponse impulsionnelle du filtre. Nous pourrons représenter dans la fig.10, l'algorithme de PARKS-Mc CLELLAN sous l'organigramme général suivant :



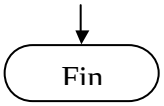
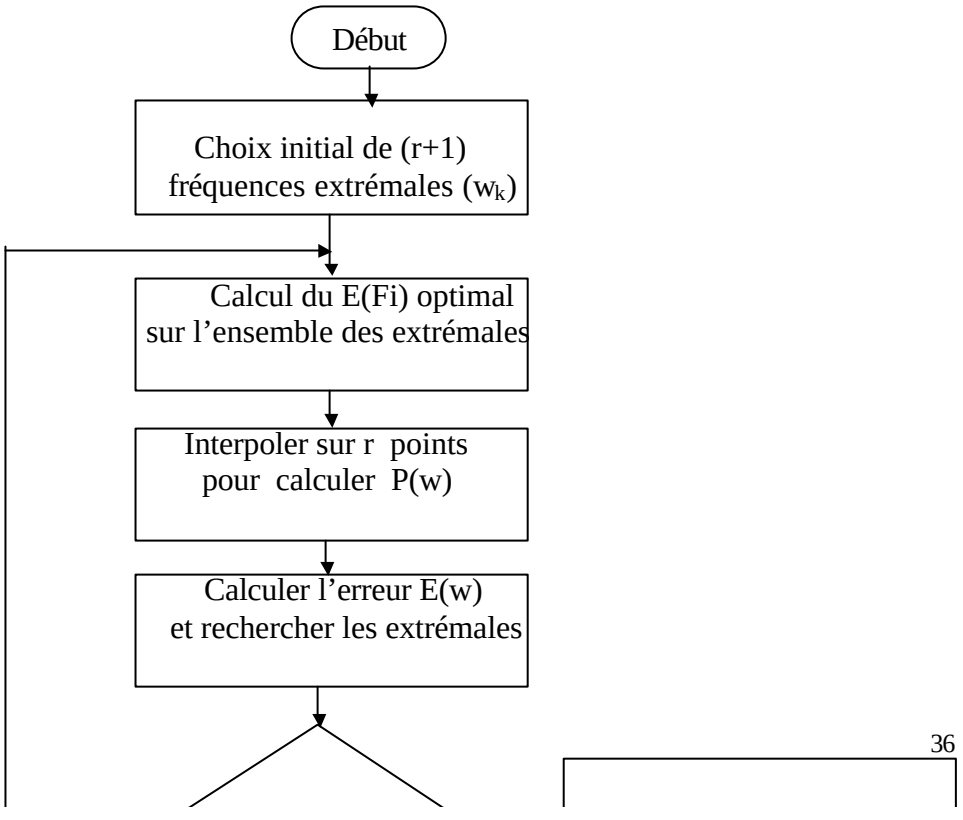


Fig.10 Organigramme de conception de filtre de PARKS-Mc CLELLAN à l'aide de méthode Remez

L'algorithme d'échange de Remez comporte 4 étapes :

1. Spécifications de la réponse en fréquence désirée D(w), de la fonction de pondération W(f) et du degré N du filtre.
2. Détermination du problème d'approximation équivalent par calcul de W'(f), D'(f) et P(f).
3. Solution du problème d'approximation en utilisant de l'algorithme d'échange de Remez.
4. Calcul des coefficients du filtre par transformée discrète de Fourier inverse.

Dans la première étape, il existe une ambiguïté. N étant fixé, rien n'assure que la solution optimale satisfera le gabarit imposé. Il existe des formules empiriques qui donnent une valeur approximative de N minimal. Ensuite, par augmentation ou diminution de celui-ci en fonction des résultats, on pourra trouver la valeur de N optimal. La figure 7 décrit l'organigramme de l'algorithme d'échange de Remez.



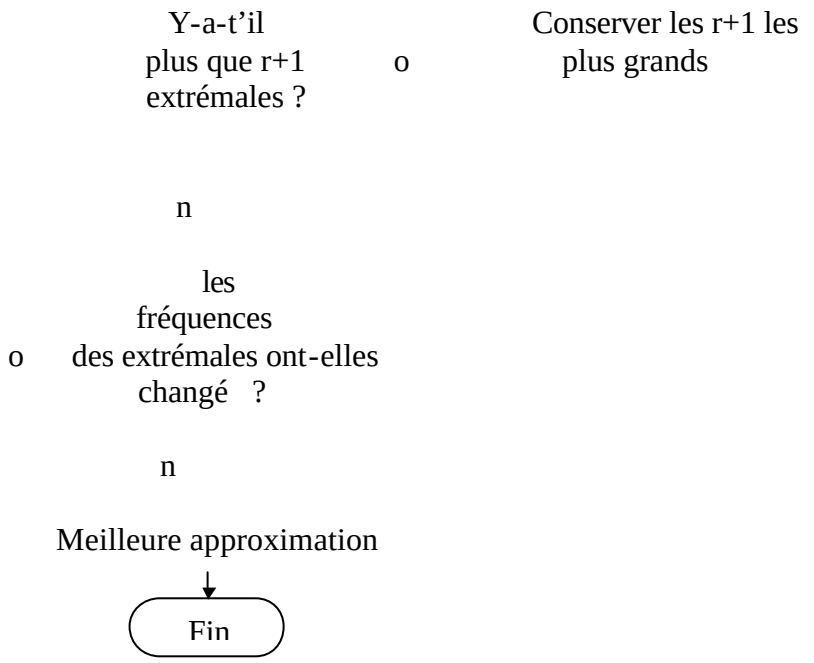


Fig. 11 Organigramme de l'algorithme d'échange de Remez

Cette organigramme présente l'approche utilisée par l'algorithme d'échange de Remez pour obtenir une solution au problème d'approximation.

Le filtre de PARKS-McCLELLAN (PMC) sous l'algorithme d'échange de Remez présente la meilleure approximation au sens de Chebyshev pour les coefficients à précision infinie. Il peut concevoir une large variété de filtres standards pour n'importe quelle réponse en amplitude désirée spécifiée par l'utilisateur. La rapidité de cet algorithme lui offre l'intérêt d'un large domaine d'application. Mais, en implantant ces coefficients dans un processeur de signaux à une longueur de mot limitée, nous allons induire une erreur qui est due à l'adaptation entre la taille (la longueur du mot en bits) des coefficients et le mot machine. Cette erreur est due à la quantification. Par conséquent, les filtres conçus dans l'espace discret ne présente pas la meilleure approximation

A ce propos, il existe dans la littérature plusieurs méthodes qui ont été mises en œuvre dans l'espace discret afin de remédier aux erreurs sur la réponse en amplitude, dues à la contrainte de la limitation de la longueur du mot [9].

Notre travail de base s'inscrit dans le même contexte et le filtre de P.M.C. avec des coefficients à précision infinie sera notre filtre de référence. Dans le paragraphe suivant, nous présentons les types des erreurs dues à la mise en œuvre des coefficients de P.M.C. dans un mot de longueur finie.

III.5. ERREUR INHERENTE A LA MISE EN ŒUVRE DES FILTRES R.I.F. A PHASE LINEAIRE SUR MACHINE :

Dans ce qui suit, nous exprimons les conséquences de la limitation de la longueur du mot machine à Nb_{ord} bits sur les coefficients des filtres RIF à phase linéaire.

Considérons un filtre RIF à phase linéaire d'ordre N-1 où sa réponse en fréquence s'exprime sous la forme (29). Il a été montré que la réponse en amplitude pour les quatre cas des filtres s'écrive sous la forme

$$P_n(e^{jw}) = \sum_{n=0}^{k-1} a(n)\cos wn \qquad (35)$$

Où k est le nombre de termes (= N/2, (N-1)/2 ou (N+1)/2) et a(n) relative à h(n) est la séquence résultante décalée dépendante du cas considéré.
Dans la méthode classique où les coefficients sont calculés par ordinateur puis quantifiés, il existe plusieurs types d'erreurs :

a- Erreur due à la limitation du mot ordinateur.
b- Erreur de quantification des valeurs des coefficients du mot ordinateur au mot processeur.
c- Erreur de la représentation.

III-5-1 ERREUR DUE A LA LIMITATION DU MOT MACHINE :

Théoriquement, les opérations arithmétiques imposées sur les séquences a(n) de la réponse en amplitude $P_n(e^{jw})$ influent sur la longueur binaire des coefficients. Si on procède par exemple, à la multiplication de deux coefficients à b bits, le résultat sera représenté sur 2.b bits. En réalité, ce fait ne coincide pas avec la longueur du mot machine disponible constante qui ne croît pas.
Cette limitation se traduit par la superposition d'une erreur $\varepsilon_{ord}(n)$ sur a(n). La séquence obtenue $a_{ord}(n)$ diffère en précision de la séquence théorique a(n).
Nous avons donc :

$$a_{ord}(n) = a(n) + \varepsilon_{ord}(n) \qquad (36)$$

$a_{ord}(n)$: séquence à lm bits calculée par ordinateur.
a(n) : séquence en précision infinie (séquence théorique).
$\varepsilon_{ord}(n)$: erreur sur la séquence due à la limitation du nombre de bits de l'ordinateur à lm bits.

III-5-2 ERREUR DE QUANTIFICATION :

L'implantation des coefficients sur un processeur de signaux exige une adaptation de la taille du coefficient calculé par ordinateur à celle du processeur qui est généralement plus petite. Cette adaptation se traduit par une quantification des coefficients qui induit une erreur sur la séquence a(n) nommée erreur de quantification $\varepsilon_{proc}(n)$
La séquence réellement implantée est :

$$a_{proc}(n) = a_{ord}(n) + \varepsilon_{proc}(n).$$
$$a_{proc}(n) = a(n) + \varepsilon_{ord}(n) + \varepsilon_{proc}(n).$$

on pose $\varepsilon_{tot}(n) = \varepsilon_{ord}(n) + \varepsilon_{proc}(n).$

on aura $$a_{proc}(n) = a(n) + \varepsilon_{tot}(n) \qquad (37)$$

$a_{proc}(n)$ = séquence du filtre implanté sur le processeur à lmp bits.
$a_{ord}(n)$: séquence à lm bits calculée par ordinateur.
a(n) : séquence en précision infinie (séquence théorique).
$\varepsilon_{ord}(n)$: erreur sur la séquence due à la limitation du nombre de bits de l'ordinateur à lm bits.

$\varepsilon_{proc}(n)$: erreur sur la séquence due à la quantification des coefficients sur un processeur de nombre de bits de lmp bits.
$\varepsilon_{tot}(n)$: erreur totale sur la séquence due à la limitation du nombre de bits de l'ordinateur et la quantification des coefficient au nombre de bits du processeur.

La réponse en amplitude du filtre réellement implanté sur un processeur de signaux devient:

$$P_{nproc}(e^{jw}) = \sum_{n=0}^{k-1} a_{proc}(n)\ \cos w$$
$$= \sum_{n=0}^{k-1} (a_{ord}(n) + \varepsilon_{ord}(n) + \varepsilon_{proc}(n)).\cos wn$$
$$= P_n(e^{jw}) + E_{ord}(e^{jw}) + E_{proc}(e^{jw})$$

on pose $E_{tot}(e^{jw}) = E_{ord}(e^{jw}) + E_{proc}(e^{jw})$

on aura $P_{nproc}(e^{jw}) = P_n(e^{jw}) + E_{tot}(e^{jw}).$ (38)

où
$E_{ord}(e^{jw})$: erreur d'amplitude due à la limitation du nombre de bits d'ordinateur.
$E_{proc}(e^{jw})$: erreur d'amplitude due à la quantification des coefficients.

III-5-3- ERREUR DUE A LA REPRESENTATION :

Dans le but d'exprimer des nombres sur ordinateur, les représentations binaires les plus fréquentes à utiliser sont la représentation binaire à virgule fixe et la représentation binaire à virgule flottante. Comme déjà présenté dans le paragraphe III.1.4, l'expression des coefficients du filtre diffère d'une représentation à une autre. Cela est relatif aux ressources disponibles le temps de calcul et la précision. Vu le nombre de valeurs admissibles pour un mot de longueur donnée, nous pouvons affirmer que l'utilisation des représentations binaires à virgule fixe et flottante ont la plus grande probabilité d'avoir une erreur faible par rapport à l'utilisation de la représentation SDPD, malgré que cette dernière présente le pas minimal le plus petit.
Dans le chapitre II suivant, une étude comparative expérimentale est faite afin de mettre en évidence le choix de la représentation binaire adéquate suivant les spécifications requises la précision et le temps de calcul.
Nous exposons aussi les axes de recherche de notre travail de base concernant la synthèse des filtres dans l'espace discret des coefficients après une présentation des différents travaux reportés dans la littérature scientifique.

IV. ETAT DE L'ART 'CONCEPTION DE FILTRES NUMERIQUES':

IV.1. INTRODUCTION :

Un des problèmes fondamentaux du traitement numérique du signal est la conception des filtres numériques qui consiste à calculer les coefficients h(n) de telle manière que la représentation en fréquence $H(e^{jw})$ approche au mieux la réponse en fréquence désirée $H_D(e^{jw})$. La grande partie des travaux existants qui permettent l'approximation de ce type de filtre numérique concerne beaucoup plus les filtres à réponse impulsionnelle finie R.I.F., que les filtres à réponse impulsionnelle infinie

R.I.I., à raison de leur stabilité, possibilité de linéarité de la phase et formulation mathématique simple.

Les méthodes existantes pour la conception de filtres RII dans un espace discret sont en nombre limité vu la complexité mathématique de ce type de filtre. Tandis que la méthode d'approximation des filtres numériques RIF à phase linéaire la plus connue et intensivement utilisée est celle de Parks-McClellan[9]. Comme nous l'avons montré dans le paragraphe III.4., cette méthode utilise des ordinateurs qui procurent une grande précision. Les coefficients calculés sont généralement donnés sur plus de 8 chiffres significatifs. Cependant, lors de la mise en œuvre de tels filtres sur des processeurs de signaux, les coefficients doivent être stockés dans des mots limités en longueur. La longueur du mot est plus courte et donc, nous sommes amenés à effectuer une quantification (troncature ou arrondissement) des coefficients. Cette quantification engendre des distorsions souvent prohibitifs sur les réponses en fréquence des filtres mis en œuvre. L'intérêt de notre travail consiste à réduire l'effet de cette distorsion. Pour cela, nous proposerons des méthodes qui permettent la synthèse des filtres directement dans l'espace discret des coefficients défini par des valeurs admissibles.

Nous allons dans ce qui suit donner un historique des méthodes existantes dans le domaine de conception de filtres numériques RIF à phase linéaire, puis nous exposons la contribution du laboratoire « Signaux et Systèmes » envers ce type de problème.

IV.2. HISTORIQUE DES METHODES EXISTANTES :

Pour la synthèse des filtres numériques à coefficients discrets, il est souvent désirable d'utiliser des algorithmes dont la qualité de sortie peut être ajustée suivant les ressources disponibles telles que le temps de calcul et la précision.

L'algorithme d'échange de Remez [9]a été conçu pour l'optimisation des coefficients des filtres RIF à précision infinie. Dans le cas de filtres dans un mot de longueur finie, le problème d'optimisation devient plus complexe, où une investigation générale de la solution optimale exige un temps de calcul prohibitif. Afin de remédier à ce problème, plusieurs méthodes d'optimisation ont été appliquées pour la conception des filtres numériques à coefficients discrets. La technique du gradient simulée 'Simulated Annealing Technique' [2]-[4] a prouvé à être efficace dans plusieurs cas, mais exige un grand nombre de fonctions d'évaluations impliquant un temps de calcul très long qui dépend des températures de départ(méthode de METROPOLIS)[4]. La fonction d'évaluation est définie comme étant l'ensemble d'opérations nécessaires pour traiter un arrangement à la fois. L'arrangement est un cas de combinaison ordonnée de valeurs discrètes constituant les coefficients du filtre à concevoir

La formule de programmation linéaire en nombre entier[1],[10],[11],[13] a été appliquée comme une méthode d'optimisation discrète dans le sens minmax. Malgré, qu'il soit possible d'obtenir des résultats optimaux, le temps de calcul nécessaire même avec les super ordinateurs actuels, prohibe l'application de ces techniques pour des filtres à ordre élevé.

Les techniques d'optimisation dans l'espace discret des coefficients, et en particulier la méthode de recherche arborescente, ont été élaborées afin de remédier à ce problème d'optimum discret. Ces méthodes basées sur des techniques d'énumérations implicites nécessitent un temps de calcul très élevé[10],[11],[14],[31],[32].

Dans plusieurs méthodes de recherche locales basées sur la méthode de recherche arborescente, telles que 'la méthode de recherche en profondeur d'abord' et 'la méthode de séparation et d'évaluation progressive' (SEP), les solutions retrouvées sont meilleures que celles obtenues à partir d'une directe quantification. L'objectif majeur de ces méthodes est la détermination de stratégies de branchement et de syntonisation[14],[31],[32].
L'utilisation du critère d'erreur dépend de l'application de telles techniques. Dans la littérature, le critère d'erreur le plus fréquent est le critère minmax qui a prouvé une efficacité considérable pour les méthodes d'optimisation[3],[4],[7]-[9],[11]. En utilisant le critère d'erreur quadratique moyenne, il a été montré qu'il est possible de prédire une amélioration lorsque la programmation quadratique en nombre entier est utilisée[14],[15].
Nous allons présenter dans le paragraphe suivant les contributions du laboratoire signaux et systèmes envers le problème de synthèse de filtres numériques RIF à phase linéaire dans l'espace discret des coefficients.

IV.3. METHODES DU LABORATOIRE ' SIGNAUX ET SYSTEMES' (BOULERIAL) :

IV.3.1 METHODE DE RECHERCHE ARBORESCENTE :

La méthode de recherche arborescente effectue une recherche exhaustive du meilleur filtre dans le sens de l'erreur quadratique moyenne directement dans l'espace discret des coefficients.

Cette méthode présente l'avantage de pouvoir extraire le filtre qui présente la meilleure approximation au sens de l'erreur quadratique moyenne. Cependant elle nécessite une très grande quantité de calculs et devient très vite inexploitable quand l'ordre N-1 du filtre et la longueur du mot Nb bits augmentent.(N=8 et Nb =8, le calcul est de l'ordre de 26 heures sur un PC de fréquence CPU 300 Mhz).

la principale contribution du laboratoire signaux et systèmes dans ce contexte a été de réduire l'espace discret de recherche pour améliorer la convergence de l'algorithme. Pour cela, il a été utilisé la représentation ''SDPD''. Celle-ci possède un nombre de valeurs admissibles plus petit par rapport celui de la représentation en virgule fixe et en virgule flottante pour une même longueur de mot. On a aussi pu diminuer le nombre de fonctions d'évaluation en ignorant les arrangements de valeurs admissibles qui ne vérifient pas les conditions obtenues à partir des propriétés des filtres RIF à phase linéaire. Ces conditions sont des tests simples sur les coefficients, tel que la somme des coefficients, qui sont utilisés avant toute investigation, cela a permis de connaître à l'avance si la solution est admissible ou pas. Il a été montré que toutes ces simplifications permettent de réduire l'espace de recherche des solutions et le temps de calcul.

L'algorithme a été élaboré en utilisant le critère de l'erreur quadratique moyenne. Ensuite, un critère d'erreur a été mis en œuvre intitulé 'erreur min-max parmi les filtres à erreur quadratique moyenne la plus faible'. Ce critère combine entre les avantages des deux critères critère de l'erreur quadratique moyenne et critère de l'erreur min-max. le choix de tels critères dépend des spécifications requises du filtre à concevoir.

Il a été montré que la méthode délivre un filtre qui présente la meilleure approximation au sens de l'erreur quadratique moyenne. Cependant, la convergence

de cet algorithme est très lente. On a pu accélérer la vitesse de convergence de cet algorithme en réduisant l'espace de recherche et en injectant dans l'algorithme des conditions simplifiant largement la recherche de la meilleure solution, mais le temps de calcul reste prohibitif [32].

A ce propos, il a été proposé une deuxième méthode de synthèse de filtre dans le sens de l'erreur quadratique moyenne plus rapide que la méthode de recherche arborescente, présentée dans le paragraphe suivant.

IV.3.2. METHODE DIRECTE PAR OPTIMISATION AU SENS DES MOINDRES CARRES (D.M.C.) [32]:

On suppose souvent que les erreurs de quantification des coefficients sont indépendantes. Cette hypothèse n'implique pas que les erreurs d'amplitude de la réponse en fréquence sont indépendantes. Si on modifie la valeur d'un coefficient h(n) l'amplitude de la réponse en fréquence, à une fréquence donnée, suivra ces variations. Par conséquent on peut compenser l'erreur d'amplitude due à la quantification d'un coefficient h(m), par une variation adéquate d'un autre coefficient h(n) non encore quantifié.

Dans ce contexte est élaborée la méthode D.M.C. qui est entièrement originale[32]. Elle procède par optimisation séquentielle des coefficients, en utilisant la méthode des moindres carrés. La recherche se fait de telle manière que l'erreur due à la limitation de la longueur du mot processeur est directement considérée et optimisée dans l'algorithme. Il a été donc très intéressant de rechercher les coefficients séquentiellement l'un à la suite de l'autre, en tenant compte à chaque fois de l'erreur d'amplitude de la réponse en fréquence due à la quantification du coefficient précédemment déterminé.

Il a été montré que D.M.C. a une convergence rapide et délivre des filtres performants. Les temps de synthèse relevés sont équivalents à ceux de la synthèse de Parks et Mac Clellan sous l'algorithme d'échange de Remez.

Dans cette méthode la solution finale dépend du choix du premier coefficient à déterminer ainsi que de l'ordre dans lequel sont ensuite considérés les autres coefficients[32].

Malgré que la convergence soit très rapide, les résultats de la méthode de recherche arborescente ne sont pas assurés. A ce propos, il a été élaboré une troisième méthode dans le paragraphe suivant qui combine entre les avantages des deux méthodes, les performances de la méthode de recherche arborescente et la rapidité de convergence de la méthode DMC.

IV.3.3. METHODE DE SYNTHESE PAR SEPARATION ET EVALUATION PROGRESSIVE (SEP) :

La difficulté principale dans la méthode de recherche arborescente [33] est le temps de calcul, prohibitif pour les filtres d'ordre N-1 supérieur à 8, il est par conséquent nécessaire de réduire la quantité de calcul .

En effet, la méthode de recherche arborescente consiste à effectuer une auscultation exhaustive de tous les cas de solutions possibles dans l'espace discret des solutions E_{proc}^{N}. Malgré les tests simples apporté dans[32] et qui ont permis d'écarter un nombre important de solutions non admissibles, la quantité de calcul reste prohibitive.

Parmi les solutions à écarter ou à ne pas prendre en compte, il y a deux types :

- les solutions non acceptables et qu'on peut facilement écarter par les tests simples.
- les sous ensembles de solutions acceptables et qui ne peuvent contenir la meilleure solution à rechercher: c'est sur ce point que s'appuie la méthode S.E.P.

Pour résoudre effectivement par calculateur les problèmes de la synthèse de filtre R.I.F. d'ordre > 50, il faut donc rechercher un principe algorithmique. Cet algorithme doit nous permettre de connaître, sans avoir à tester si chaque solution est acceptable ou pas, l'existence ou non d'une meilleure solution parmi un sous-ensemble de solutions acceptables. L'idée de base de la méthode SEP est l'hybridation entre les deux méthodes ' méthode de recherche arborescente et la méthode DMC'. Il a été jugé intéressant de réunir ces deux méthodes en tirant profit de leur avantage respectif, en d'autres termes de concevoir une méthode qui exploite d'une part la garantie des performances de la méthode de recherche arborescente et d'autre part la vitesse de convergence de D.M.C. [32].

La méthode S.E.P. se base sur la réduction de l'espace discret de recherche et cibler les régions susceptibles de contenir la meilleure solution en utilisant la méthode D.M.C. modifiée. Ensuite, la méthode de recherche arborescente a été utilisée pour la recherche des solutions dans la régions ainsi définies.

Malgré que cette méthode a procuré des résultats satisfaisants, mais les performances ne sont pas garanties à être égales à celle de la méthode de recherche arborescente, et même le temps de calcul est beaucoup plus grand que celui de la méthode DMC. La combinaison entre les avantages des deux méthodes recherche arborescente et DMC peut être néfaste sur les résultats finaux si l'évaluation n'est pas faite convenablement. Par conséquent, cette méthode S.E.P. n'est pas prise en considération dans la suite de ce travail, contrairement aux deux méthodes recherche arborescente et DMC.

Dans le paragraphe suivant, nous allons exposer les problèmes liés à la synthèse des filtres numériques dans l'espace discret des coefficients avant de se prononcer pour les méthodes de base utilisées dans la suite de ce travail.

V. POSITION DU PROBLEME :

V.1. INTRODUCTION :

Lorsqu'un filtre numérique est implanté sur un processeur de signaux de longueur de mot 'lm' bits, chaque coefficient du filtre doit être représenté donc, par un nombre fini 'lm' bits. L'approche habituellement utilisée consiste à quantifier les coefficients à grande précision obtenus dans le cas des filtres R.I.F. à phase linéaire par Parks-Mc Clellan [9]. Les filtres ainsi implantés perdent leur optimalité, ils existent d'autres coefficients de même longueur de mot finie qui donnent une meilleure approximation au sens Chebyshev par rapport à la réponse en fréquence désirée. Afin de retrouver ces coefficients, il est nécessaire donc d'inclure la limitation de la longueur de mot dans la procédure de la conception de filtre.

Pour concevoir des filtres numériques à coefficients de longueur de mot finie, il est souvent désirable d'utiliser des algorithmes

de recherche dans l'espace discret des coefficients dont la qualité de sortie peut être ajustée suivant la disponibilité des ressources telles que, la précision et le temps de calcul.

V.2. FORMULATION DU PROBLEME :

D'une manière générale, toutes les méthodes de conception de filtres existantes utilisent un ordinateur pour calculer les coefficients du filtre désiré puis, les quantifier suivant la longueur du mot processeur. Notre propos consiste à calculer les coefficients du filtre directement dans l'espace E_{proc}.

En d'autres termes il s'agit de trouver les meilleurs coefficients du filtre désiré directement dans l'espace E_{proc}. Plusieurs méthodes seront proposées dans ce mémoire, la plupart sont des améliorations des travaux du laboratoire signaux et systèmes, et nous verrons que nous pourrons accélérer la convergence des algorithmes classiques lents, comme nous pourrons améliorer les résultats relatifs aux algorithmes à faibles performances.

Avant d'introduire la première méthode, nous allons présenter et justifier tout d'abord le choix, que nous avons fait, de l'espace de définition et de la représentation des valeurs des coefficients.

V.3. CHOIX DE l'ESPACE DE DEFINITION, DES METHODES ET DES REPRESENTATIONS ADEQUATS :

V.3.1 NORMALISATION DE L'ENSEMBLE DE DEFINITION :

L'ensemble de définition des coefficients doit être nécessairement limité pour les raisons suivantes :

a) En traitement numérique du signal, on préfère le plus souvent utiliser la représentation fractionnaire (les nombres compris entre -1 et +1). Son intérêt est de pouvoir majorer tous les nombres par 1 et donc aussi leurs produits.

b) Si nous considérons la réponse en fréquence d'un filtre R.I.F, donnée par Eq. 17. On peut multiplier et diviser tous les coefficients h(n) par un même nombre réel positif α. Ce coefficient est appelé facteur d'échelle. Nous obtenons :

$$H(e^{jw}) = \sum_{n=0}^{N-1} \alpha \frac{h(n)}{\alpha} e^{-jnw} = \alpha \sum_{n=0}^{N-1} \frac{h(n)}{\alpha} e^{-jnw}$$

si on choisit α tel que $\alpha \geq |h(n)| \ \forall$ n, nous pouvons remplacer :

$h(n)/\alpha = h^*(n)$

h*(n) est appelé coefficient normalisé, les valeurs prises par ce coefficient sont dans l'intervalle [-1, 1].

$$H(e^{jw}) = \alpha \sum_{n=0}^{N-1} h^*(n) e^{-jnw} = \alpha H^*(e^{jw}) \qquad (39)$$

La multiplication des coefficients par un facteur d'échelle α [17], se traduit par une modification du gain du filtre, mais n'affecte pas la forme de la réponse en fréquence. Le gain du filtre étant spécifié avec une certaine tolérance à une fréquence donnée, il faut simplement s'assurer que la représentation binaire du facteur d'échelle α permet de satisfaire cette contrainte.

Ainsi on peut toujours retrouver le filtre dont le gain est différent de 1, à partir du filtre obtenu avec les coefficients normalisés.

Nous avons choisi dans notre travail de limiter l'ensemble de définition des coefficients à l'intervalle [-1,1].[32]

V.3.2. CHOIX DE LA METHODE :

Notre travail de recherche se subdivise en deux axes de recherches.

- le premier est basé sur la méthode de recherche arborescente [11], [14], [32]. Nous allons procéder par une nouvelle stratégie de branchement afin de réduire le temps de calcul tout en maintenant les performances de la méthode de recherche arborescente. L'approche utilisée est nommée la méthode de Recherche Séquentielle et Progressive 'R.S.P.'.

- Le deuxième axe de recherche est élaboré autour de la méthode 'Directe à Moindre Carré' D.M.C.. Dans cette section, nous allons introduire une méthode itérative qui permet d'améliorer les performances de la méthode DMC en exploitant sa rapidité de convergence. Cette nouvelle méthode est nommée méthode Directe à Moindre Carré Itérative 'D.M.C.I.'.

V.3.2. CHOIX DE LA REPRESENTATION :

Avant de se prononcer pour une représentation adéquate entre la représentation binaire en virgule fixe, la représentation binaire en virgule flottante et la représentation S.D.P.D, une étude statistique comparative sera menée dans le chapitre 2. Dans ce chapitre, les méthodes de synthèse utilisées sont la méthode de recherche arborescente et la méthode DMC. La comparaison entre ces trois représentations binaires est purement pratique se basant sur plusieurs exemples de filtres. Pour une comparaison significative, nous avons testé ces méthodes sur plusieurs centaines de filtres dont nous avons choisi de présenter quatre dans le chapitre 2. Les conséquences de la comparaison restent relatives au cas de filtre considéré et très importantes pour la suite de notre travail de recherche.

VI. CONCLUSION :

Dans ce chapitre, nous avons donné les notions fondamentales permettant une bonne compréhension des caractéristiques d'un espace discret, notre domaine de définition.

Nous avons présenté aussi un rappel sur les représentations binaires des nombres, l'effet de la limitation de la longueur des coefficient et les méthodes de synthèse de filtre R.I.F. à phase linéaire et spécialement les filtres de Parks – Mc Clellan.

Nous avons exposé les méthodes de synthèse de filtres numériques dans l 'espace discret existantes dans la littérature scientifique. Enfin, nous avons défini les deux grands axes de recherche de ce travail qui sont la méthode de recherche arborescente et la méthode directe par moindre carré.

Chapitre II.

PERFORMANCES DES ALGORITHMES RA ET DMC EN FONCTION DE LA REPRESENTATION ET DU CRITERE CHOISI

XII. INTRODUCTION :

Pour des raisons de complexité algorithmique, le laboratoire Signaux et Systèmes [32] a utilisé la représentation SDPD concernant la synthèse des filtres RIF à phase linéaire dans l'espace discret des coefficients. Notre propos dans ce chapitre, est d'étendre les travaux de celui-ci pour les représentations en virgule fixe et en virgule flottante afin de voir la perte de performance comparativement au gain de complexité découlant de l'utilisation de la représentation SDPD. En effet, comme nous l'avons montré dans le chapitre précédent, les représentations en virgule fixe et en virgule flottante offrent un nombre de valeurs admissibles plus grand que celui de la représentation SDPD pour une même longueur de mot. Les méthodes de base utilisées dans ce chapitre et dans la suite de notre travail, sont la méthode de recherche arborescente RA et la méthode DMC pour leur avantage, respectivement, leurs performances et rapidité de convergence.

Dans ce chapitre, nous présentons aussi les critères d'approximation utilisés dans la suite de notre travail, l'erreur quadratique moyenne et l'erreur de Chebyshev ou 'minmax'. Ensuite, nous effectuons une étude comparative des représentations en virgule fixe en virgule flottante et en SDPD en utilisant la méthode de recherche arborescente et la méthode DMC.

XIII. CRITERES D'APPROXIMATION :

II.1. L'ERREUR QUADRATIQUE MOYENNE 'Ems' :

L'écart qui existe entre la courbe d'amplitude approchée de la réponse en fréquence $H(e^{jw})$ et celle du filtre idéal $H_D(e^{jw})$ est évalué en prenant la moyenne

quadratique des erreurs déterminées sur un intervalle fréquentiel, à l'intérieur des bandes (passante et atténuée).

L'erreur non pondérée en fonction de la fréquence est définie par :

$$E(e^{jw}) = H_D(e^{jw}) - H(e^{jw}) \qquad (40)$$

où $H_D(e^{jw})$: est la réponse en fréquence désirée.

$H(e^{jw})$: est la réponse en fréquence approchée.

L'erreur quadratique moyenne Ems est définie par :

$$Ems = \left\{ \frac{1}{\pi} \int_0^{\pi} E(e^{jw})\, E(e^{jw})^* dw \right\}^{\frac{1}{2}}$$

Avec $E(e^{jw})^*$: conjugué de $E(e^{jw})$.

et d'une manière plus explicite

$$Ems = \left\{ \frac{1}{\pi} \int_0^{\pi} \left| H_D(e^{jw}) - H(e^{jw}) \right|^2 dw \right\}^{\frac{1}{2}} \qquad (41)$$

Dans les travaux du laboratoire Signaux et Systèmes [32] concernant la synthèse des filtres dans l'espace discret des coefficients, on a approché Ems en calculant la moyenne des erreurs quadratiques sur 4N points de fréquences dans l'intervalle [0 π]. (N: étant la longueur du filtre). Pour une comparaison plus significative, nous avons gardé la même démarche.

$$Ems \approx \left\{ \frac{1}{4N} \sum_{i=0}^{4N-1} \left(\left| H_D(e^{jw_i}) \right| - \left| H(e^{jw_i}) \right| \right)^2 \right\}^{\frac{1}{2}} \qquad (42)$$

où $i = 0,1,2, \ldots ,4N-1.$

Si nous considérons le filtre passe-bas idéal avec :

$$|H_D(e^{jw})| = \begin{cases} 1 & \text{pour } w_i \in \text{B.P. (Bande Passante)} \\ 0 & \text{pour } w_i \in \text{B.A. (Bande Atténuée)} \end{cases}$$

L'expression de Ems devient alors :

$$Ems = \left\{ \frac{1}{4N} \sum_{i=0}^{K-1} \left| 1 - \left| H(e^{jw_i}) \right| \right|^2 + \sum_{i=K}^{4N-1} \left| H(e^{jw_i}) \right|^2 \right\}^{\frac{1}{2}} \qquad (43)$$

La valeur de K, exprime le nombre de points pris dans la bande passante. K est choisi tels que les pas ($\Delta w = w_i - w_{i-1}$) dans la bande passante et dans la bande atténuée soient sensiblement égaux. On ne considère aucun point dans la bande de transition, la différence (4N-K) est le nombre de points pris dans la bande atténuée.

II.2. L'ERREUR DE CHEBYSHEV OU MIN-MAX 'Emm' :

Cette erreur est définie comme suit:

$$Emm = \min. \{ \max. W(wi).||H_D(e^{jwi})| - |H(e^{jwi})|| \}_{wi \in B.P. \cup B.A.}. \qquad (44)$$

Avec W(wi) : la fonction poids réelle et positive définie sur l'union des fréquences 'wi' dans la B.P. et la B.A.

Dans le cas d'un filtre passe bas, nous avons :

$$Emm = \min. \{ \max. W(wi).[|1- |H(e^{jwi})||_{wi \in B.P.}, |H(e^{jwi})|_{wi \in B.A.}] \}. \qquad (45)$$

où $i = 0,1,2, \ldots, 4N - 1.$

Cette définition est applicable lorsque les erreurs dans les bandes (passante et atténuée) sont considérées avec n'importe quelle fonction poids.

Dans la suite de notre travail, nous allons noter comme :

Erreur minmax dans la B.P. $Emp = \max. |1- |H(e^{jwi})||_{wi \in B.P.}$ (46.a.)

Erreur minmax dans la B.A. $Ema = \max. |H(e^{jwi})|_{wi \in B.A.}$ (46.b.)

Ces deux critères d'erreurs seront utilisés dans la suite de ce chapitre dans l'évaluation du filtre.

Le critère de base utilisé est l'erreur quadratique moyenne. Les erreurs minmax dans la B.P. et la B.A. ne seront prises en considération qu'en cas d'égalité de l'erreur Ems entre deux filtres.

XIV. ALGORITHME RA DANS LES TROIS REPRESENTATIONS :

Dans ce paragraphe, nous allons tout d'abord présenter la méthode l'algorithme de recherche arborescente 'RA' tels qu'ils sont décrits dans [10]et [31]. Ensuite, nous donnerons un ensemble de filtres avec lequel la méthode RA a été testée dans les trois représentations binaires en virgule fixe, en virgule flottante et en SDPD. Enfin, une étude des résultats obtenus sera donnée, avec une comparaison de l'utilisation de la représentation en se référant aux travaux de [31].

III.1. DESCRIPTION DE LA METHODE :

La technique utilisant la méthode de recherche arborescente que nous notons par 'R.A.', est une méthode simple qui consiste à parcourir l'espace discret des solutions E_{proc}^{N} suivant l'ordre des arrangements établi et tester tous les cas de solutions possibles de cet espace. Pour chaque cas de solution, l'erreur sur l'amplitude de la réponse en fréquence $H(e^{jw})$ est calculée, ensuite comparée dans le sens de Ems avec celle du meilleur filtre obtenu. A la fin de la recherche, lorsque tous les cas de solutions sont testés, la solution retenue sera celle qui présente la meilleure approximation dans le sens de Ems. Pour respecter la contrainte de la linéarité de la phase, les coefficients sont choisis soit symétriques ou antisymétriques.

Nous avons utilisé dans le programme une variable de comparaison E_{min}. Elle est initialisée à la première valeur de l'erreur trouvée. Cette variable conserve l'erreur provisoire du filtre qui a la meilleure approximation dans le sens de Ems, elle est utilisée au cours de la recherche pour comparer le filtre courant. Dans la méthode R.A., le critère de l'erreur choisi est le critère de l'erreur quadratique moyenne (Ems), par conséquent, E_{min} est du même type que Ems.

Pour rappeler les différentes étapes qui composent notre algorithme nous utilisons l'exemple suivant :

Soit la recherche d'un filtre de longueur deux (deux coefficients h(0), h(1)) dans un espace des solutions discrètes E_{proc}^{2} où les valeurs permises sont les valeurs appartenant à un ensemble E_{proc} = [v0, v1, v2, v3, v4]. L'espace des solutions à E_{proc}^{2}, est donc constitué de 5^2=25 vecteurs [vi vj] i et j $\in$ [0, 1, ... 4]. Chaque cas de solution se compose de deux valeurs discrètes écrites dans une représentation binaire choisie, prises dans l'ensemble discret E_{proc} composé de 25 solutions. L'optimisation des coefficients de ce filtre par la méthode proposée, demande les opérations suivantes :

a) Affecter aux coefficients h(1) et h(0), suivant l'ordre des arrangements, des valeurs de l'ensemble discret E_{proc} autrement dit prendre un cas de solution de l'espace E_{proc}^{2}. Au début, le vecteur [h(1) h(0)] est pris égal à [v4 v4].

b) Déterminer la courbe d'amplitude correspondante :

- Si la courbe d'amplitude n'entre pas dans le gabarit du filtre, la solution est rejetée. Nous allons au point (d).

- Si la courbe d'amplitude est contenue à l'intérieur du gabarit. Nous allons au point (c).

c) Evaluer l'erreur Ems.

- Si l'erreur Ems est plus grande que la valeur de comparaison E_{min}. La solution n'est pas prise en compte. On va au point (d).
- Si l'erreur Ems est inférieure à E_{min}, la solution est retenue provisoirement comme bonne. Nous sauvegardons la solution et, nous donnons à la variable E_{min} la valeur de l'erreur calculée Ems. Nous allons au point (d).

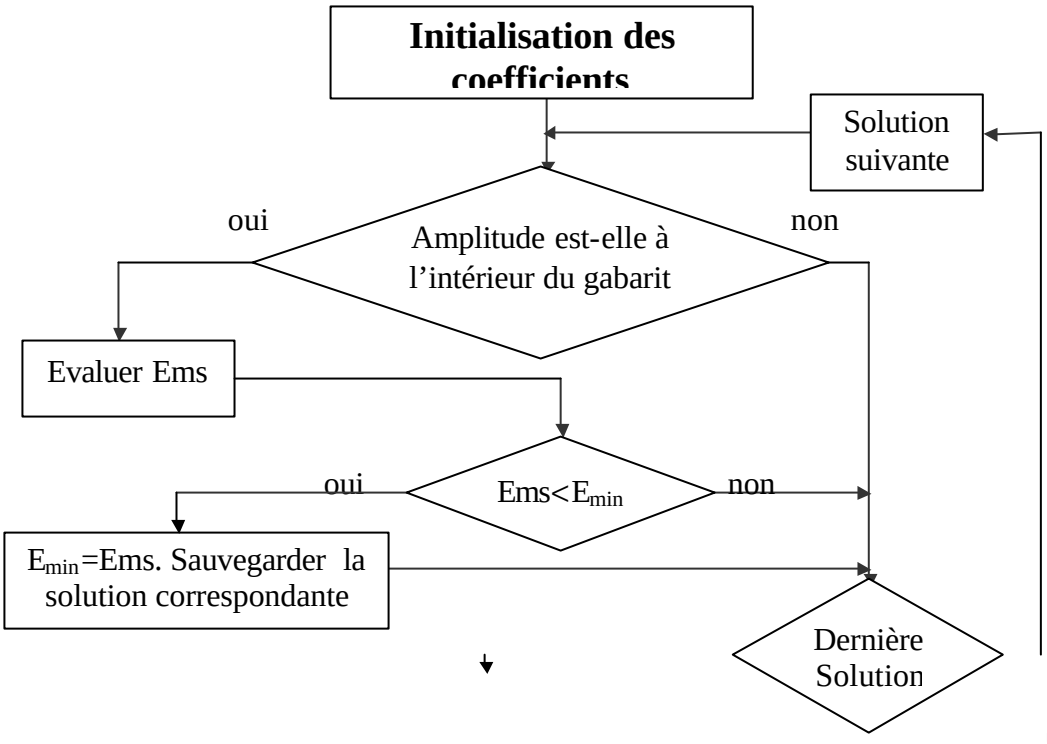
d) Tester si toutes les solutions ont été considérées (en testant si la solution courante est égale à [v0 v0]). Sinon nous revenons au point (a). [v0 v0] est la dernière solution qui sera vérifiée.

A la fin de la recherche, si, et c'est un cas très probable, la solution optimale du filtre est celle qui correspond à [v4 v4] alors, le programme va sortir en ne trouvant aucune solution. Alors, la valeur de comparaison E_{min} ne changerait pas de sa première valeur qui correspond dans l'exemple précédent à la valeur Ems de l'arrangement [v4 v4]. Si la valeur de comparaison E_{min} a diminué, alors, une solution a été trouvée et elle correspond au filtre qui présente la meilleure approximation dans le sens de Ems dans l'espace discret E^2_{proc}. Ces opérations restent évidemment valables quelque soit le filtre et sa ordre.

Nous présenterons dans le paragraphe suivant, un organigramme qui contient les démarches de la méthode R.A. prises en considération dans la suite de notre travail.

III.2. ORGANIGRAMME :

Cet organigramme regroupe les étapes précédemment décrites. Nous notons par Ems(1), l'erreur quadratique moyenne du premier arrangement considéré.



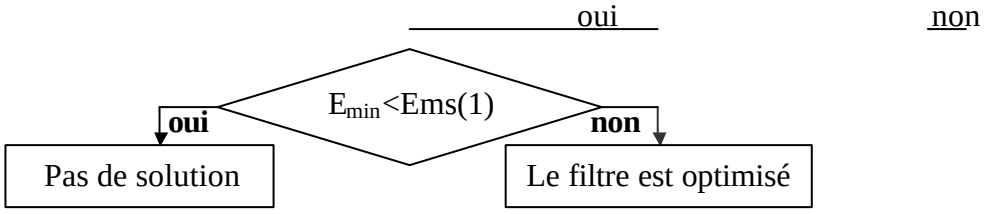


Fig. 12: Organigramme de Synthèse par la méthode de recherche arborescente.

Cette méthode par la recherche arborescente sera utilisée dans ce chapitre (dans le paragraphe IV) dans les différentes représentations binaires citées auparavant dans le sens de la Ems. Pour chaque filtre considéré, nous avons relevé les erreurs Emp et Ema. Les résultats trouvés sont donnés dans le paragraphe suivant.

XV. RESULTATS DE LA SYNTHESE PAR LA METHODE DE RECHERCHE ARBORESCENTE 'R.A.' DANS LE SENS DE EMS

La méthode R.A. a été testée sur plusieurs centaines de filtres dont nous reportons quatre. Ces 4 filtres (repérés par les numéros de 1 à 4) sont des filtres de type RIF à phase linéaire passe bas, conçus avec la méthode R.A. dans le sens de Ems. Pour mieux tester ces capacités, nous avons choisi des filtres avec des spécifications différentes (bande passante large ou étroite, avec bande de transition large ou étroite). Nous nous sommes aussi intéressés, pour chaque filtre, à étudier l'évolution de ses coefficients en fonction du nombre de bits lm (dans ce qui suit nous posons lm= Nb_{proc}) .

L'intérêt de ces exemples est l'étude comparative de la qualité de sortie (performances|complexité algorithmique) en utilisant la méthode R.A. avec les trois représentations binaires.

Pour ces filtres, nous avons choisi les notations suivantes :

R.A. : algorithme de la méthode de la recherche arborescente.
P.M.C.Q. : l'algorithme de Parks-McClellan avec les coefficients quantifiés par arrondissement.
N : la longueur du filtre.
lm : longueur du mot machine 'processeur de signaux'.
wp : fréquence normalisée à la fin de la bande passante.
ws : fréquence normalisée au début de la bande atténuée.
Vfx : représentation binaire en virgule fixe.
Vfl : représentation binaire en virgule flottante.
Sdpd : représentation binaire en somme de deux de puissance de deux.
Tps : temps de synthèse du filtre.

IV.1. FILTRE 1 :

Soit à concevoir un filtre de spécifications suivantes :

N=8.
lm = 8 bits.
wp = 0.159.

ws = 0.295.

La synthèse du filtre 1, nous a donné les résultats qui sont groupés dans les tableaux suivants. Ils ont été obtenus à partir d'un PC de fréquence CPU 300 Mhz et de 32 Mo RAM.

	R.A.dans le sens de Ems				P.M.C.Q.	
	Ems	Emp	Ema	Tps	Ems	Tps
Vfx	0.0319	0.0843	0.0893	26 h.	0.0358	0.05 s.
Vfl	0.0320	0.100	0.060	18 h.	0.0392	0.05 s.
Sdpd	0.127	0.320	0.210	12 s.	0.127	0.05 s.

Tableau 4 : Représentation de l'erreur quadratique moyenne par les deux méthodes (R.A. et P.M.C.Q.) dans les différentes représentations à lm =8 bits.

Coefficients de filtre de longueur :8	Vfx R.A. | P.M.C.Q.		Vfl R.A. | P.M.C.Q.		Sdpd R.A. | P.M.C.Q.	
h(0) =h(7)	-0.0546875	-0.0625000	-0.05468750	-0.06250000	0	0
h(1)=h(6)	-0.0390625	-0.0468750	-0.03515625	-0.05078125	0	0
h(2) =h(5)	0.1640625	0.1718750	0.17187500	0.17187500	0.125	0.125
h(3)=h(4)	0.4140625	0.4140625	0.40625000	0.40625000	0.375	0.375

Tableau5 : Tableau de coefficients du filtre obtenu par R.A. dans chaque représentation à lm=8 bits.

Le tableau4 présente les erreurs des filtres conçus au moyen des deux méthodes R.A. et P.M.C.Q dans les représentations Vfx, Vfl et Sdpd. Sur ce tableau, nous avons présenté l'erreur quadratique moyenne et l'erreur maximale dans la bande passante et dans la bande atténuée.

Nous remarquons que l'erreur quadratique moyenne du filtre conçu par la méthode R.A. dans la représentation à virgule fixe est la plus petite. Le temps de calcul correspondant est prohibitif (26 heures). Tous les résultats obtenus par la méthode R.A. sont meilleurs que ceux obtenus par arrondissement à partir de PMCQ.

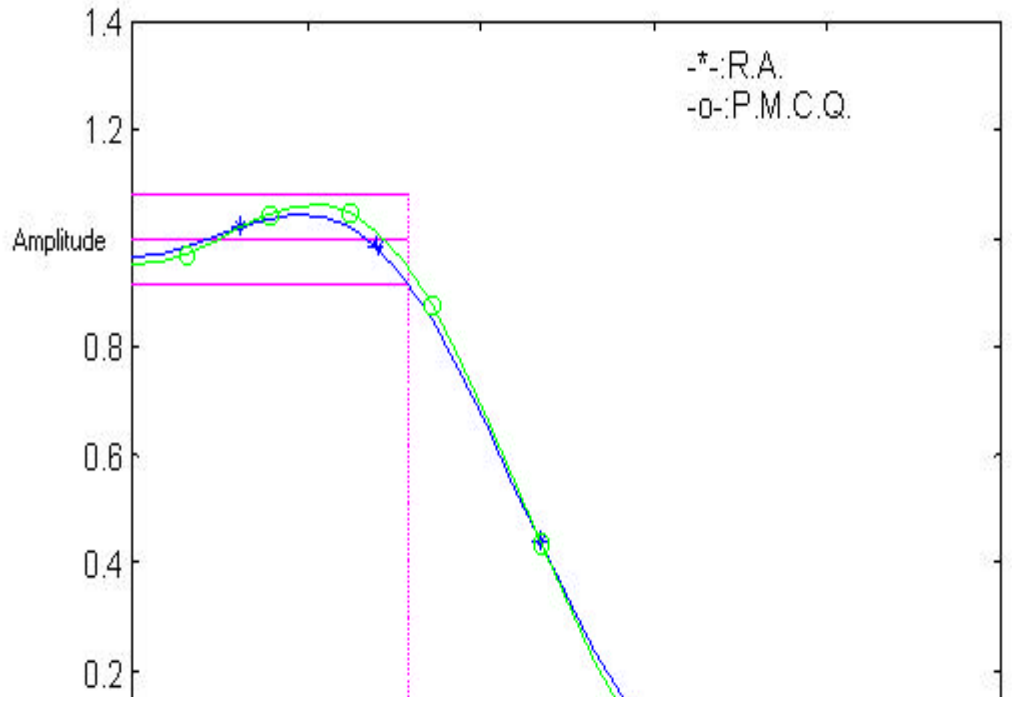
Dans la représentation Sdpd, pour un temps de synthèse de 12 secondes, l'erreur quadratique moyenne du filtre conçu est grande par rapport à celle des deux autres représentations dans les deux méthodes RA et PMCQ. Si nous comparons la qualité de sortie (Ems | tps) pour les deux méthodes nous aurons ce qui suit :

- Pour PMCQ nous avons (0.0358|0.05s), (0.0392|0.05s) et (0.127|0.05s) respectivement en Vfx, Vfl et Sdpd. Nous constatons pour un temps de calcul identique, nous avons des performances différentes. Dans ce cas, nous concluons l'utilisation de la Vfx est recommandée pour avoir la meilleure solution dans le sens Ems.
- Pour RA nous avons (0.0319|26h), (0.0320|18h) et (0.127|12s) respectivement en Vfx, Vfl et Sdpd. Nous constatons l'algorithme dans la Sdpd présente la convergence la plus rapide à précision moindre, tandis qu'en Vfx, il a la convergence la plus lente pour la plus grande précision.

Nous concluons le choix de la représentation dépend des spécifications requises (précision|temps de calcul).

Le tableau 5 présente les coefficients des filtres obtenus avec les deux méthodes R.A. et P.M.C.Q dans les trois représentations binaires. La différence entre les valeurs des coefficients n'est pas très grande dans les trois représentations binaires, mais très influente sur la réponse en amplitude. Nous remarquons aussi, la grande précision des coefficients dans la représentation à virgule flottante, elle réside dans le grand nombre de chiffres significatifs.

A partir des coefficients obtenus, nous avons schématisé l'allure d'amplitude des filtres correspondants à chaque représentation. Nous avons obtenu les figures



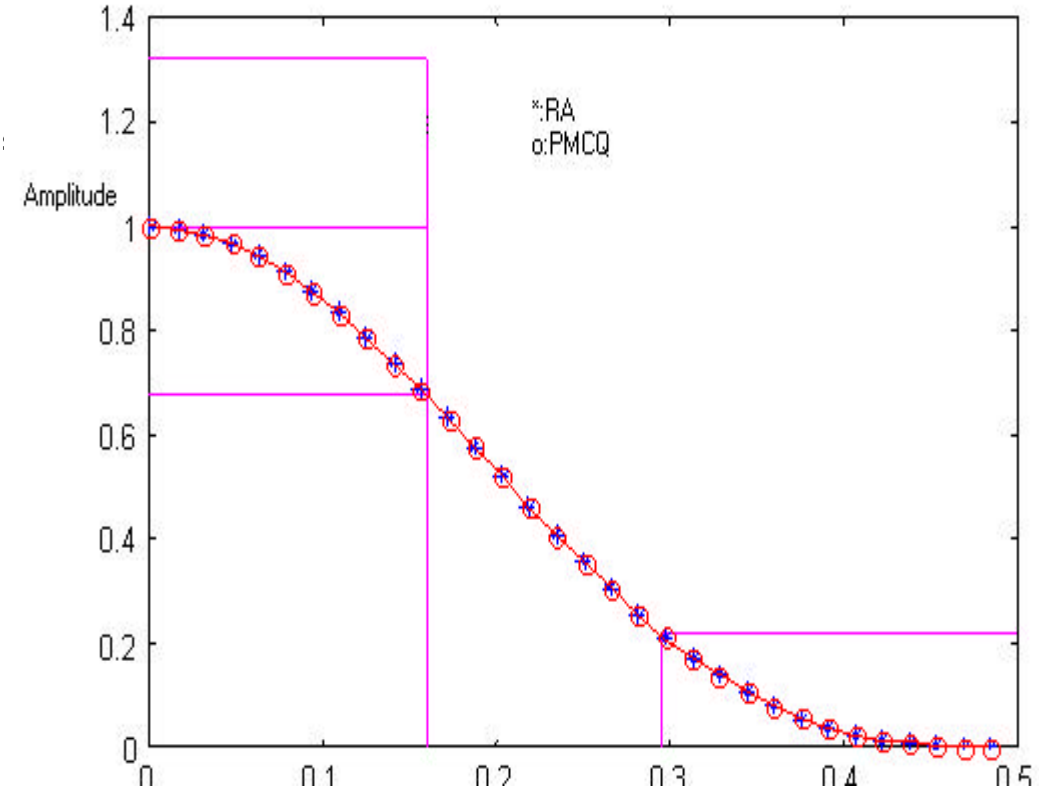


Fig.15. RA et PMCQ dans la représentation Sdpd avec lm =8 bits

Nous remarquons que les filtres en représentation sdpd présente la tolérance la plus grande (Ema et Emp) dans la BP et la BA. Ceux en Vfx et Vfl entrent dans le gabarit. La différence entre les filtres de ces deux représentations est faible.

Pour mieux expliquer ces résultats, le tableau 6 a été conçu pour un filtre de longueur 8 en tenant compte des tableaux 1, 2 et 3 du chapitre 1 correspondants aux nombres de valeurs admissibles, des valeurs du plus petit pas et des valeurs du plus grand pas. Dans ce tableau, nous considérerons la symétrie (antisymétrie) des filtres RIF à phase linéaire. En effet, nous avons à retrouver N/2 coefficients dans le cas d'un filtre de longueur N. Par conséquent, le nombre de fonctions d'évaluation 'N_{fe}' est calculé comme suit :

$$N_{fe} = (Nbad)^{N/2}$$

Avec Nbad : le nombre de valeurs admissibles dans un mot de longueur lm.
Pour N=8 nous aurons

$$N_{fe} = Nbad^4$$

Lm	Vfx	Vfl	Sdpd
8 bits	4228250625	418161601	83521
16 bits	65535^4	36863^4	7817^4

Tableau 6: Nombre de fonctions d'évaluation dans l'espace discret
à lm bits suivant chaque représentation pour un filtre de longueur 8.

A partir de l'Eq. 47 et du tableau 6, nous remarquons que le nombre de coefficients à retrouver a plus d'importance sur le nombre de fonctions d'évaluation par rapport au nombre de valeurs admissibles. Nous remarquons aussi que la plus grande valeur de N_{fe} est celle de Vfx, tandis celle la plus petite est de Sdpd. Cela est dû au nombre de valeurs admissibles correspondant dans ces représentations.

Par conséquent, le temps de calcul est plus petit en utilisant la représentation Sdpd et le plus grand en utilisant la représentation Vfx.

A ce propos, pour mieux tester les limites de la complexité algorithmique de RA, nous proposerons dans le paragraphe une synthèse de filtres dont les coefficients sont représentés sur un mot de longueur plus grande lm=12 bits.

IV.2. FILTRE 2 :

N=8 .
lm = 12 bits.
wp = 0.159.
ws = 0.295.

Les résultats de notre expérimentation sont rassemblés dans les deux tableaux qui suivent :

	R.A. dans le sens de Ems				P.M.C.Q.	
	Ems	Emp	Ema	Tps	Ems	Tps
Vfx	---------	---------	--------	∞	0.0370	0.44 s.
Vfl	---------	---------	--------	∞	0.0372	0.38 s.
Sdpd	0.0447	0.0711	0.1498	246 h.	0.0624	0.27 s.

Tableau 7 : : Représentation de l'erreur quadratique moyenne par les deux méthodes (R.A. et P.M.C.Q.)
dans la représentation Sdpd à lm=12 bits.

Coef. de filt. de longueur:8	Vfx R.A.	Vfx P.M.C.Q.	Vfl R.A.	Vfl P.M.C.Q.	Sdpd R.A.	Sdpd P.M.C.Q.
h(0) =h(7)	----------	-0.0615234375	--------	-0.0617675781	-0.0468750	-0.06201171875
h(1)=h(6)	----------	-0.0498046875	--------	-0.0498046875	-0.0546875	-0.04687500000
h(2) =h(5)	----------	0.1718750000	--------	0.1718750000	0.1562500	0.18750000000
h(3)=h(4)	----------	0.4106445312	--------	0.4101562500	0.4375000	0.43750000000

Tableau 8 : Tableau de coefficients du filtre obtenu par R.A. dans la représentation Sdpd et les coefficients de P.MC.Q. à lm=12 bits.

Le tableau 7 présente le temps de synthèse et les erreurs Ems des filtres conçus avec les deux méthodes R.A. et P.M.C.Q. et la représentation binaire Sdpd.

La méthode R.A. n'est pas utilisable dans les représentations Vfx et Vfl à cause du temps de calcul prohibitif. C'est pourquoi, les cases qui y correspondent sont en pointillé.

A partir du tableau 7, nous remarquons que le qualité de sortie (Ems|tps) en représentation Sdpd est (0.0447|246h).

Comparés aux résultats du filtre 1 qui sont de (0.0319|26h), (0.0320|18h) respectivement en Vfx et Vfl, nous constatons que le temps de calcul est approximativement dix fois plus petit pour une précision 1.5 meilleure.

Concernant cet exemple, nous pouvons affirmer que la représentation Sdpd n'est pas adéquate pour la synthèse de filtres dans un espace discret, puisque les résultats montrent que les représentations Vfx et Vfl ont procurés des performances meilleurs dans un temps de calcul plus petit.

De plus, ces deux dernières représentations ont été utilisées dans un espace discret de longueur de mot plus petite, où l'erreur due la représentation est plus grande, comparée à celle due à l'espace discret où Sdpd a été utilisée.

Nous trouverons dans le paragraphe V, une étude comparative de l'efficacité des trois représentations Vfx, Vfl et Sdpd dans l'espace discret des valeurs des coefficients. Nous concluons que les représentations Vfx et Vfl offrent une grande représentativité des coefficients vu leur représentation régulière dans toute la plage des valeurs [-1, 1] par rapport à Sdpd.

Le temps de calcul reste l'inconvénient majeur de l'utilisation de ces représentations binaires Vfx et Vfl. Les exemples 1 et 2 montrent que le qualité de sortie (Ems| tps) doit être étudié d'une manière plus approprié concernant le choix entre la représentation Sdpd et les deux représentations Vfx et Vfl.

Nous noterons par PMCQ le filtre obtenu, à partir des spécifications identiques, dont les coefficients ont été tronqués sur la même longueur que les filtres en étude.

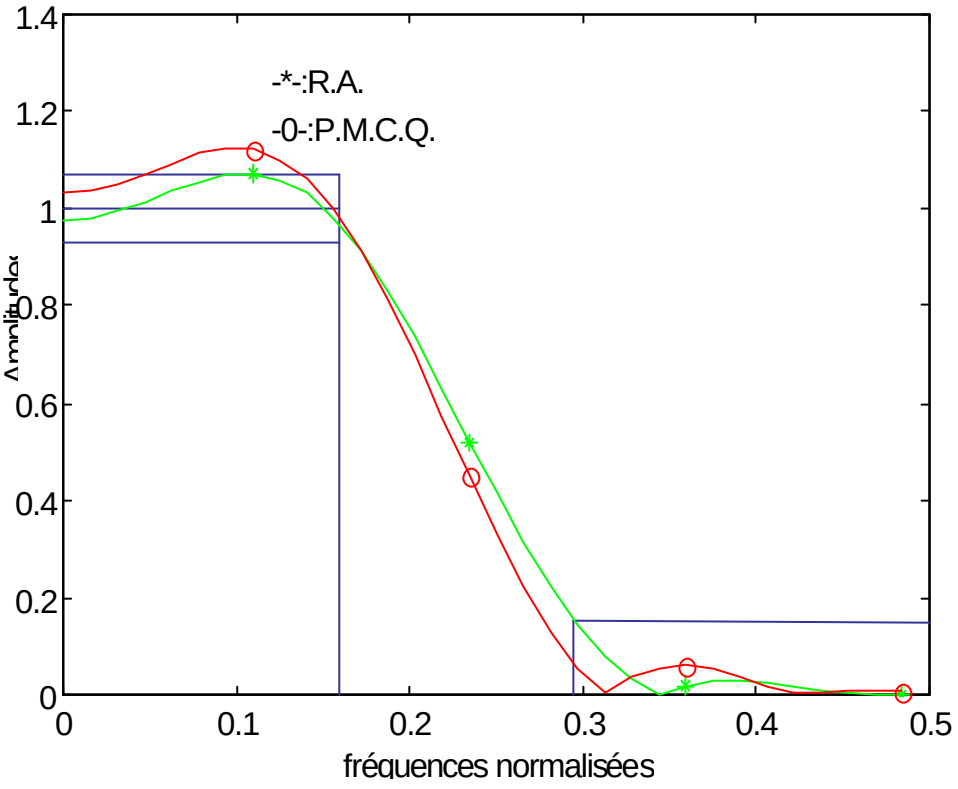


Fig.16. RA et PMCQ dans la représentation Sdpd avec lm =12 bits

Nous remarquons que le filtre de P.M.C.Q. sort du gabarit prescrit par celui de la méthode de R.A.

Afin de confirmer les résultats obtenus dans ce cas de filtre, nous avons étendu notre travail dans les deux paragraphes suivants pour d'autres cas de filtres de spécifications différentes. Le premier filtre possède une BP large et une BA étroite, tandis que le deuxième possède une BP étroite et une BA large.

IV.3. FILTRE 3 :

N=8.
lm = 8 bits.

wp = 0.45.
ws = 0.48.

La synthèse du filtre choisi, nous a donné les résultats qui sont groupés dans les tableaux suivants :

	R.A. dans le sens de Ems				P.M.C.Q.	
	Ems	Emp	Ema	Tps	Ems	Tps
Vfx	0.09562	0.34907	0.27739	16 h.	0.21120	0.06 s.
Vfl	0.09624	0.35396	0.27543	18 h.	0.20760	0.05 s.
Sdpd	0.15065	0.26895	0.35587	12 s.	0.17338	0.05 s.

Tableau9 : Représentation de l'erreur quadratique moyenne par les deux méthodes (R.A. et P.M.C.Q.) dans les différentes représentations avec lm =8 bits.

Coefficients de filtre de longueur :8	Vfx		Vfl		Sdpd	
	R.A.	P.M.C.Q.	R.A.	P.M.C.Q.	R.A.	P.M.C.Q.
h(0) =h(7)	-0.0625000	0	-0.06250000	-0.00341796	-0.125	0
h(1)=h(6)	0.1093750	0.2187500	0.10937500	0.21875000	0.125	0.125
h(2) =h(5)	-0.2031250	-0.2031250	-0.20312500	-0.20312500	-0.125	-0.125
h(3)=h(4)	0.6328125	0.6328125	0.62500000	0.62500000	0.625	0.625

Tableau 10 : Tableau de coefficients du filtre obtenu par R.A. dans chaque représentation avec lm=8 bits.

Nous remarquons du tableau 9 que les filtres conçus par RA sont tous meilleurs en performance dans le sens de Ems que ceux dePMCQ par arrondissement pour des temps de calcul plus grands. Par ailleurs la qualité de sortie (Ems| tps) de RA est (0.09562|26h), (0.09624|18h) et (0.15065|12s) respectivement en Vfx, Vfl et Sdpd.

Les mêmes remarques faites au filtre 1 sont valables pour ce cas de filtre. Le filtre conçu par RA en Vfx présente l'erreur Ems la plus faible pour un temps de calcul de 26 heures, tandis que celui de Sdpd présente un tps de 12 secondes pour des performances moins bonnes. Le choix de la représentation dépend des spécifications (précision | tps) requises.

A partir des coefficients du tableau 10, nous avons schématisé les filtres suivants :

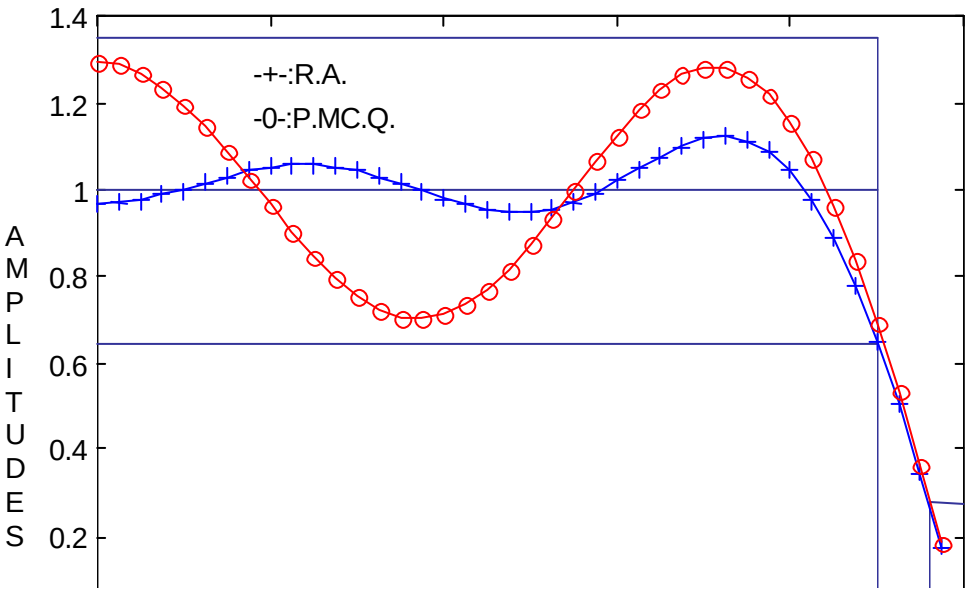


Fig.17. RA et PMCQ dans la représentation Vfx avec lm =8 bits

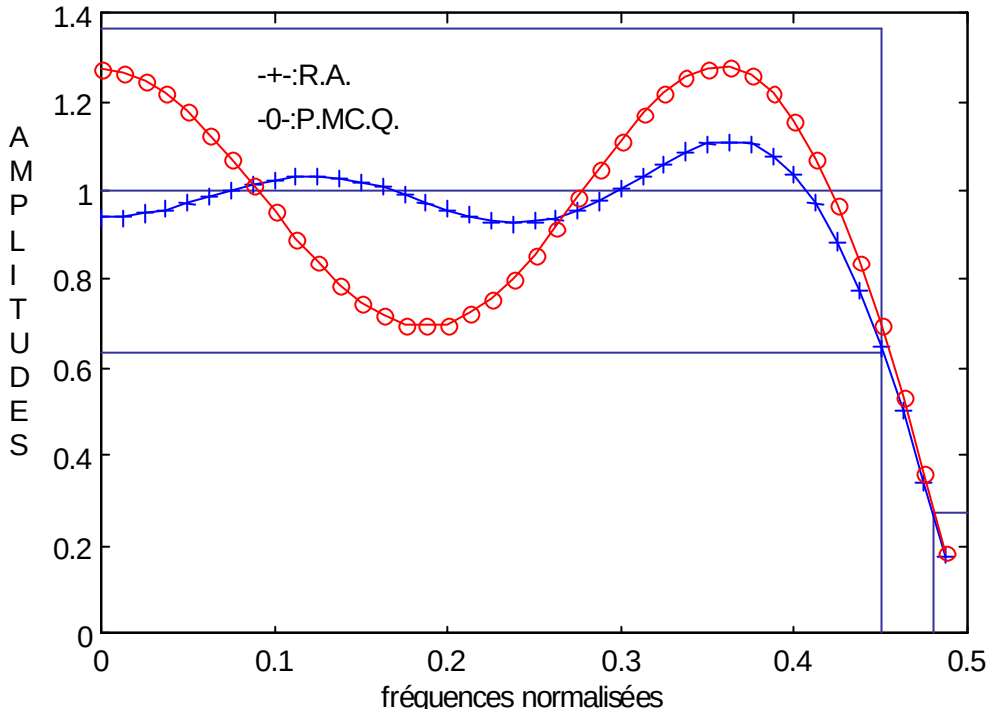


Fig.18. RA et PMCQ dans la représentation Vfl avec lm =8 bits

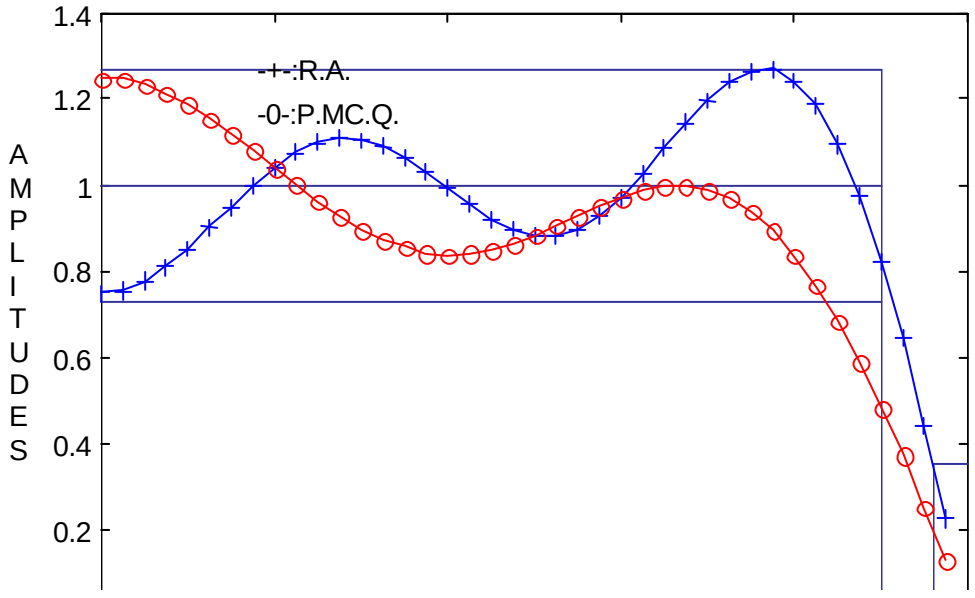


Fig.19. RA et PMCQ dans la représentation Sdpd avec lm =8 bits

Les figures précédentes représentent les filtres conçus par la méthode R.A. et ceux P.M.C.Q. dans les trois représentations binaires Vfx, Vfl et Sdpd. Nous remarquons que l'allure des filtres est cohérente. La différence entre les filtres de ces deux représentations ne peut pas être déterminée à l'œil nu. Le filtre conçu par P.M.C.Q. par arrondissement en représentation Sdpd n'entre pas dans le gabarit prescrit.

IV.4. FILTRE 4 :

N=8.
lm = 8 bits.
wp = 0.1.
ws = 0.25.

La synthèse du filtre choisi, nous a donné les résultats suivants :

	R.A.				P.M.C.Q.	
	Ems	Emp	Ema	Tps	Ems	Tps
Vfx	0.04201	0.07980	0.13258	26 h.	0.05290	0.05 s.
Vfl	0.04201	0.07980	0.13258	18 h.	0.05473	0.05 s.
Sdpd	0.13453	0.25000	0.19719	11 s.	0.17571	0.05 s.

Tableau11 : Représentation de l'erreur quadratique moyenne par les deux méthodes (R.A. et P.M.C.Q.) dans les différentes représentations avec lm =8 bits.

Coefficients de filtre de longueur :8	Vfx		Vfl		Sdpd	
	R.A.	P.M.C.Q.	R.A.	P.M.C.Q.	R.A.	P.M.C.Q.
h(0) =h(7)	-0.0390625	-0.0468750	-0.0390625	-0.04296875	0	0
h(1)=h(6)	0.0234375	0. 0468750	0.0234375	0.04296875	0.125	0
h(2) =h(5)	0.1875000	0.1953125	0.1875000	0.20312500	0.250	0.125

h(3)=h(4)	0.3437500	0.3359375	0.3437500	0.34375000	0.250	0.250

Tableau12 : Tableau de coefficients du filtre obtenu par R.A. à chaque représentation avec lm=8 bits.

A partir du tableau 11, nous constatons que les mêmes remarques faites au filtre 1 et 3 sont confirmés pour ce cas de filtre. le qualité de sortie (Ems| tps) de RA est de (0.04201|26h), (0.04201|18h) et (0.13453|11s) respectivement en Vfx, Vfl et Sdpd. Le temps de calcul en Sdpd est le plus petit pour l'erreur Ems la plus grande. Tandis que tps en Vfx est 1,5 plus grand que celui en Vfl pour une Ems identique. Par conséquent, entre Vfx, et Vfl, il est recommandé de choisir Vfl dans ce cas, alors que le choix de Sdpd dépend des spécifications (Ems|tps) requises. A partir des coefficients du tableau 12, nous avons les filtres suivants :

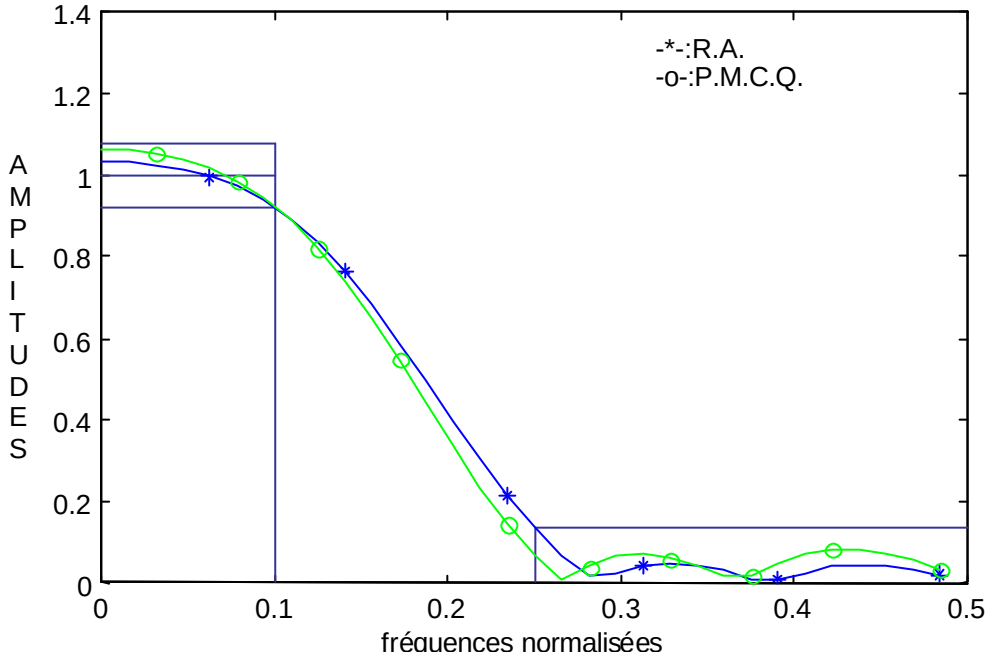


Fig.20. RA et PMCQ dans la représentation Vfx avec lm =8 bits

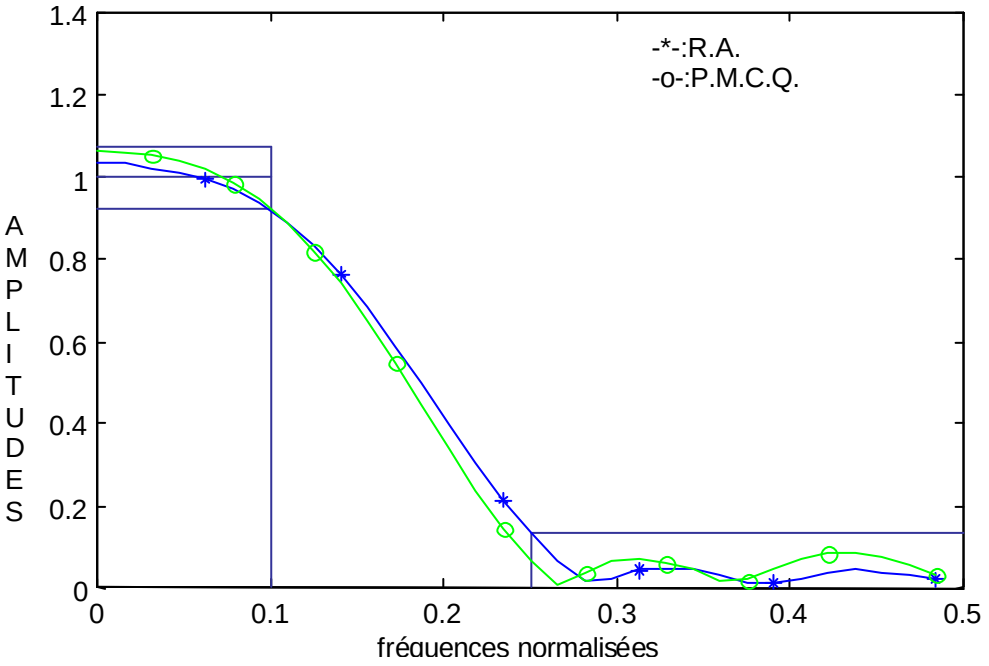


Fig.21. RA et PMCQ dans la représentation Vfl avec lm =8 bits

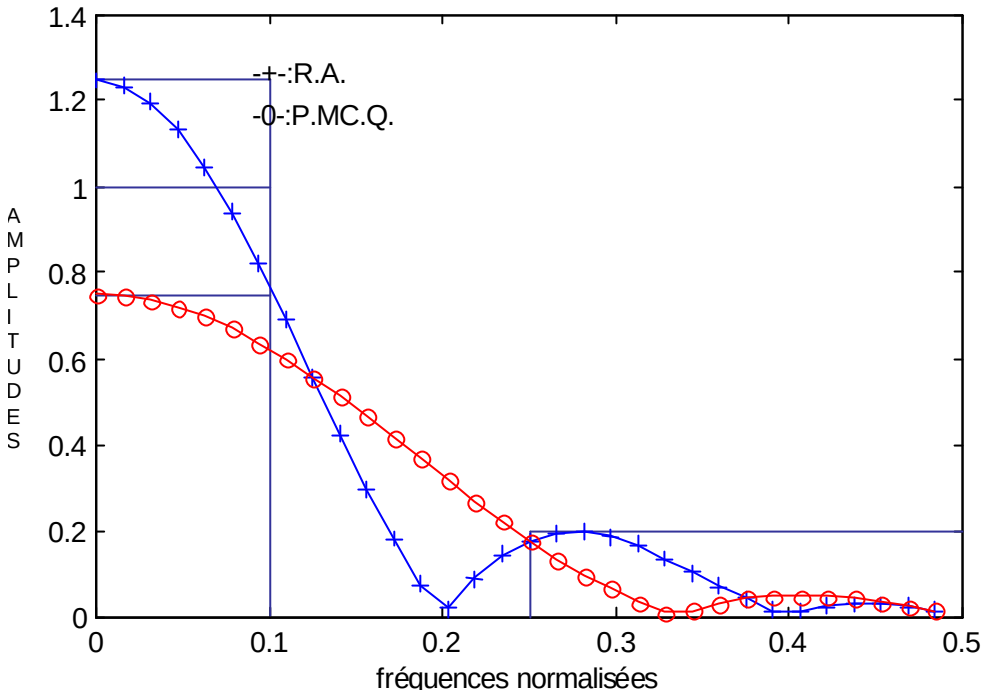


Fig.22. RA et PMCQ dans la représentation Sdpd avec lm =8 bits

Les mêmes remarques faites aux figures des filtres 1 et 3 sont vérifiées pour ce cas de filtre. Le filtre conçu par P.M.C.Q. par arrondissement dans la représentation Sdpd n'entre pas dans le gabarit prescrit. Par ailleurs, comparé à Vfx et Vfl, le filtre de RA dans Sdpd présente les erreurs Emp et Ema les plus grandes respectivement dans BP et dans BA.

Dans le paragraphe suivant, nous allons présenter une étude comparative des quatre cas de filtres reportés précédemment en précision et en complexité algorithmique en utilisant la méthode RA.

XVI. ETUDE DES RESULTATS DE LA METHODE RA :

Une étude comparative des résultats obtenus dans les quatre exemples nous montre que les filtres obtenus dans les représentations en virgule fixe et en virgule flottante possèdent de bonnes performances, tandis que ceux conçus dans une représentation Sdpd sont de performances moindres. Relativement à ces résultats, le temps de calcul associé à la synthèse de filtres dans la représentation Sdpd est très petit contrairement aux représentations en virgule fixe et en virgule flottante qui présentent un temps prohibitif lorsque la longueur du mot machine 'lm' et celle du filtre sont élevées.

Afin de mieux interpréter ces résultats, nous donnons l'explication en nous aidant des exemples de filtre 1 et 2.

Nous remarquons dans ces deux exemples que les valeurs de (Ems | temps de calcul | lm) sont de (0.0319 | 26 h | 8 bits), (0.0320 | 18 h | 8 bits) et (0.0447 | 246 h | 12 bits) respectivement dans les représentations en Vfx , en Vfl et en sdpd. Malgré que la représentation Sdpd a été utilisée dans un espace discret de longueur de mot plus grande que celle utilisée en représentation Vfx et vfl, nous constatons que Vfx a procuré le meilleur résultat en Ems et que Vfl a procuré un temps de calcul le plus petit à une erreur presque égale à celle de Vfx. Alors qu'en représentation Sdpd, nous avons le temps de calcul le plus grand avec des performances moindres.

Afin d'expliquer le comportement de ces résultats, nous présentons les valeurs du coefficient h(3) = h(4) du filtre 1 et du filtre 2 obtenu par la méthode RA et PMCQ par arrondissement. Ce coefficient est celui à plus grande valeur ou du plus grand poids (en valeur absolue) par rapport aux autres coefficients du filtre. Pour les filtres RIF à phase linéaire passe bas ce coefficient est d'une grande importance sur le gabarit du filtre.

- En représentation Vfx sur lm =8 bits :
  - h(3)=0.4140625 par la méthode RA.
  - h(3)=0.4140625 par la méthode PMCQ.
- En représentation Vfl sur lm =8 bits :
  - h(3)= 0.40625 par la méthode RA.
  - h(3)= 0.40625 par la méthode PMCQ.
- En représentation Sdpd sur lm =12 bits :
  - h(3)= 0.43750 par la méthode RA.
  - h(3)= 0.43750 par la méthode PMCQ.

La valeur du coefficient h(3) en représentation Vfx et Vfl dans un espace discret à lm=8 bits et celle du coefficient h(3) en représentation Sdpd dans un espace discret à lm=12 bits est différente. C'est pourquoi, nous avons obtenu des erreurs Ems différentes dans chaque représentation. Afin de connaître la raison pour laquelle la représentation Sdpd n'a pas procuré des performances meilleures ou identiques à ceux de Vfx malgré qu'elle est configurée sur un mot de longueur plus grande, nous présentons une fenêtre de l'espace discret d'un intervalle de]0.35, 0.5] dans chaque représentation.

- En représentation Vfx sur lm =8 bits :

  E_{proc} ={ 0.3515625 0.359375 0.3671875 0.375 0.38281250000000 0.3906250 0.3984375 0.40625 0.4140625 0.421875 0.4296875 0.4375 0.4453125 0.453125 0.4609375 0.46875 0.4765625 0.484375 0.4921875 0.5}
- En représentation Vfl sur lm =8 bits :

  E_{proc} **={ 0.3750 0.4062 0.4375 0.4688 0.5000}.**

- En représentation Sdpd sur lm = 12 bits :

  E_{proc} **={0.3750 0.4375 0.46875 0.484375 0.4921875 0.49609375 0.498046875 0.4990234375 0.49951171875 0.499755859375 0.4998779296875 0.49993896484375 0.49996948242188 0.5}**

Nous remarquons que la valeur du coefficient h(3) = 0.4140625 en Vfx n'est pas admissible (représentable) dans un espace à lm=12 bits de représentation Sdpd. Les deux valeurs discrètes représentables en Sdpd les plus proches sont 0.3750 et 0.4375. La différence est assez importante. Par conséquent, le filtre obtenu en cette

représentation Sdpd présente une erreur quadratique moyenne plus grande que celle dans Vfx.

Ce cas de filtre présente un échantillon de plusieurs autres où la représentation Sdpd peut être déconseillée pour la synthèse de filtres numériques dans un espace discret de longueur de mot 'lm'. Il est souvent préférable d'utiliser la représentation Vfx ou Vfl sur un mot de longueur 'lm-c'(avec c un entier inférieur à lm-1), pour gagner en performances et en complexité algorithmique par rapport à l'utilisation de Sdpd sur lm bits.

V.I. OPTIMISATION PAR D.M.C. DANS LES TROIS REPRESENTATIONS :

VI.1. INTRODUCTION :

Dans ce paragraphe, nous allons exploiter la rapidité de convergence de la méthode DMC en utilisant des représentations binaires qui procurent un nombre de valeurs admissibles plus grand afin de tester ses capacités. Ensuite, nous allons effectuer une étude comparative des résultats obtenus en utilisant les représentations Vfx et Vfl par rapport à l'utilisation de la représentation Sdpd en évaluant le qualité de sortie (précision| temps de calcul) pour chaque cas.

Nous différencions entre les coefficients non quantifiés et quantifiés, par l'utilisation de la notation suivante :

h(n) : le coefficient non quantifié, représenté par un mot de longueur Nb_{ord} bits.

$h_d(n)$: le coefficient h(n) quantifié par arrondissement dans les représentations Vfx, Vfl et Sdpd est représenté par un mot de longueur Nb_{proc} bits < Nb_{ord} bits.

Dans la méthode D.M.C.[31], que nous rappelons en annexe, les coefficients sont calculés d'une manière séquentielle. Le critère d'évaluation utilisé par cette méthode est l'erreur quadratique moyenne 'Ems'. Nous allons dans le paragraphe suivant présenter l'essentiel de l'algorithme DMC.

VI.2. DESCRIPTION DE L'ALGORITHME :

La méthode D.M.C., se compose de N/2 étapes. A chaque étape, on calcule un seul coefficient à la fois.

Etape 1:

Déterminer le coefficient $h(\frac{N}{2}-1)$. Ce premier coefficient est optimisé suivant l'Eq. 93 dans l'annexe, sa valeur est ensuite quantifiée par arrondissement en $h_d(N/2-1)$. Après sa détermination, il en résultera une erreur sur l'amplitude de la réponse en fréquence, qu'on a appelé erreur d'amplitude résiduelle effective $E_1(w)$ par rapport à l'amplitude du filtre idéal $H_D(e^{jw})$, et qui est calculée par:

$$E_1(w)=\left|H_D(e^{jw})\right| - 2h_d(\frac{N}{2}-1)\cos(\frac{w}{2}) \qquad (47.a)$$

Si on pose $E_0(w)= |H_D(e^{jw})|$

on obtient :

$$E_1(w)=E_0(w) - 2h_d(\frac{N}{2}-1)\cos(\frac{w}{2}) \qquad (47.b)$$

Etape 2:

A partir de E_1 (w), on recherche le coefficient suivant $h(\frac{N}{2}-2)$ par la méthode d'optimisation donnée par l'Eq. 98.b en annexe. La valeur obtenue est quantifiée par arrondissement.

Après la détermination du second coefficient $h_d(\frac{N}{2}-2)$, de la même façon il en résulte une nouvelle erreur d'amplitude résiduelle effective E_2 (w) par rapport à l'erreur d'amplitude résiduelle effective E_1 (w) précédente. E_2 (w) est calculée par :

$$E_2(w) = E_1(w) - 2h_d(\frac{N}{2}-2)\cos(\frac{3w}{2}) \qquad (48.a)$$

$$E_2(w) = E_0(w) - 2h_d(\frac{N}{2}-1)\cos(\frac{w}{2}) - 2h_d(\frac{N}{2}-2)\cos(\frac{3w}{2}) \qquad (48.b)$$

Etape 3:

De la même façon, le coefficient $h(\frac{N}{2}-3)$ est optimisé en fonction de E_2 (w), puis sa valeur est quantifiée. Et ainsi de suite ...

Etape n :

D'une façon générale, le coefficient $h(\frac{N}{2}-n-1)$ est optimisé et quantifié par arrondissement de l'erreur d'amplitude résiduelle effective E_n (w), obtenue a la fin de l'étape n-1. Apres sa détermination, il reste en fin de l'étape n une erreur d'amplitude résiduelle effective E_{n+1}(w) qui est calculée par :

$$E_{n+1}(w) = E_n(w) - 2h_d(N/2 - n - 1)\cos(n + 1/2)w$$

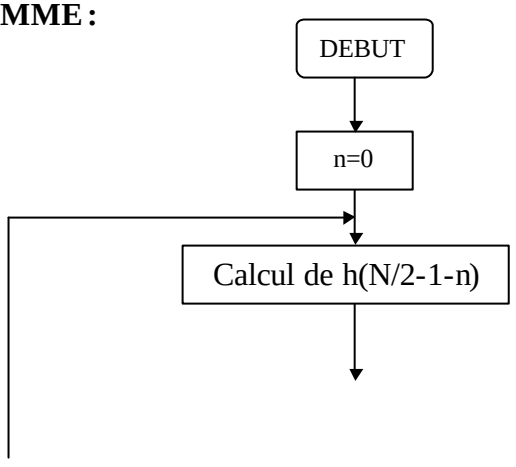
$$E_{n+1}(w) = E_0(w) - \sum_{k=0}^{n} 2h_d(\frac{N}{2}-1-k)\cos((k+\frac{1}{2})w)$$

avec n = 1 ... N/2

Etape $\frac{N}{2}-1$: (dernière étape)

Le dernier coefficient h(0) est optimisé en fonction de l'erreur $E_{N/2-1}$(w) puis sa valeur est quantifiée. L'erreur d'amplitude résiduelle effective $E_{N/2}$(w) qui résultera après la quantification du dernier coefficient ne pourra plus être diminuée; les coefficients étant tous calculés.

VI.3. ORGANIGRAMME :



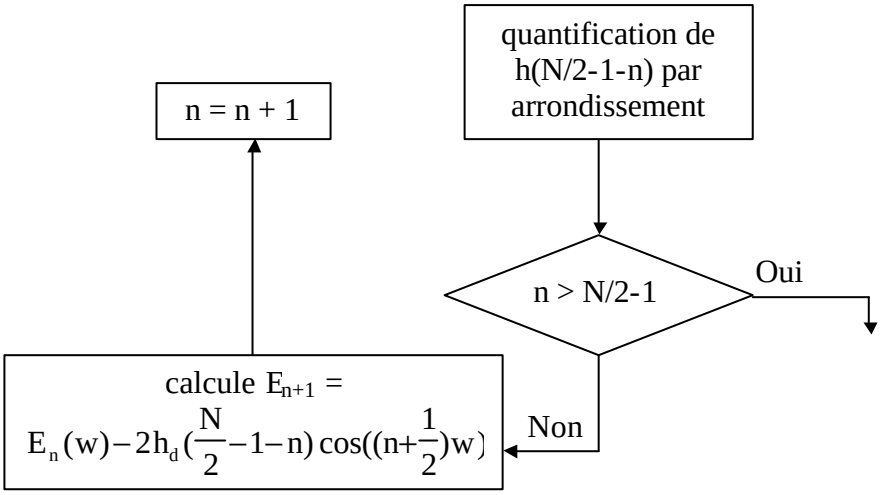


Fig.23 Organigramme de la synthèse de filtre par la méthode D.M.C..

Cet organigramme présente la démarche algorithmique de la méthode DMC donnée dans [31]. Dans le paragraphe suivant, nous allons étudier les performances et la complexité de cette méthode avec les trois représentations binaires Vfx, Vfl et Sdpd.

VII. RESULTATS DES FILTRES CONCUS PAR LA METHODE D.M.C. :

Cette méthode a été testée sur plusieurs centaines de filtres dont nous présenterons quatre repérés par les numéros de 1 à 4. Ces 4 filtres choisis sont des filtres RIF à phase linéaire passe bas dont les spécifications sont identiques à ceux obtenus par la méthode RA. Nous nous sommes aussi intéressé, pour chaque filtre, d'étudier l'évolution de ses coefficients en fonction du nombre de bits lm.

VII.1. FILTRE 1 :

N=8.
lm = 8 bits.
wp = 0.159.
ws = 0.295.

La synthèse du filtre choisi, nous a donné les résultats qui sont groupés dans les tableaux suivants :

	D.M.C.				P.M.C.Q.	
	Ems	Emp	Ema	Tps	Ems	Tps
Vfx	0.0330	0.0718	0.0844	0.05 s.	0.0358	0.05 s.
Vfl	0.0374	0.1002	0.1028	0.05 s.	0.0392	0.05 s.
Sdpd	0.1270	0.3235	0.2164	0.05 s.	0.127	0.05 s.

Tableau13 : Représentation des erreurs par les deux méthodes (D.M.C. et P.M.C.Q.) dans les différentes représentations à lm =8 bits.

Coefficients de filtre de longueur :8	Vfx D.M.C. | P.M.C.Q.	Vfl D.M.C. | P.M.C.Q.	Sdpd D.M.C. | P.M.C.Q.

h(0) =h(7)	-0.0546875	-0.0625000	-0.05078125	-0.06250000	0	0
h(1)=h(6)	-0.0468750	-0.0468750	-0.04296875	-0.05078125	0	0
h(2) =h(5)	0.1640625	0.1718750	0.15625000	0.17187500	0.125	0.125
h(3)=h(4)	0.4140625	0.4140625	0.40625000	0.40625000	0.375	0.375

Tableau14 : Tableau de coefficients du filtre obtenu par D.M.C. à chaque représentation pour lm=8 bits.

Le tableau 13 présente les performances et le temps de calcul de DMC et PMCQ par arrondissement dans les représentations binaires. Nous remarquons que les erreurs Ems de DMC sont inférieures à celles de PMCQ avec un temps de calcul identique.

Pour la méthode DMC nous avons la qualité de sortie (Ems | tps) de (0.0330| 0.05s), (0.0374| 0.05s) et (0.127| 0.05s) respectivement en Vfx, Vfl et Sdpd. Nous remarquons que Vfx offre le meilleur résultat pour un temps de calcul identique à celui de Vfl et de Sdpd.

A partir des coefficients du tableau 14, nous avons schématisé l'allure des filtres retrouvés à chaque représentation. Nous avons obtenu les figures suivantes :

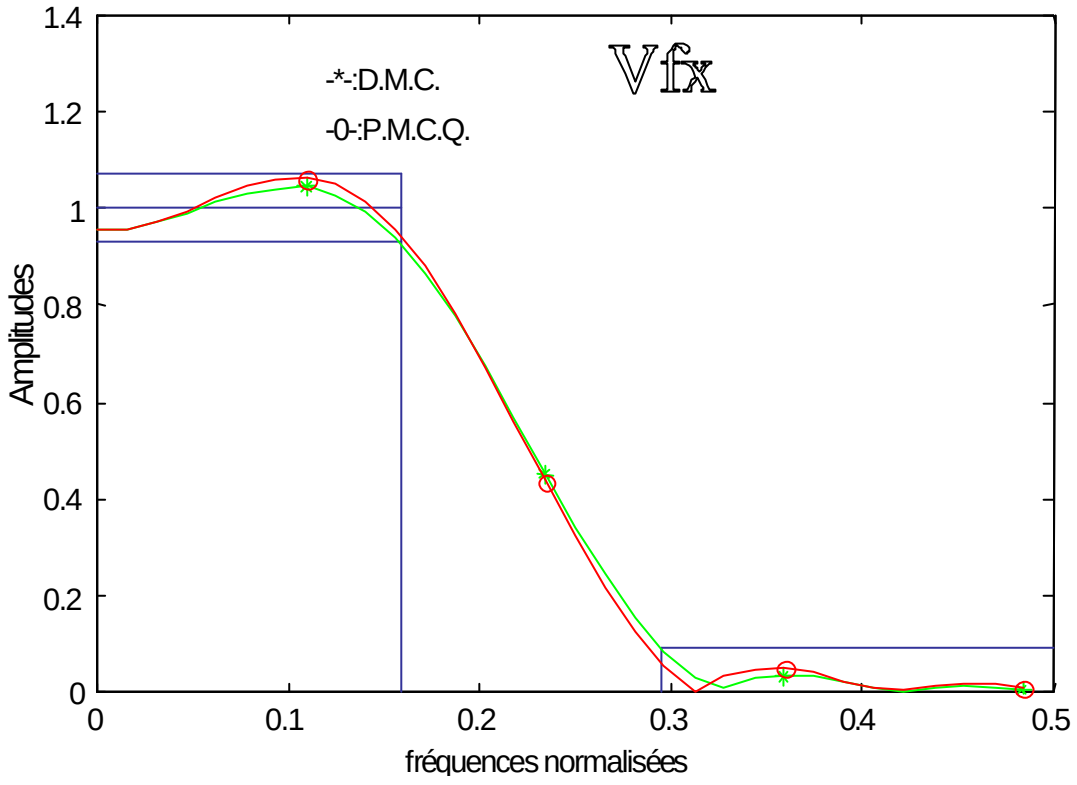


Fig.23. DMC et PMCQ dans la représentation Vfx sur lm =8 bits

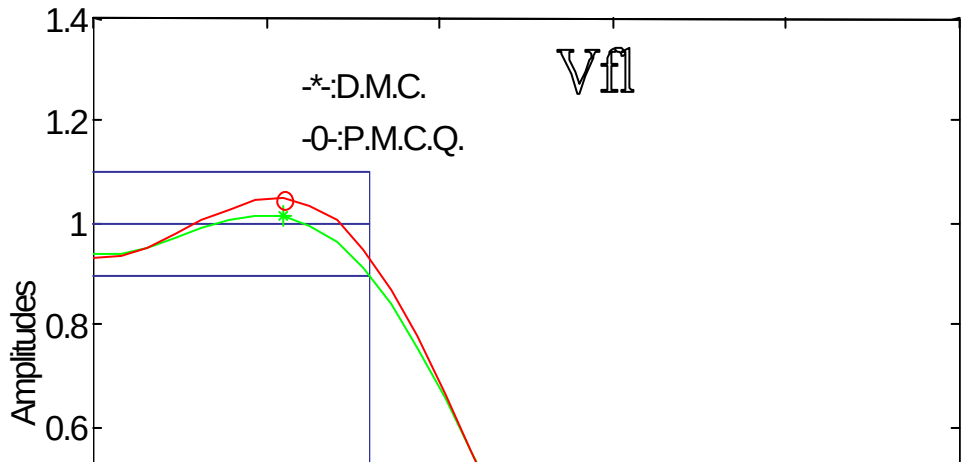


Fig.24. DMC et PMCQ en représentation Vfl sur lm =8 bits

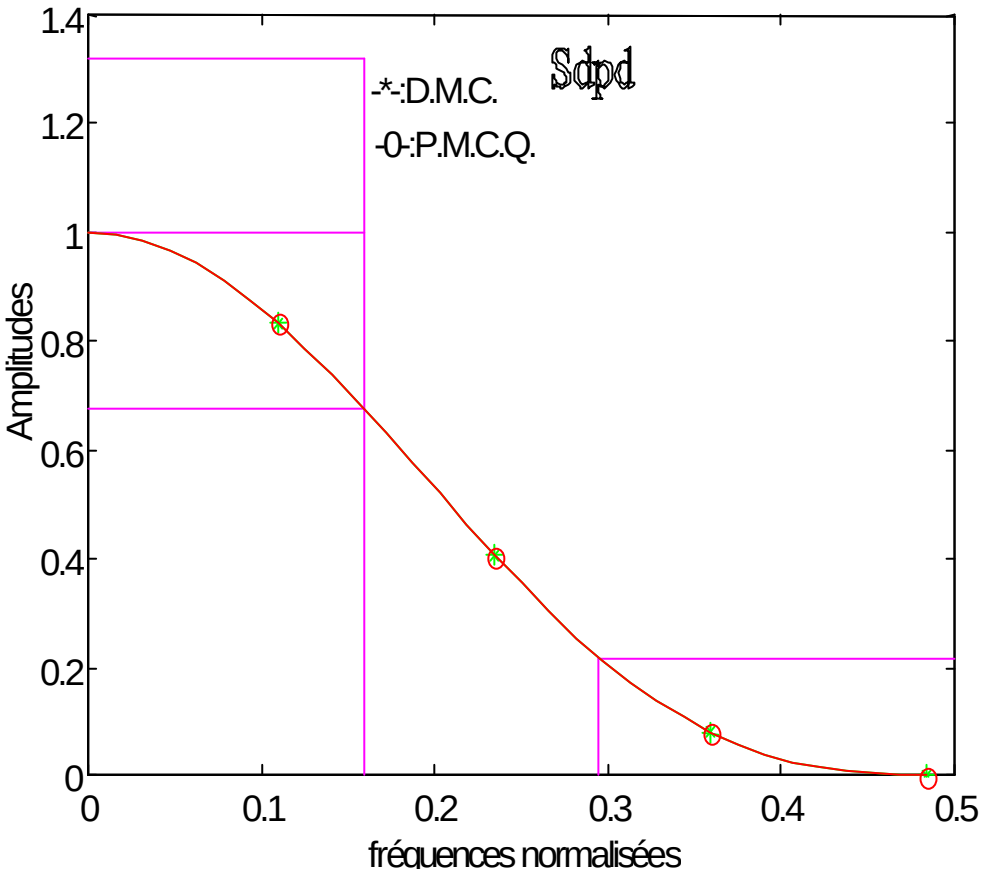


Fig.25. DMC et PMCQ en représentation Sdpd sur lm =8 bits

Ces figures 23, 24 et 25 montrent que tous les réponses en amplitude des filtres rentrent dans le du gabarit. Par ailleurs, nous avons les plus grandes valeurs des erreurs maximales dans la BP et la BA en représentation Sdpd comparées à celles en Vfx et en Vfl. Dans le paragraphe suivant, nous allons choisir un mot de longueur plus grande lm = 16 bits afin de mieux exploiter la rapidité de convergence de la méthode DMC et de voir ces performances dans l'espace discret correspondant à cette longueur de mot.

VII.2. FILTRE 2 :

N=8.
lm = 16 bits.
wp = 0.159.
ws = 0.295.

La synthèse du filtre choisi, nous a donné les résultats qui sont groupés dans les tableaux suivants :

	D.M.C.				P.M.C.Q.	
	Ems	Emp	Ema	Tps	Ems	Tps
Vfx	0.03318	0.08290	0.09250	3.30 s.	0.03743	11.09s.
Vfl	0.03317	0.08295	0.09237	2.85 s.	0.03744	7.52 s.
Sdpd	0.04520	0.07144	0.15043	0.66 s.	0.06242	2.09 s.

Tableau15 : Représentation de l'erreur quadratique moyenne par les deux méthodes (D.M.C. et P.M.C.Q.) dans les différentes représentations à lm =16 bits.

Coef. Filt. De longueur :8	Vfx D.M.C. | P.M.C.Q.		Vfl D.M.C. | P.M.C.Q.		Sdpd D.M.C. | P.M.C.Q.	
H(0) =h(7)	-0.05349731	-0.06198120	-0.05351257	-0.06199645	-0.046875	-0.062011
h(1)=h(6)	-0.04498291	-0.04998779	-0.04496765	-0.04998779	-0.058593	-0.046875
H(2) =h(5)	0.16207885	0.17230224	0.16210937	0.17230224	0.156250	0.187500
h(3)=h(4)	0.41091918	0.41091918	0.41088867	0.41088867	0.437500	0.437500

Tableau16 : Tableau de coefficients du filtre obtenu par D.M.C. à chaque représentation pour lm=16 bits.

Les remarques faites pour le filtre 1 (VII.1) sont valables pour ce filtre. A part la représentation en Sdpd, dans cet exemple choisi, nous remarquons que les erreurs quadratiques moyennes des filtres conçus par la méthode P.M.C.Q. sont plus grandes que celles de la méthode DMC.

De plus, le temps de synthèse de la méthode D.M.C. est plus petit que celui de la méthode de P.M.C.Q. Dans les six cas de filtre du tableau 14, le meilleur filtre PMCQ au sens de l'erreur quadratique moyenne est celui défini dans la représentation à virgule fixe ou flottante.

Les coefficients retrouvés par les deux méthodes de synthèse dans les deux représentations Vfx et Vfl sont identiques, et très différents de ceux en représentation Sdpd. Comparé au filtre 1 où la conception s'est effectuée sur un mot de 8 bits, nous remarquons que les erreurs Ems de la méthode DMC pour le même filtre sur 16 bits sont plus petites, à part celle en Vfx. L'erreur Ems sur 8 bits a été de 0.0330, tandis que celle sur 16 bits est de 0.03318. Ce cas particulier arrive souvent où l'augmentation de mot n'améliore pas les performances.

A partir des coefficients du tableau 16 , nous avons schématisé l'allure des filtres correspondants dans chaque représentation binaire. Nous avons obtenu les figures suivantes :

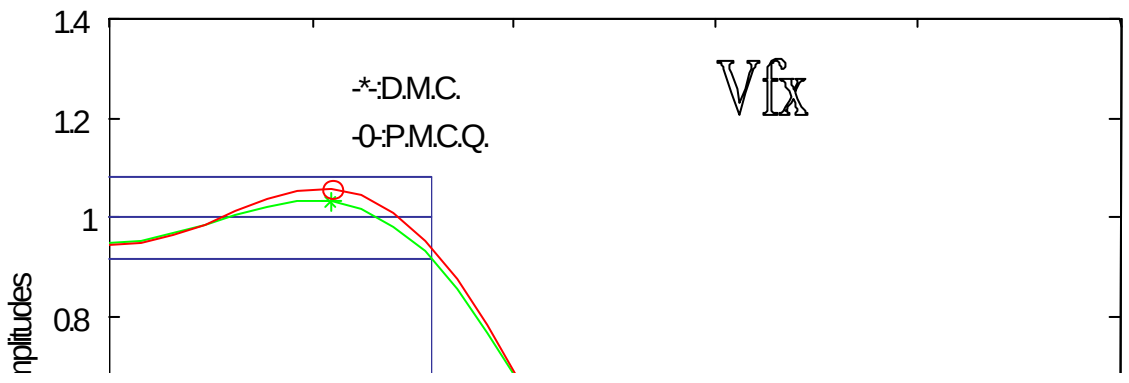


Fig.26. DMC et PMCQ en représentation Vfx sur lm =16 bits

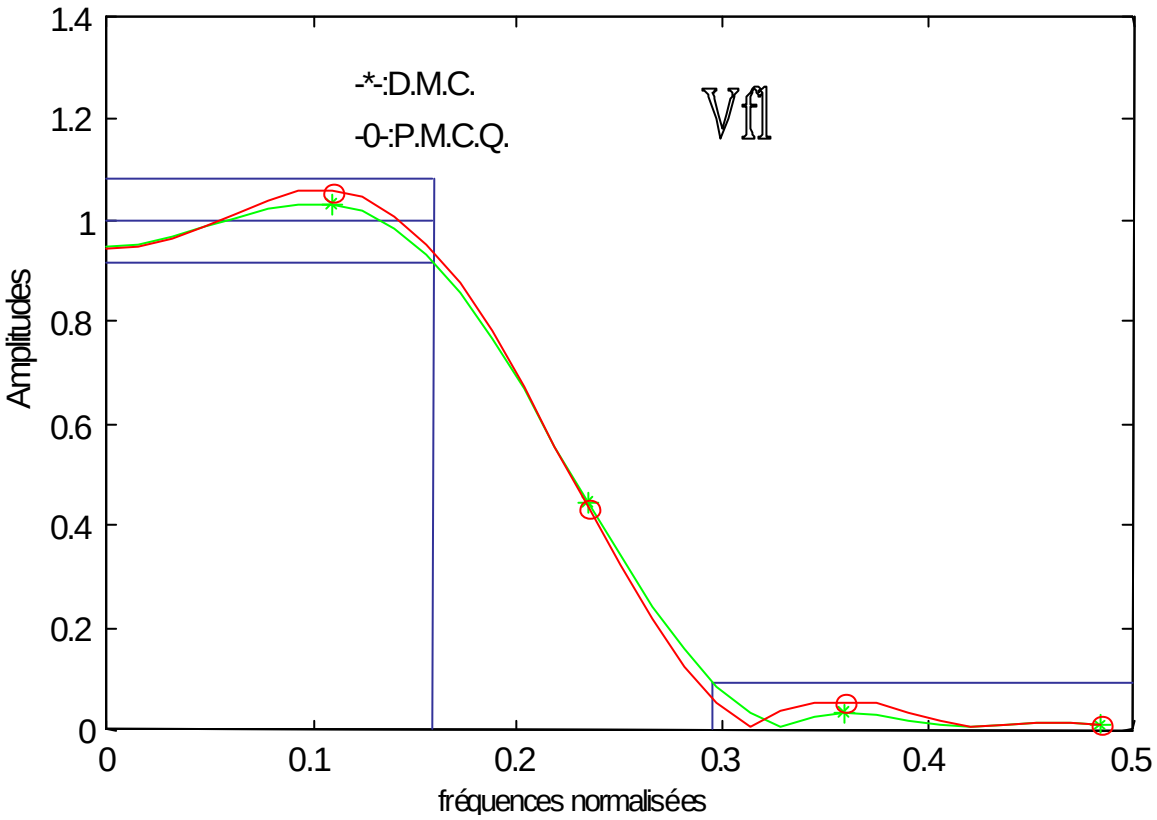


Fig.27. DMC et PMCQ en représentation Vfl sur lm =16 bits

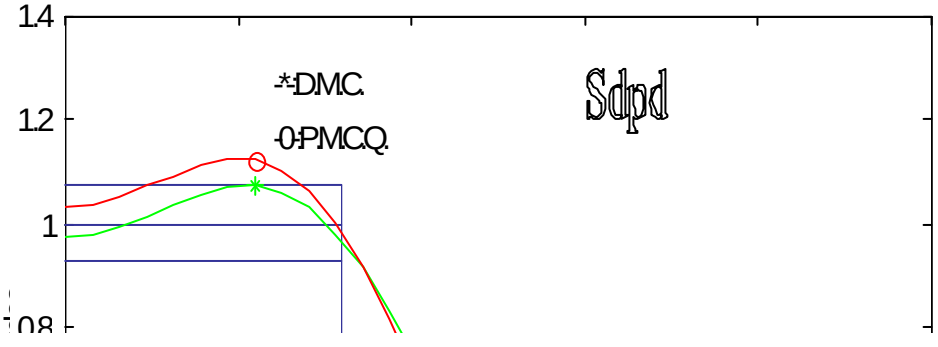


Fig.28. DMC et PMCQ en représentation Sdpd sur lm =16 bits

Concernant la représentation Sdpd, nous remarquons que la réponse en amplitude du filtre de PMCQ sort du gabarit défini pas celle de DMC. Afin de confirmer ces résultats, une extension aux autres cas de filtres est faite dans les deux paragraphes suivants. Le premier correspond à un filtre de longueur identique '8', tandis que le second correspond à un filtre de longueur supérieur '16'.

VII.3. FILTRE 3 :

N=8.
lm = 8 bits.
wp = 0.45.
ws = 0.48.

La synthèse du filtre choisi, nous a donné les résultats qui sont groupés dans les tableaux suivants :

	D.M.C.				P.M.C.Q.	
	Ems	Emp	Ema	Tps	Ems	Tps
Vfx	0.09678	0.37210	0.26767	0.05 s.	0.20987	0.05 s.
Vfl	0.09680	0.36501	0.27060	0.05 s.	0.20765	0.05 s.
Sdpd	0.15065	0.26895	0.35587	0.05 s.	0.25489	0.05 s.

Tableau17 : Représentation de l'erreur quadratique moyenne par les deux méthodes (D.M.C. et P.M.C.Q.) dans les différentes représentations à lm =8 bits.

Coef. De filtre de longueur :8	Vfx		Vfl		Sdpd	
	D.M.C.	P.M.C.Q.	D.M.C.	P.M.C.Q.	D.M.C.	P.M.C.Q.
h(0) =h(7)	-0.062500	0	-0.0625000	-0.00341796875	-0.125	0
h(1)=h(6)	0.1015625	0.2187500	0.1015625	0.21875000000	0.125	0.250
h(2) =h(5)	-0.1953125	-0.2031250	-0.2031250	-0.20312500000	-0.250	-0.250
h(3)=h(4)	0.625000	0.6328125	0.6250000	0.62500000000	0.625	0.625

Tableau18 : Tableau de coefficients du filtre obtenu par D.M.C. à chaque représentation pour lm=8 bits.

Le tableau 17 présente les performances et le temps de calcul de DMC et PMCQ par arrondissement dans les 3 représentations binaires. Les mêmes remarques faites à l'exemple 1 sont vérifiées pour ce cas de filtre. L'erreur Ems de DMC est inférieur à celle de PMCQ pour un temps de calcul identique.
Pour la méthode DMC nous avons le qualité de sortie (Ems| tps) de (0.09678| 0.05s), (0.09680| 0.05s) et (0.15065| 0.05s) respectivement en Vfx, Vfl et Sdpd. Nous remarquons aussi que Vfx offre le meilleur résultat pour un temps de calcul identique à celui de Vfl et de Sdpd.

A partir des coefficients du tableau 18, nous avons schématisé l'allure des filtres retrouvés à chaque représentation. Nous avons obtenu les figures suivantes :

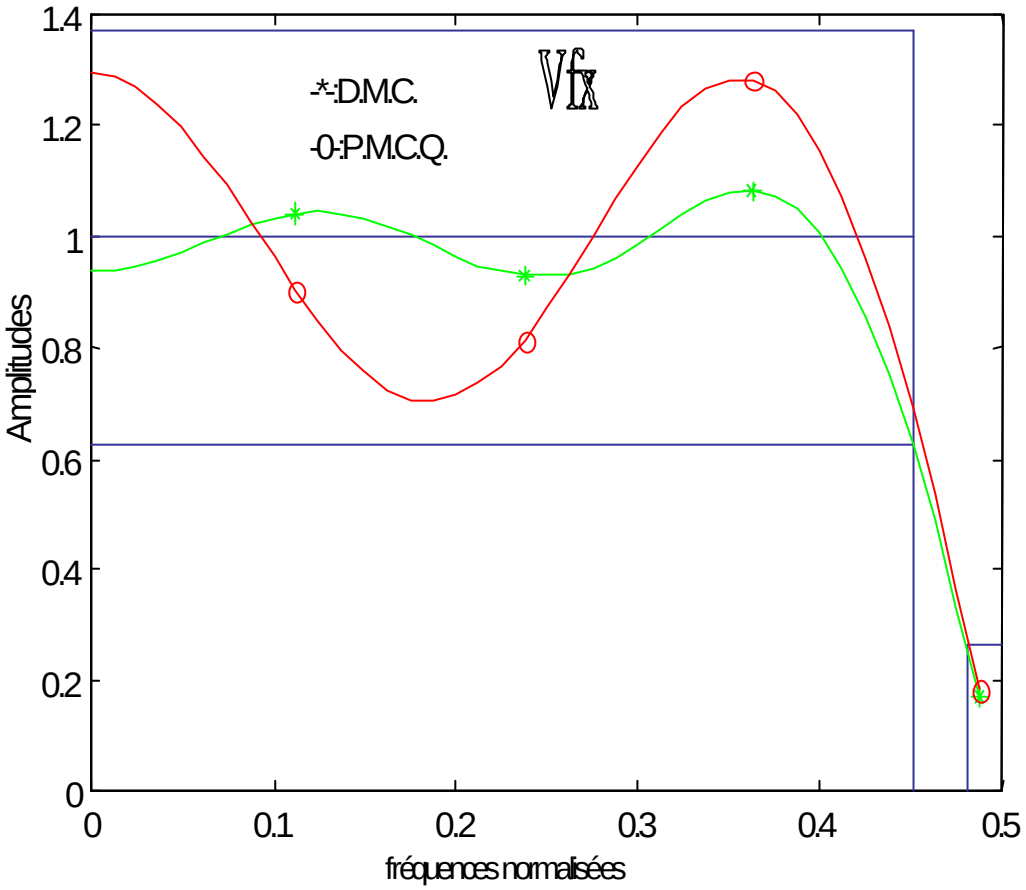


Fig.29. DMC et PMCQ en représentation Vfx sur lm =8 bits

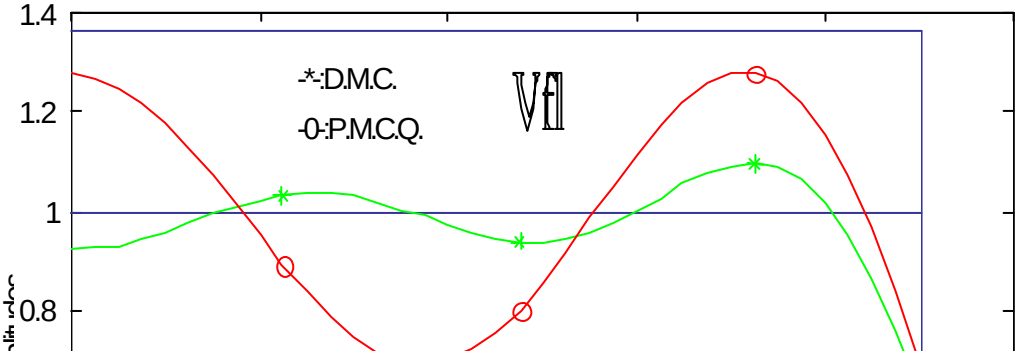


Fig.30. DMC et PMCQ en représentation Vfl sur lm =8 bits

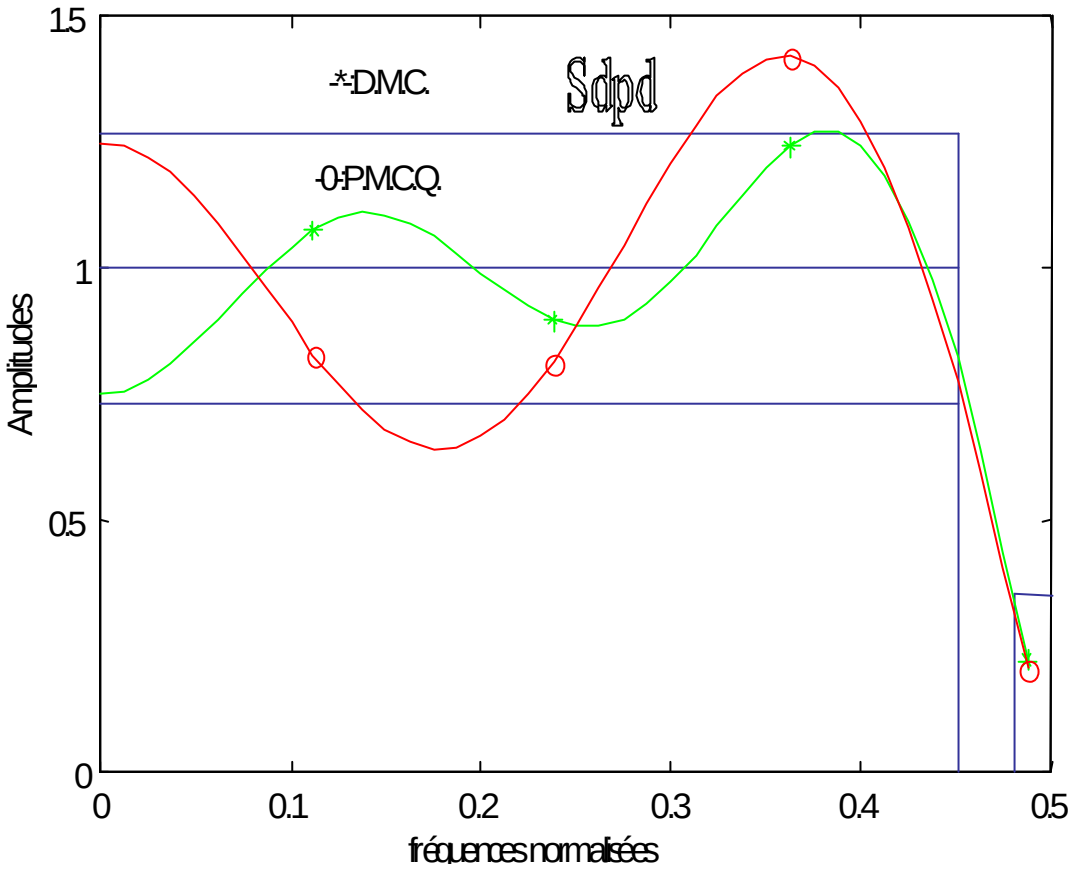


Fig.31. DMC et PMCQ en représentation Sdpd sur lm = 8 bits

Nous remarquons de la figure 31 que la réponse en amplitude du filtre de PMCQ par arrondissement en représentation Sdpd sort du gabarit défini par celle de la méthode DMC.

VII.4. FILTRE 4 :

N=16.
lm = 8 bits.
wp = 0.1.
ws = 0.25.

La synthèse du filtre choisi, nous a donné les résultats qui sont groupés dans les tableaux suivants :

	D.M.C.				P.M.C.Q.	
	Ems	Emp	Ema	Tps	Ems	tps
Vfx	0.02697	0.08331	0.06629	0.05 s.	0.01203	0.05 s.
Vfl	0.01861	0.03539	0.07112	0.05 s.	0.01896	0.05 s.
Sdpd	0.12856	0.13976	0.35355	0.05 s.	0.14280	0.05 s.

Tableau19 : Représentation de l'erreur quadratique moyenne par les deux méthodes (D.M.C. et P.M.C.Q.) dans les différentes représentations à lm =8 bits.

Coef. de filtre de longueur :8	Vfx		Vfl		Sdpd	
	D.M.C.	P.M.C.Q.	D.M.C.	P.M.C.Q.	D.M.C.	P.M.C.Q.
H(0) =h(15)	-0.0078125	0.0078125	-0.00244140625	0.0087890625	0	0
h(1)=h(14)	0	0.0156250	0.00488281250	0.0126953125	0	0
H(2) =h(13)	0	-0.0078125	0.00097656250	-0.0068359375	0	0
h(3)=h(12)	-0.0234375	-0.0390625	-0.02734375000	-0.0429687500	0	0
h(4)=h(11)	-0.0312500	-0.0468750	-0.03906250000	-0.0429687500	0	0
h(5)=h(10)	0.0390625	0.0468750	0.03125000000	0.0429687500	0	0
h(6)=h(9)	0.1796875	0.2031250	0.18750000000	0.2031250000	0.125	0.250
h(7)=h(8)	0.3281250	0.3281250	0.34375000000	0.3437500000	0.375	0.375

Tableau20 : Tableau de coefficients du filtre obtenu par D.M.C. à chaque représentation pour lm=8 bits.

Puisque la méthode D.M.C. présente l'intérêt de la rapidité, nous avons choisi la conception d'un filtre de longueur 16 (à 16 coefficients) afin de mieux interpréter le choix de la représentation binaire utilisée.

Le tableau 19 présente les performances et le temps de calcul de DMC et PMCQ par arrondissement dans les représentations binaires. A part en Vfx, l'erreur Ems de DMC est inférieur à celle de PMCQ pour un temps de calcul identique. Pour les six cas de filtres le filtre PMCQ en Vfx présente la meilleure approximation dans le sens de Ems. Le qualité de sortie (Ems| tps) pour la méthode DMC sont de (0.02697| 0.05s), (0.01861| 0.05s) et (0.12856| 0.05s) respectivement en Vfx, Vfl et Sdpd. Nous remarquons aussi que Vfl offre le meilleur résultat pour un temps de calcul identique à celui de Vfx et de Sdpd.

A partir des coefficients du tableau20, nus avons l'allure des filtres suivants :

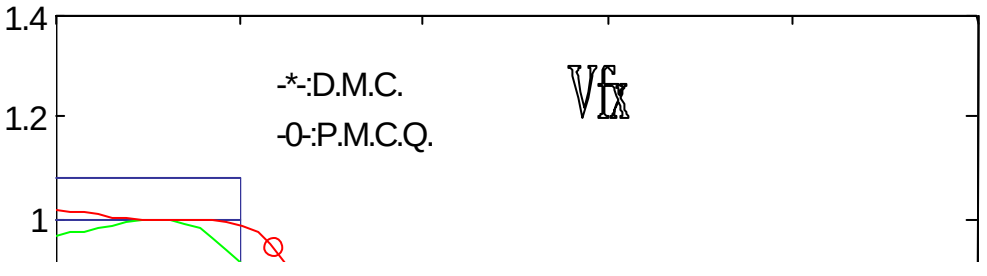


Fig.32. DMC et PMCQ en représentation Vfx sur lm = 8 bits

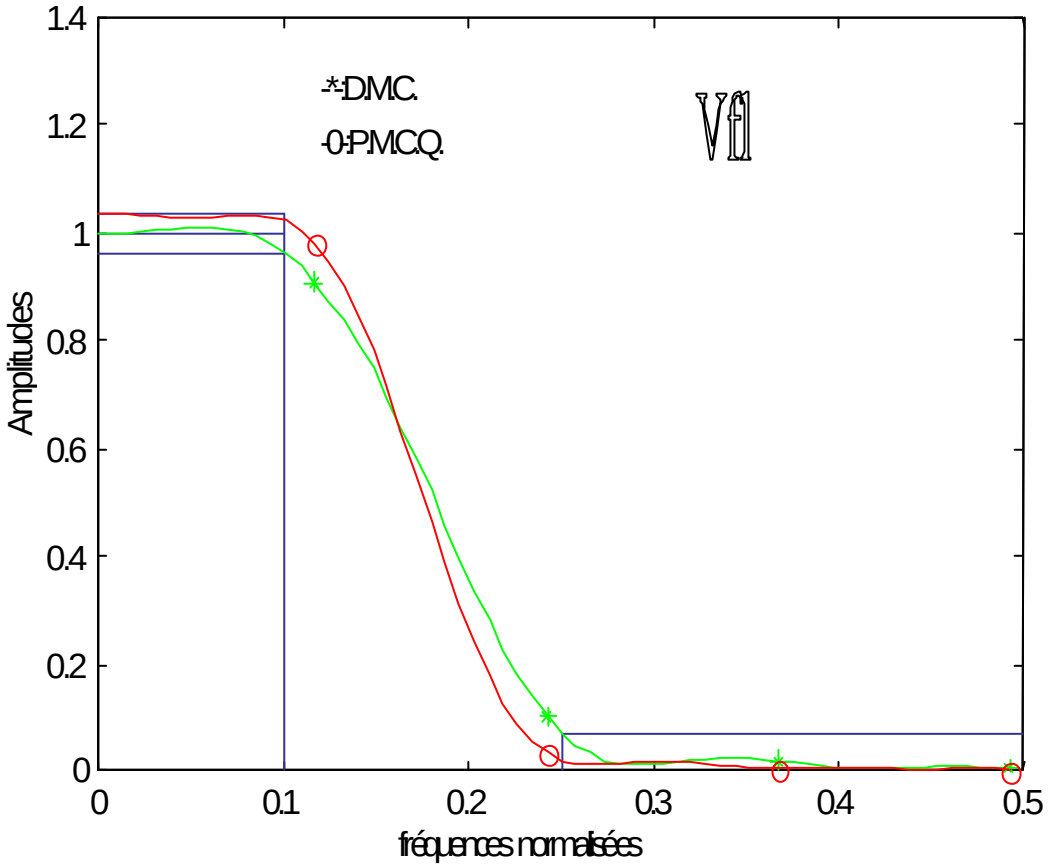


Fig.33. DMC et PMCQ en représentation Vfl sur lm = 8 bits

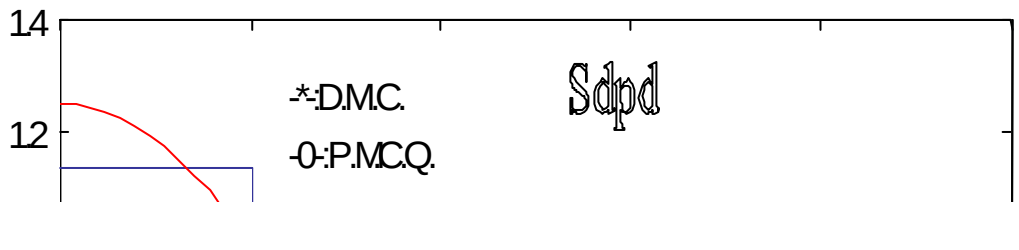


Fig.34. DMC et PMCQ en représentation Sdpd sur lm = 8 bits

Les mêmes remarques faites aux figures des exemples précédents sont vérifiées pour cet exemple; la réponse en amplitude du filtre de la méthode PMCQ dans la représentation Sdpd sort du gabarit défini par celle du filtre obtenu par la méthode DMC en même représentation.

VIII. ETUDE DES RESULTATS DE LA METHODE DMC :

Une observation primitive des résultats de la méthode D.M.C. pour les quatre filtres dans les différentes représentations binaires montre que les performances ne sont pas cohérentes. Dans 3 exemples sur 4, l'erreur Ems entre l'amplitude du filtre conçu par la méthode DMC et celle de la réponse en fréquence désirée, est plus petite que celle de PMCQ par arrondissement avec un temps de calcul égal ou plus petit. Une étude comparative de la qualité de sortie (précision | temps de calcul) pour la méthode DMC dans les trois représentations binaires, montre que la représentation SDPD présente l'erreur Ems la plus grande, tandis qu'une des deux représentations Vfx ou Vfl offre le meilleur résultat dépendamment du cas de filtre à concevoir. La Vfl procure à chaque fois un temps de calcul plus petit que celui de Vfx. Dans l'exemple 4, où la longueur de filtre est plus grande que celle de ses précédents exemples, nous remarquons que l'erreur Ems du filtre de DMC dans une représentation en Vfx présente une erreur plus grande que celle de PMCQ. Le choix de la représentation binaire dépend des spécifications requises, mais, compte tenu du qualité de sortie (Ems | tps) dans les précédents exemples, nous recommandons l'utilisation de la représentation à virgule flottante.

Dans le paragraphe suivant, nous allons présenter un récapitulatif du choix de la représentation binaire dans la synthèse des filtres numériques dans l'espace discret utilisant les deux méthodes RA et DMC. Des conclusions seront données après le diagnostic des résultats de ce chapitre et une étude comparative avec ceux obtenu par le laboratoire Signaux et Systèmes.

IX. ETUDE COMPARATIVE AVEC LES TRAVAUX DE M. BOULERIAL [31] :

Pour des raisons de complexité algorithmique, le laboratoire Signaux et Systèmes a élaboré les deux méthodes R.A. et D.M.C., concernant les travaux de la synthèse des filtres numériques, en représentation binaire Sdpd. Cela est du à ses avantages décrits en [31] tels que la faible erreur de quantification au début de la dynamique entre (- 0.5 et + 0.5) et le nombre réduit des valeurs admissibles par rapport aux représentations Vfx et Vfl, pour une même longueur de mot machine 'lm'.

Notre propos dans ce chapitre a été d'étendre ces travaux dans les représentations binaires en virgule fixe et en virgule flottante afin d'analyser le comportement des performances comparativement à celui de la complexité algorithmique découlant de l'utilisation de chaque représentation binaire. En effet, nous avons étudier le qualité de sortie (précision| temps de calcul) pour les deux méthodes dans les trois représentations.

D'après les résultats précédemment donnés, nous constatons l'utilisation de la méthode R.A. avec les représentations binaires Vfx et Vfl a prouvé de meilleure qualité des résultats par rapport à Sdpd, vu la grande représentativité des valeurs discrètes. Tandis que la complexité numérique est devenu plus grande et le temps de calcul devient prohibitif pour un espace discret à 'lm > 8 bits'. Le choix de la représentation binaire adéquate dépend des spécifications de la qualité de sortie requise (précision | temps de calcul), relative aux ressources matérielles disponibles.

Notre but dans ce travail est de calculer les filtres RIF à phase linéaire qui présente la meilleure approximation dans le sens de Ems. Par conséquent, nous avons choisi dans la suite de notre travail d'utiliser la représentation Vfx pour la version améliorée de la méthode RA nommée méthode de Recherche Séquentielle et Progressive 'R.S.P.' cette méthode permet de réduire la complexité algorithmique de la méthode RA. Le choix de la Vfx est du à sa grande représentativité discrète qui procure de meilleures performances.

Contrairement à RA, la méthode D.M.C. présente une faible complexité algorithmique. L'intérêt de notre travail dans ce chapitre a été d'exploiter la rapidité de cette méthode afin d'évaluer ses performances dans les différentes représentations binaires. Dans les exemples précédents, les meilleurs résultats que nous avions obtenu sont ceux en représentations binaires Vfx et Vfl. Par contre, nous remarquons que la méthode PMCQ par arrondissement peut donner des résultats meilleurs que ceux de DMC (exemple4). Cela est du à la dépendance séquentielle des coefficients dans la méthode DMC. Le calcul d'un coefficient dépend de l'erreur sur l'amplitude du filtre du au calcul du coefficient précédent. Alors, l'erreur finale du filtre correspond à celle du dernier coefficient. Cette erreur n'est pas réinjectée pour la reoptimisation du premier coefficient. Notre but dans la suite de ce travail, est l'amélioration des performances de DMC en ajoutant d'autre itérations. Cette méthode globale sera nommée méthode Directe par Moindre Carré Itérative ' D.M.C.I.'. Nous allons dans le chapitre 3 tester les performances de cette méthode utilisant les trois représentations avant de se prononcer pour une représentation adéquate.

X. CONCLUSION :

Ce chapitre est une extension des travaux de Boulerial [31]u laboratoire Signaux et Systèmes, dans la synthèse des filtres numériques dans l'espace discret des coefficients, concernant les méthodes RA et DMC dans les représentations en virgule fixe et en virgule flottante. Nous avons montré que l'utilisation de la représentation Sdpd dans la méthode RA comparativement à Vfx et Vfl, a permis un gain acceptable en complexité algorithmique mais avec une grande perte de performances. Tandis que pour la méthode DMC dans Sdpd, nous avons constaté une grande perte de

performances pour un temps de calcul identique à celui de Vfx et Vfl. De plus, les performances de la méthode PMCQ par arrondissement sont, en général moindres que ceux de DMC pour un temps de calcul au moins égal.

A ce propos dans la suite de ce travail, nous allons améliorer la qualité des résultats de la méthode DMC en ajoutant une approche itérative, et réduire la complexité algorithmique de RA utilisant la représentation Vfx en effectuant une nouvelle stratégie de branchement.

Chapitre III

METHODE DIRECTE PAR MOINDRE CARRE ITERATIVE 'D.M.C.I.'

VIII. INTRODUCTION :

Il a été montré dans le chapitre précédent que la méthode D.M.C. est très rapide dans la synthèse des filtres numériques, seulement, les résultats obtenus sont moins performants par rapport à ceux de la recherche arborescente. A ce propos, une nouvelle méthode nommée «méthode d'optimisation Directe par Moindre Carré Itérative 'D.M.C.I.' » est présentée dans ce chapitre, afin d'améliorer les résultats obtenus par la méthode DMC dans le sens de l'erreur quadratique moyenne.

La méthode D.M.C.I. que nous proposons ci-dessous, est basée sur une approche itérative qui améliore les résultats de DMC. Cette approche constitue un algorithme annexé à la fin de celui de DMC. Il effectue une recherche exhaustive de la meilleure solution dans le sens de Ems dans un sous espace discret de rayon prédéfini autour de chaque coefficient en plusieurs itérations.

Avant d'exposer la démarche de cette nouvelle méthode DMCI, une étude mathématique de l'espace discret et de ses caractéristiques, basée sur les travaux de Lim [14], a été menée dans ce chapitre. Ensuite, nous proposons des exemples afin de tester les capacités. Nous avons nommé par 'optimisabilité', la possibilité d'améliorer les performances des filtres. Ce terme a été introduit dans la littérature scientifique par Lim[14].

IX. OPTIMISABILITE DISCRETE :

II.1. POSITION DU PROBLEME :

Dans [14], Lim a introduit, le concept d'optimisabilité discrète qui sert à la mesure du gain en qualité des résultats par une optimisation discrète comparée à la simple quantification des coefficients obtenus par Parks- Mc Clellan. Nous introduisons ce concept dans la suite de notre travail pour mieux expliquer la démarche de la méthode DMCI. A ce propos, nous donnons la notion de base de l'optimisabilité discrète dans le sens de Ems.

Soit la réponse en fréquence $H(e^{jw})$ d'un filtre RIF à phase linéaire de longueur N exprimée en fonction trigonométrique de la fréquence w. En omettant le facteur de phase linéaire $e^{-jw((N-1)/2)}$ à partir de l'eq. 25, la réponse en fréquence pour le cas de filtre 1 (réponse impulsionnelle symétrique et N impaire) est donnée par:

$$H(e^{jw}) = a(0) + 2.\sum_{n=0}^{\frac{N-1}{2}} a(n)\cos wn. \qquad (50)$$

et $$h(n) = a((N-1)/2-n) = h(N-1-n). \qquad (51)$$

avec $n = 0, 1, 2, \ldots\ldots, \frac{N-1}{2}$

L'extension aux autres cas de filtres peut être faite facilement. Le problème de la conception de filtre est d'obtenir les coefficients a(n) telle que $H(e^{jw})$ soit la

meilleure approximation dans le sens de Ems de la fonction désirée $H_D(e^{jw})$ dans les bandes passante et atténuée.
Alors que la valeur de $H(e^{jw})$ est sous les contraintes suivantes :

$$H(e^{jw}) \le H_D(e^{jw}) + \delta k(e^{jw}) \quad (52.a)$$
$$H(e^{jw}) \ge H_D(e^{jw}) - \delta k(e^{jw}) \quad (52.b)$$

Où $\delta k(e^{jw})$ est la tolérance (la différence maximale entre H et H_D permise par l'utilisateur du filtre dans la bande passante et la bande atténuée).

La raison du problème de conception de filtre numérique dans un espace à longueur de mot finie 'lm' est l'ajustement de la taille des coefficients a(n) à 'lm' bits tout en respectant les contraintes (52). Dans ce contexte, un autre critère est possible pour l'optimisation de a(n) est de réduire la puissance de l'erreur de sortie de Fig.35.

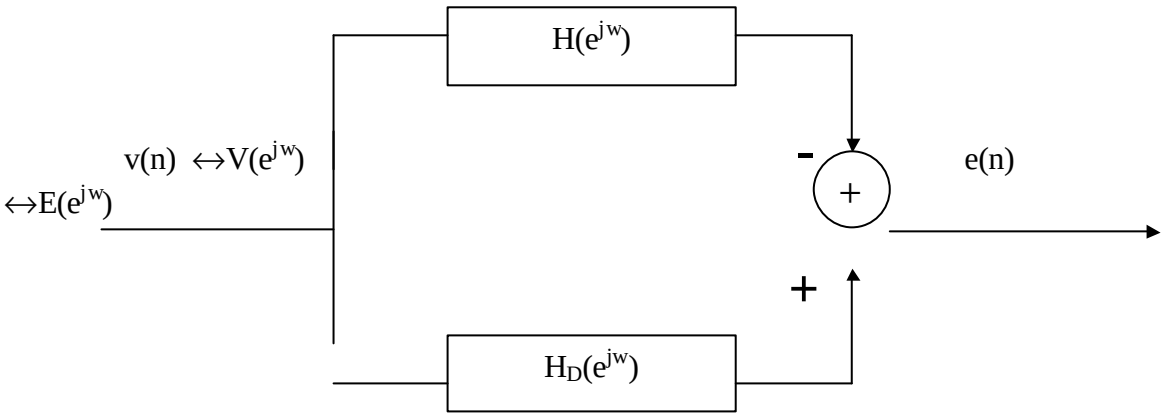


Fig. 35 Schéma synoptique de la minimisation de la puissance de l'erreur

Dans cette figure, v(n) et e(n) sont respectivement, le signal d'entrée et le signal erreur. $V(e^{jw})$ et $E(e^{jw})$ sont respectivement les spectre de fréquence de v(n) et e(n). Il a été montré que

$$|E(e^{jw})|^2 = |V(e^{jw})|^2 . | H_D(e^{jw}) - H(e^{jw})|^2 . \quad (53)$$

puisque

$$\sum_{n=0}^{\infty} e(n)^2 \propto \int_0^p |E(e^{jw})|^2 \, dw, \quad (54)$$

alors, minimiser la puissance d'erreur de sortie implique la minimisation de l'erreur quadratique pondérée 'J' qui s'écrit sous la forme :

$$J = \int_0^p W(e^{jw}) . | H_D(e^{jw}) - H(e^{jw})|^2 \, dw \quad (55)$$

Avec $W(e^{jw}) = |V(e^{jw})|^2 .$

Le facteur $| H_D(e^{jw}) - H(e^{jw})|^2$ et $W(e^{jw})$ peuvent être interprété respectivement comme le carré de l'erreur de la réponse en fréquence et la fonction de pondération de l'erreur de la réponse en fréquence. L'intégrale (55) peut correspondre à une sommation comme suit :

$$J = \sum_i W(e^{jwi}). \mid H_D(e^{jw}) - H(e^{jwi})\mid^2 . \qquad (56)$$

Pour des raisons de simplicité, nous n'avons pas pris en compte la constante de proportionalité dans (56). En écrivant (50) sous forme de vecteur nous aurons :

$$H(e^{jw}) = C(e^{jw})^T.a \qquad (57)$$

Avec $C(e^{jw})^T$ =[1, 2cosw, 2cos2w,……2 cosw(N-1)/2]

Et a^T =[a(0), a(1), ……..,a((N-1)/2)].

En remplaçant (57) dans (56) on aura

$$J = \sum_i \{W(e^{jwi}). H_D(e^{jw})^2 - 2. W(e^{jwi}). H_D(e^{jw}). C(e^{jwi})^T.a + W(e^{jwi}). a^T.C(e^{jwi}).C(e^{jwi})^T.a\} \quad (58)$$

XVII. L'objet du travail de ce chapitre est la minimisation de J en utilisant la méthode directe par moindre carré itérative qui sera décrite dans le paragraphe III. Cette méthode se base sur la notion d'optimisabilité des coefficients qui sera l'objet du paragraphe suivant.

II.2. OPTIMISABILITE DISCRETE DANS LA CONCEPTION DE FILTRE AU SENS DE L'ERREUR QUADRATIQUE MOYENNE 'Ems' :

Dans ce paragraphe, nous présentons la notion d'optimisabilité. Le principe de base est comme suit.
Après obtention des coefficients (a(n) avec n :0, 1,…N-1) à précision infinie par PMC, nous fixons le premier coefficient (a(0)) à une autre valeur loin de son optimum. Cela induit une augmentation de l'erreur. nous reoptimisons les autres valeurs des coefficients (a(n) avec n :1, 2,….N-1) pour compenser la variation de la valeur du précédent coefficient. Cette erreur diminue. Nous appelons par la mesure d'optimisabilité discrète, la mesure de changement des autres valeurs des coefficients dans le but de compenser la fixation de la valeur du coefficient loin de sa valeur optimale. Ce changement ne peut pas être obtenue sans l'optimisation discrète. Soit :

$$\mathbf{a = aopt + q.} \qquad \mathbf{(59.a.)}$$

avec 'θ' l'erreur commise sur a et 'aopt' la valeur optimale de 'a' qui minimise (58). aopt satisfait l'équation :

$$[\sum_i \{ W(e^{jwi}).C(e^{jwi}).C(e^{jwi})^T\}].aopt = \sum_i \{. W(e^{jwi}). C(e^{jwi}). H_D(e^{jw})\} \qquad (59.b.)$$

et $Jopt = \{\sum_i W(e^{jwi}). H_D(e^{jw})^2\} - aopt^T.[\sum_i \{W(e^{jwi}). C(e^{jwi}).C(e^{jwi})^T\}].aopt.$ (59.c.)

Jopt est la valeur de J quand a= aopt. Posons

$$C=\sum_{i} \{W(e^{jwi}).\, C(e^{jwi}).C(e^{jwi})^{T}.\, W(e^{jwi})\} \qquad (60)$$

'C' est une matrice définie positive et symétrique. En remplaçant (59) et (60) dans (58) et en notant par $aopt^{T}.C.\theta = \theta^{T}.C.aopt^{T}$ on obtient :

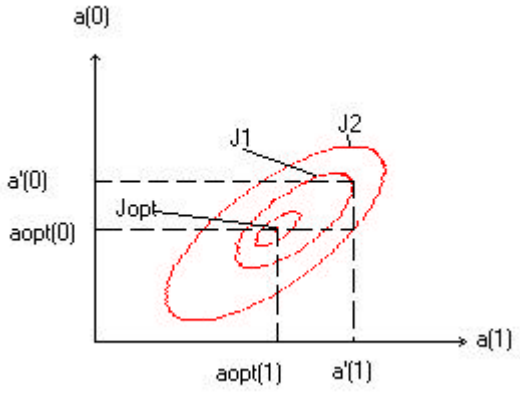
$$J=Jopt+ \theta^{T}.C.\theta \qquad (61)$$

Le principe de base pour obtenir une optimisabilité discrète du problème dans la conception de filtre peut être illustré en utilisant un exemple à deux coefficients.

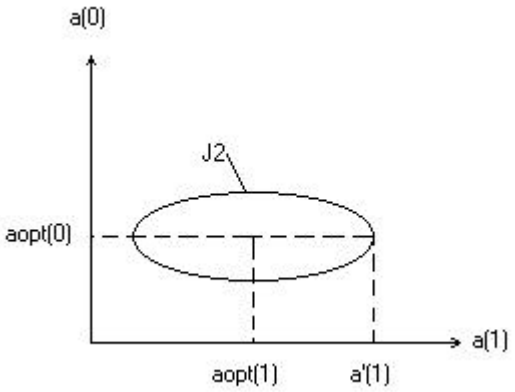
Soit : $a=[a(0)\ a(1)]^{T}$ (62.a.)

et $aopt=[aopt(0)\ aopt(1)]^{T}$ (62.b.)

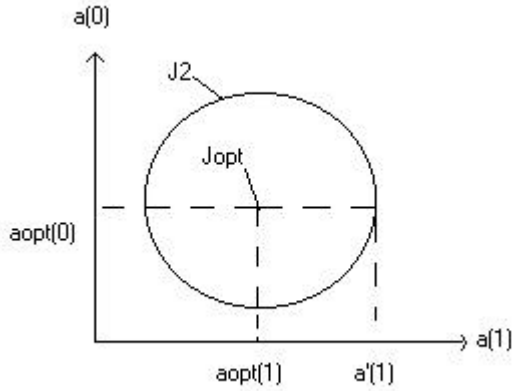
En remplaçant (62) dans (61), le contour d'une valeur donnée de J est une ellipse comme le montre la Fig. 36 où J2>J1. Considérons Fig. 36.a. En déplaçant la valeur de a(1) de aopt(1) vers a'(1) en gardant a(0) fixe à aopt(0), J augmente de Jopt vers J2. Si a(1) est fixée à a'(1), un processus d'optimisation va déplacer a(0) de aopt(0) vers a'(0) pour minimiser J. Considérons à présent Fig. 36.b et Fig. 36.c. En déplaçant la valeur de a(1) de aopt(1) vers a'(1), J augmente vers la valeur de J2. En fixant la valeur de a(1) à a'(1), il est



(a)



(b)



(c)

Fig.36 les contours elliptiques de J

clair qu'aucun processus d'optimisation ne peut choisir une valeur de a(0) pour réduire la valeur de J, car la valeur optimale de a(0) pour n'importe quelle valeur de a(1) est aopt(0). L'effet de déplacer la valeur de a(1) de son optimum ne peut être compensé en changeant la valeur de a(0).
Après observation de la fig. 36, nous concluons qu'un problème de compensation de filtre possède une faible optimisabilité discrète si

- Les axes principaux des contours elliptiques de J ont une longueur égale « cercle » (Fig.36.c).
- Les axes principaux des ellipses sont parallèles aux axes de a(0) et a(1) (Fig. 36.b).

En général, (61) peut être écrite comme suit :

$$J=Jopt+ \theta^T.M.S.M.\theta \qquad (63)$$

Avec
S : matrice spectrale (matrice diagonale dont les éléments sont les valeurs propres de C).
M : matrice à modèle normalisée (matrice dont les colonnes sont les vecteurs propres normalisés de C).
Equation (63) montre que les racines carrés des valeurs propres de C sont les longueurs des principaux axes de l'ellipse. Les éléments des vecteurs propres normalisés de C sont les directions cosinusoïdales des principaux axes. De là, nous concluons que la conception de filtre possède une faible optimisabilité discrète si

- Les valeurs propres de C sont égales.
- Une grande portion des éléments de M sont soit '0' soit '1'.

Comme cas spécial de ce problème, nous avons $W(e^{jwi}) =1$. Nous avons C matrice diagonale, et les valeurs de tous les éléments de M sont soit '0' soit '1'. Ce qui veut dire que les axes principaux de l'ellipse sont parallèles aux axes de 'a', dans ce cas la solution optimale discrète est l'arrondi P.MC.Q. [14].

Nous nommons par DMCI, Ce processus d'optimisation de filtres. La démarche principale de cette méthode est décrite dans le paragraphe suivant.

III. OPTIMISATION PAR D.M.C.I. :

III.1. INTRODUCTION :

Il a été montré dans [31] que la méthode 'D.M.C.' est plus rapide que la méthode 'R.A.'. La méthode effectue un calcul séquentiel des coefficients du filtre dans un ordre bien déterminé. Il a été aussi montré que les performances des filtres conçus par cette méthode DMC ne sont pas garanties à être égales à ceux de la méthode RA. La solution obtenue peut être un optimum local 'O.L.'. Nous nommons par optimum local la solution pouvant être améliorée.

Notre propos dans ce chapitre est de présenter une nouvelle approche qui exploite la rapidité de la convergence de la méthode 'D.M.C.' et permet de concevoir des filtres approchant les performances obtenus par la méthode 'R.A.'.
En même temps, nous nous proposons de palier aux deux inconvénients suivants :

- l'influence du premier coefficient calculé dans la méthode DMC sur les performances du filtre obtenu ainsi que l'ordre dans lequel sont calculés les coefficients.

- Le temps de calcul prohibitif de la méthode RA lorsque la longueur du filtre est supérieure à 8 pour une longueur de mot machine supérieure à 8 bits [31].

A ce propos, nous avons élaboré la méthode Directe par Moindre Carré Itérative 'D.M.C.I.' qui permet d'améliorer la précision des résultats de la méthode 'D.M.C.' en effectuant avec plusieurs itérations une recherche exhaustive autour de la solution calculée au moyen de la méthode 'D.M.C.' en se basant sur l'optimisabilité discrète des coefficients.

III.2. DESCRIPTION DE L'ALGORITHME :

Nous donnons l'algorithme précédent sous la forme d'un schéma simplifié dans la figure 37 ci-dessous.

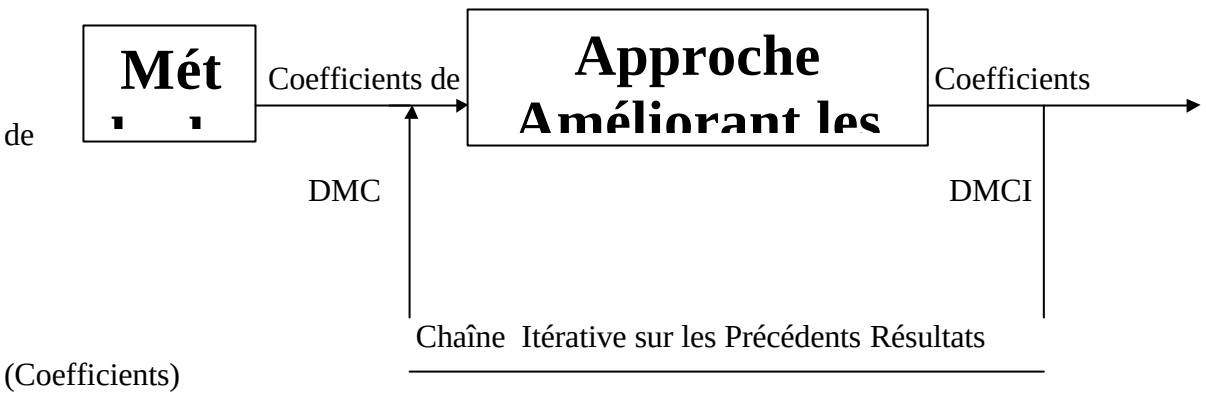


Fig.37. Schéma Représentatif de la Méthode D.M.C.I.

L'algorithme de cette méthode 'D.M.C.I.' se subdivise en deux parties (Fig.37) :

- La première partie représente l'algorithme de la méthode 'D.M.C.' définie par [31],[32], où un calcul séquentiel des coefficients est effectué.
- La deuxième partie représente une méthode itérative qui est basée sur l'amélioration des résultats obtenus dans la première partie (méthode DMC) après 'It' itérations, en faisant le balayage des valeurs discrètes dans un espace de rayon 'v' centré par le coefficient de départ (coefficient calculé par DMC).

Pour des raisons de simplicité pour expliquer la méthode, nous choisissons l'exemple simple suivant :

Le calcul d'un filtre de longueur de deux (deux coefficients) {h(0), h(1)} dans un espace discret constitué de 'va' valeurs admissibles.

Les étapes de la méthode 'D.M.C.I.' sont les suivantes :

1- Au début, nous utiliserons la méthode 'D.M.C.' [32] pour le calcul des coefficients hdmc(0) et hdmc(1). L'erreur Ems du filtre calculé par rapport au filtre idéal est de Edmc.
Posons Er=Edmc.

2- Nous fixons le coefficient hdmc(0) et nous varions le coefficient hdmc(1) dans un intervalle contenant 2 fois 'v' valeurs centré autour du coefficient hdmc(1).

3- A chaque solution discrète, nous calculons l'erreur quadratique moyenne Edmci.
Si Edmci < Er, nous sauvegardons les coefficients correspondants soit {hdmc(0), hr(1)} ,on pose alors Er = Edmci.

4- Nous fixons le coefficient hr(1) et nous varions cette fois-ci le coefficient hdmc(0) dans un intervalle contenant 2 fois 'v' valeurs centré autour du coefficient hdmc(0). ('v' est un nombre de valeurs admissibles inférieur à 'va').

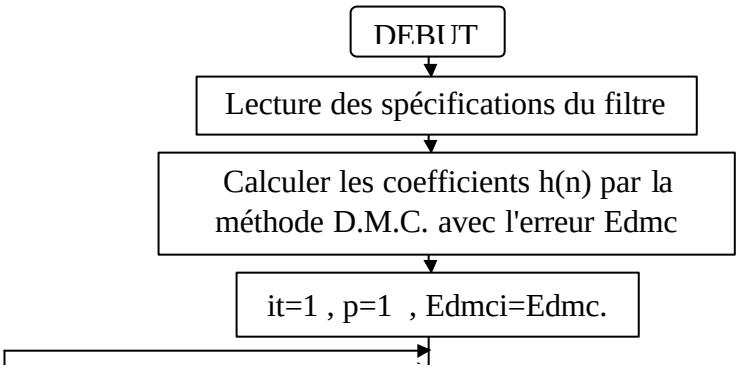
5- A chaque solution discrète, nous calculons l'erreur quadratique moyenne Edmci.
Si Edmci < Er, nous sauvegardons les coefficients correspondants soit {hr(0), hr(1)} et on pose Er = Edmci.

Les étapes 2-5 sont refaites en plusieurs itérations 'It', jusqu'à ce que l'erreur Er n'est plus modifiée. Ainsi les coefficients finaux {hdmci(0), hdmci(1)} sont obtenus par la méthode 'D.M.C.I.'.

Le choix du rayon 'v' est expérimentale, il dépend du nombre des valeurs admissibles dans l'espace discret, de la longueur du filtre et de la représentation binaire. Dans les exemples qui suivent, la valeur maximale de 'v' constitue le quart de l'ensemble des valeurs admissibles.

III.3. ORGANIGRAMME :

Nous avons le fonctionnement de la méthode DMCI dans l'organigramme suivant.



Non

XVIII. Oui

Non

Oui

Non

XIX.

Oui

Non

Oui

Fig. 38. Organigramme de la méthode D.MC.I.

Dans cet organigramme, nous avons utilisé les indices suivants :

2. v : le nombre de valeurs que va prendre le coefficient en balayage dans son scrutin du filtre optimal.

Niter : le nombre d'itérations de cet algorithme.
h(n) : le vecteur des coefficients du filtre à réaliser.
n : indice du coefficient.

N : longueur du filtre.
it : indice de la valeur d'itération.[1, 2,…..Niter]
p : pointeur du numéro de coefficient dans le vecteur h(n).
Edmc : erreur quadratique moyenne minimale du filtre optimal trouvée par la méthode DMC.
Edmci : erreur quadratique moyenne minimale du filtre optimal trouvée par la méthode DMCI.
Ex : erreur quadratique moyenne pour un filtre quelconque de vecteur h(n) trouvée par la méthode DMCI.

III.4. ETUDE DE L'OPTIMALITE DES COEFFICIENTS :

L'inconvénient de la méthode DMC correspondant à l'ordre dont lequel sont considérés les coefficients, persiste dans la méthode DMCI, mais d'une façon moindre. De plus, cette méthode DMCI, comme DMC, est basée sur une optimisation d'un coefficient à la fois, alors, la dépendance des coefficients et leur compensation mutuelle pour retrouver la réponse en fréquence de meilleure approximation dans le sens de l'erreur Ems ne sont pas totalement considérées.

Ces deux problèmes influent d'une façon directe sur les performances de la méthode DMCI d'où les performances de la méthode RA ne sont pas atteintes d'une manière permanente. La stabilité de l'erreur quadratique moyenne après un certain nombre d'itérations à une valeur fixe n'implique pas d'une façon générale, que la solution finale soit égale à celle de la méthode RA. A l'exception des travaux de Lim [14] concernant l'optimisabilité de la solution, une étude théorique sur l'optimalité des résultats s'avère très difficile à mettre en œuvre.

Par ailleurs, une étude statistique a été menée sur plusieurs centaines de filtres afin de mettre en relief les performances de la méthode DMCI. Le repère de comparaison est la méthode de recherche arborescente R.A. en raison de la garantie de ses performances. Pour cette raison, les filtres que nous avons considérés sont ceux de longueur inférieure à ou égal 8, dans un espace discret à $lm \leq 8$ bits.

Les filtres pris en compte sont de spécifications différentes et diverses. L'étude statistique a montré que 98 % des cas traités par DMCI permet de retrouver les mêmes résultats de RA avec un temps de calcul allant jusqu'à 36000 fois plus petit. Puisque la méthode R.A. est prohibitive pour $N > 8$ et $lm > 8$ bits, cette étude statistique ne peut pas être confirmées pour tous les cas de filtres.

L'étude heuristique a montré que la méthode DMCI possède une convergence rapide. A titre d'exemple, pour montrer les performances de la méthode D.M.C.I., nous avons choisi de donner les exemples dans le paragraphe suivant.

IV. RESULTATS DES FILTRES PAR LA METHODE D.M.C.I. :

Un ensemble de 4 filtres typiques RIF à phase linéaire passe bas, repérés par les numéros de 1 à 4, ont été choisis parmi des centaines de filtres avec lesquels la méthode DMCI a été testée. Pour une étude comparative significative avec les méthodes RA et DMC, nous avons choisi des filtres dont les spécifications sont identiques à ceux du chapitre II.

L'intérêt de ce travail est d'étudier la qualité de sortie (précision | complexité algorithmique) de DMCI dans les différentes représentations binaires, comparé à celui de RA et DMC.

IV.1. FILTRE 1 :

N=8.
lm = 8 bits.
wp = 0.159.
ws = 0.295.

La synthèse du filtre choisi, nous a donné les résultats qui sont groupés dans les tableaux suivants :

	D.M.C.I.						D.M.C.		P.M.C.Q.		R.A.	
	Ems	Emp	Ema	Tps	V	It	Ems	Tps	Ems	Tps	Ems	Tps
Vfx	0.0319	0.084	0.089	1.1 s.	10	4	0.0330	0.05 s.	0.0358	0.05 s.	0.0319	26 h.
Vfl	0.0320	0.100	0.060	0.8 s.	10	4	0.0374	0.05 s.	0.0392	0.05 s.	0.0320	18 h.
Sdpd	0.1270	0.320	0.210	0.5 s.	4	1	0.1270	0.05 s.	0.1270	0.05 s.	0.1270	12 s.

Tableau 21 : Représentation de l'erreur quadratique moyenne par les quatre méthodes DMCI, RA, DMC et PMCQ dans les différentes représentations sur lm =8 bits.

Coeff. De filtre de longueur :8	Vfx		Vfl		Sdpd	
	D.M.C.I.	P.M.C.Q.	D.M.C.I.	P.M.C.Q.	D.M.C.I.	P.M.C.Q.
h(0) =h(7)	-0.0546875	-0.0625000	-0.05468750	-0.06250000	0	0
h(1)=h(6)	-0.0390625	-0.0468750	-0.03515625	-0.05078125	0	0
h(2) =h(5)	0.1640625	0.1718750	0.17187500	0.17187500	0.125	0.125
h(3)=h(4)	0.4140625	0.4140625	0.40625000	0.40625000	0.375	0.375

Tableau 22 : Tableau de coefficients du filtre retrouvé par DMCI à chaque représentation sur lm=8 bits.

Le tableau 21 présente l'erreur du filtre conçu au moyen des méthodes DMCI, DMC, RA, et PMCQ par arrondissement dans les représentations Vfx, Vfl et Sdpd. Sur ce tableau, nous avons présenté l'erreur quadratique moyenne avec le rayon 'v' du sous espace discret et le nombre d'itérations 'It' nécessaire pour l'obtention de tels résultats. Les résultats de DMCI sont meilleurs en performances et en temps de calcul que ceux de la méthode PMCQ

Nous remarquons que la méthode DMCI a permis d'améliorer les performances de la méthode DMC et que l'erreur quadratique moyenne du filtre conçu est identique à celle de méthode RA dans toutes les 3 représentations. Le temps de calcul correspondant est de 80000 fois plus petit en Vfx et Vfl et 24 fois plus petit en Sdpd. Si nous comparons la qualité de sortie (Ems | tps) pour les deux méthodes RA et DMCI nous aurons ce qui suit :

- Pour RA nous avons (0.0319|26h), (0.0320|18h) et (0.127|12s) respectivement en Vfx, Vfl et Sdpd. Nous constatons l'algorithme dans la Sdpd présente la convergence la plus rapide à précision moindre, tandis qu'en Vfx, il a la convergence la plus lente pour la plus grande précision.
- Pour DMCI nous avons (0.0319|1.1 s), (0.0320|0.8 s) et (0.127|0.5 s) respectivement en Vfx, Vfl et Sdpd. Nous constatons l'algorithme dans la Vfx présente la plus faible erreur Ems à un temps acceptable, les résultats de RA sont retrouvés après un temps de calcul très petit. Nous recommandons l'utilisation de cette représentation Vfx pour de bons résultats.

Le tableau 22 présente les coefficients des filtres conçus par les deux méthodes DMCI et PMCQ dans les trois représentations binaires, d'où nous avons schématisé l'allure d'amplitude des filtres correspondants dans les figures suivantes :

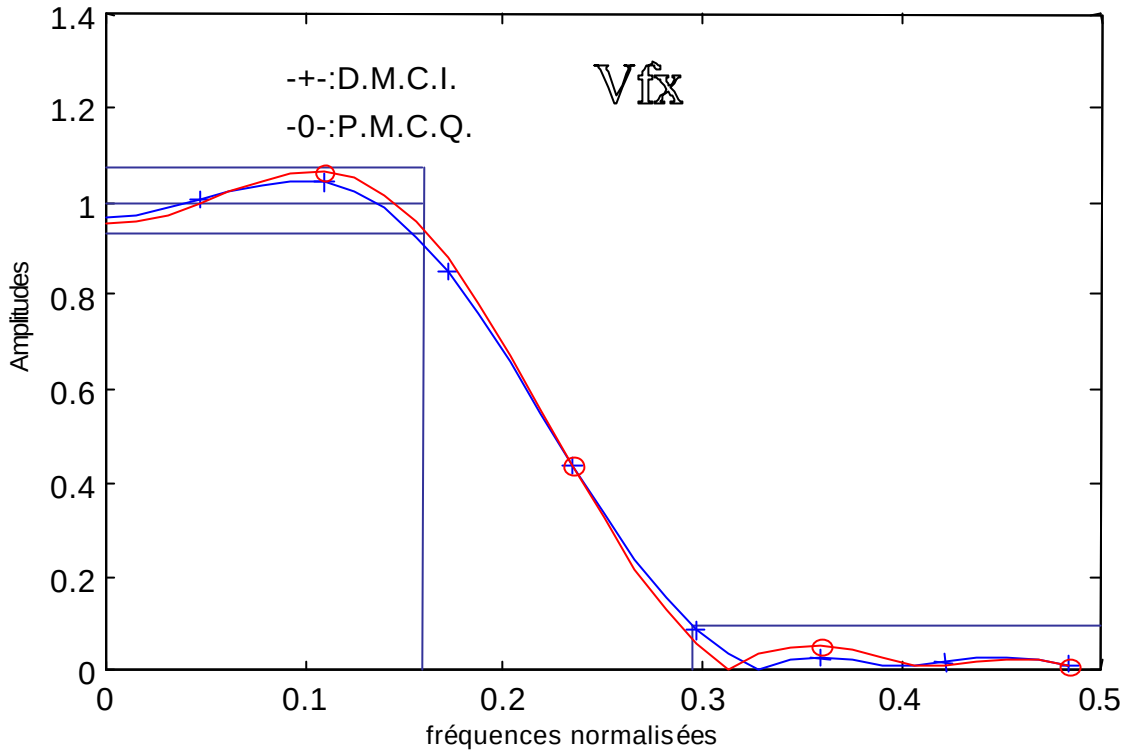


Fig.39. DMCI et PMCQ en représentation Vfx sur lm = 8 bits

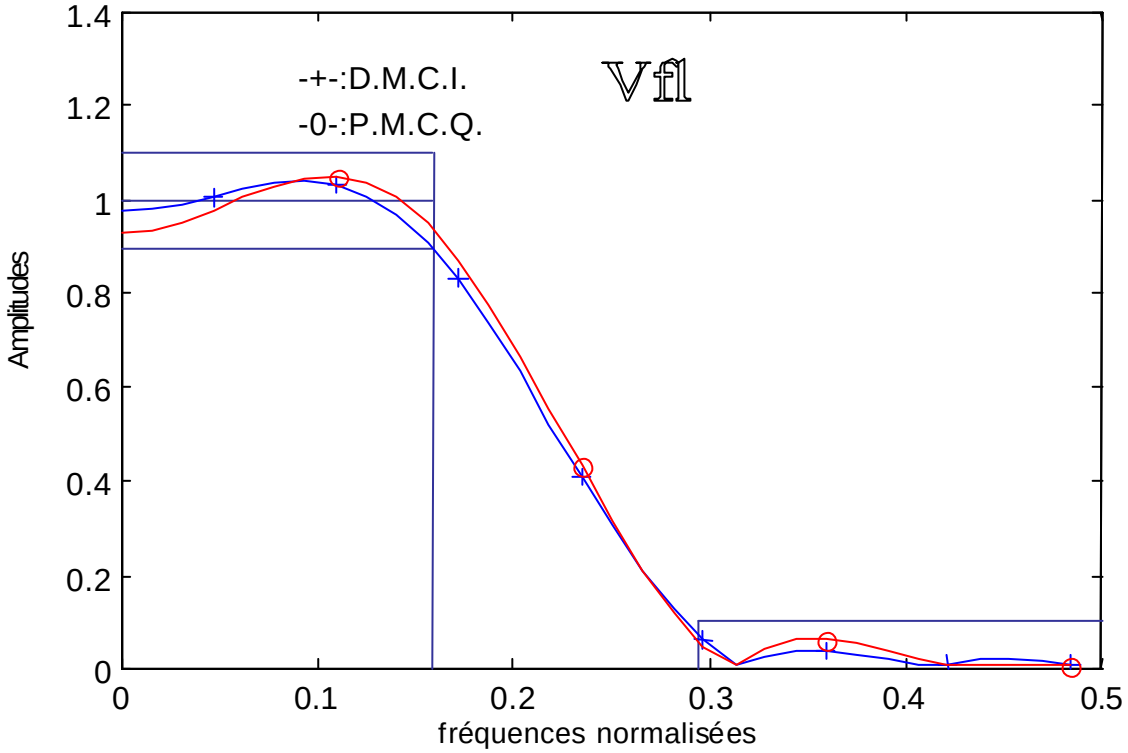


Fig.40. DMCI et PMCQ en représentation Vfl sur lm = 8 bits

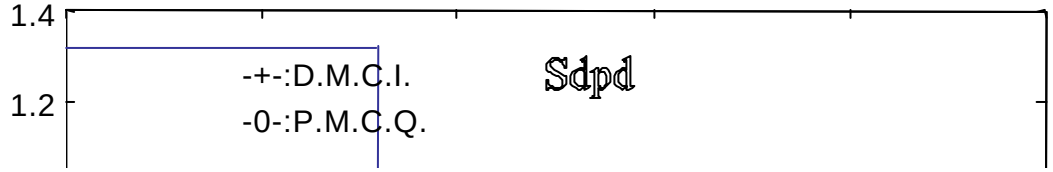


Fig.41. DMCI et PMCQ en représentation Sdpd sur lm = 8 bits

Suivant ces trois figures, nous remarquons que les filtres en Sdpd de DMCI et PMCQ sont superposés et présentent la plus grande valeur des erreurs Ema et Emp, tandis que les filtres en Vfx et Vfl soient équivalents.

IV.1. FILTRE 2 :

N=8.
lm = 16 bits.
wp = 0.159.
ws = 0.295.

La synthèse du filtre choisi, nous a donné les résultats qui sont groupés dans les tableaux suivants :

	D.M.C.I.						D.M.C.		P.M.C.Q.		R.A.	
	Ems	Emp	Ema	Tps	V	It	Ems	Tps	Ems	Tps	Ems	Tps
Vfx	0.0314	0.0912	0.0763	38 s.	40	5	0.0331	5.70 s.	0.0374	12.09s.	--------	-------
Vfl	0.0316	0.0846	0.0816	27 s.	40	5	0.0331	3.85 s.	0.0374	7.52 s.	--------	-------
Sdpd	0.0447	0.0711	0.1498	5 s.	10	5	0.0452	0.66 s.	0.0624	2.09 s.	--------	-------

Tableau 23 : Représentation de l'erreur quadratique moyenne par les quatre méthodes DMCI, RA, DMC, et PMCQ dans les différentes représentations à lm=16 bits

Coef. du filt.	Vfx	Vfl	Sdpd

de long. :8	D.M.C.I.	P.M.C.Q.	D.M.C.I.	P.M.C.Q.	D.M.C.I.	P.M.C.Q.
H(0) =h(7)	-0.05520629	-0.06198120	-0.05467224	-0.0619964	-0.0468750	-0.0620117
H(1)=h(6)	-0.03820800	-0.04998774	-0.04130554	-0.0499877	-0.0546875	-0.0468750
H(2) =h(5)	0.16778564	0.17230229	0.16680908	0.1723022	0.1562500	0.1875000
H(3)=h(4)	0.41006469	0.41091918	0.41162109	0.4108886	0.4375000	0.4375000

Tableau 24 : Tableau de coefficients du filtre retrouvé par DMCI à chaque représentation sur lm=16 bits.

Le tableau 23 présente le temps de calcul et les erreurs Ems des filtres conçus avec les méthodes DMC, DMCI et PMCQ dans les trois représentations binaires.

La méthode R.A. n'est pas réalisable pour les représentations Vfx, Vfl, et Sdpd sur lm=16 bits à cause du temps de calcul prohibitif. C'est pourquoi, les cases qui y correspondent sont en pointillé.A partir du tableau 23, nous remarquons que DMCI a permis d'améliorer les performances de DMC dans un temps 7 fois plus grand. La qualité de sortie (Ems|tps) en représentation Sdpd est (0.0447|246h). Comparés aux résultats du filtre 1 où la configuration est sur 8 bits, nous constatons que seulement en représentation Sdpd il y a une nette amélioration en performances, tandis que dans les représentations Vfx et Vfl, cette amélioration n'est pas très grande vu le grand changement du temps de calcul. Nous ne pouvons pas confirmer que les meilleurs résultats soient obtenus, mais nous pouvons affirmer que la méthode D.M.C.I. dans une représentation binaire Vfx à permis de concevoir le meilleur filtre dans le sens de l'erreur quadratique moyenne.

Les coefficients du tableau 24 nous mènent à avoir les figures suivantes :

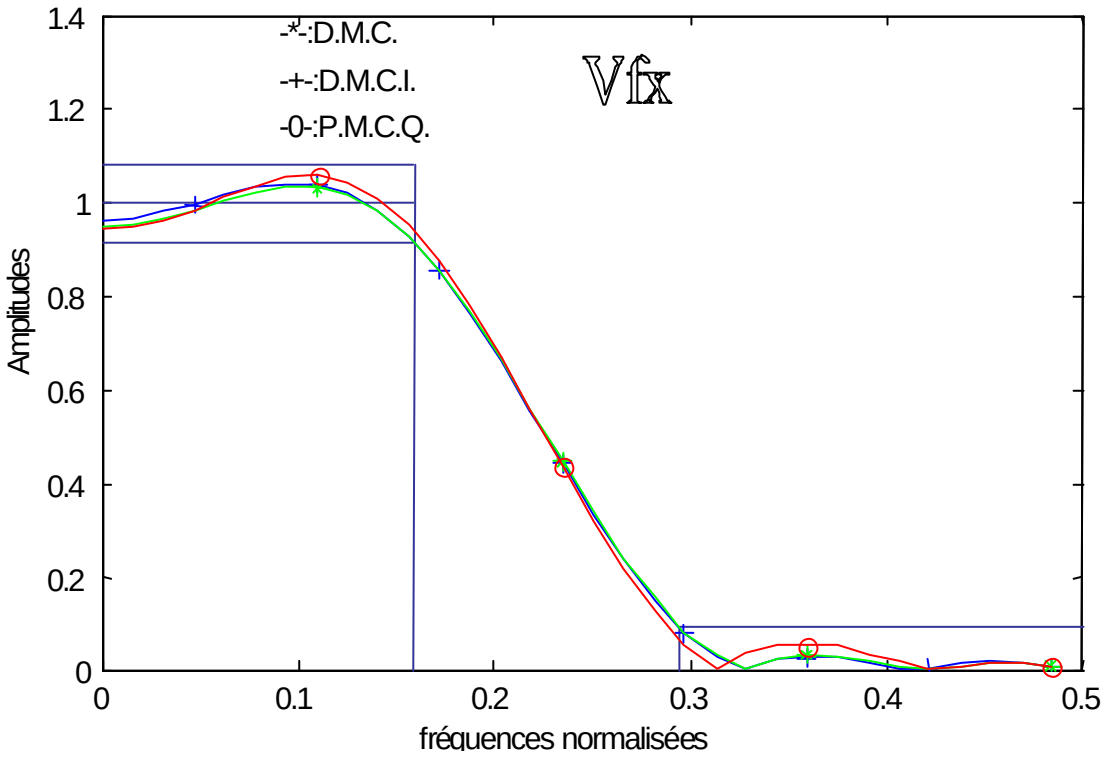


Fig.42. DMC, DMCI et PMCQ en représentation Vfx sur lm = 16 bits

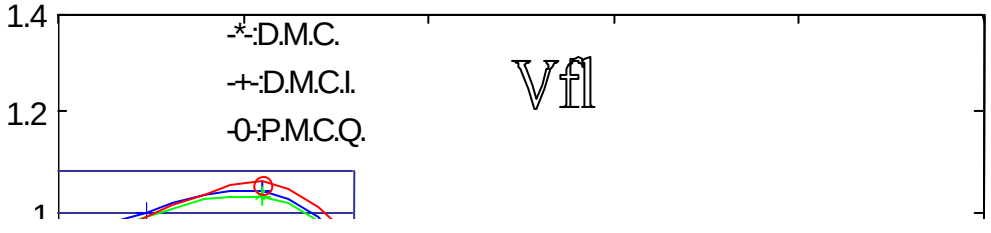


Fig.43. DMC, DMCI et PMCQ en représentation Vfl sur lm = 16 bits

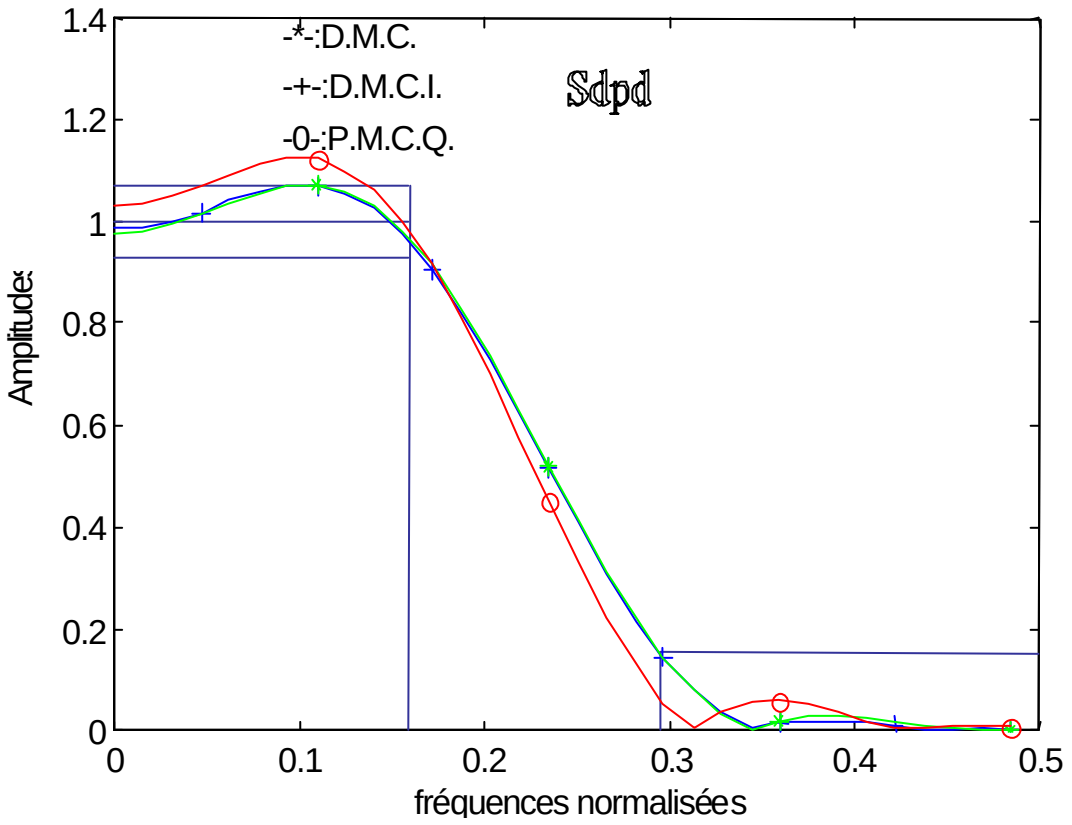


Fig.44. DMC, DMCI et PMCQ en représentation Sdpd sur lm = 16 bits

Suivant ces trois figures, nous remarquons que la réponse en amplitude du filtre de PMCQ en Sdpd sorte du gabarit défini par celle du filtre de DMCI. Ces deux filtres présentent les plus grandes valeurs des erreurs Emp et Ema.

Afin de valider les résultats retrouvés dans ces deux exemples précédents, nous allons effectuer deux autres exemples où nous testerons la méthode sur des filtres de longueur plus grande et dans un espace discret de longueur de mot plus grande.

IV.1. FILTRE 3 :

N=16.
lm = 8 bits.
wp = 0.1.
ws = 0.25.

Pour des raisons de complexité (la méthode RA ne peut pas concevoir des filtres d'ordre N>8) nous nous contenterons de comparer les résultats seulement du DMCI, DMC et PMCQ.

La synthèse du filtre choisi, nous a donné les résultats qui sont groupés dans les tableaux suivants :

	D.M.C.I.						D.M.C.		P.M.C.Q.	
	Ems	Emp	Ema	Tps	V	It	Ems	Tps	Ems	Tps
Vfx	0.0160	0.0168	0.0552	1.2 s.	10	4	0.0293	0.05 s.	0.0120	0.05 s.
Vfl	0.0154	0.0355	0.0607	1 s.	10	4	0.0186	0.05 s.	0.0189	0.05 s.
Sdpd	0.1285	0.1397	0.3535	0.5 s.	5	1	0.1285	0.05 s.	0.1428	0.05 s.

Tableau25 : Représentation de l'erreur quadratique moyenne par les trois méthodes DMCI, DMC et PMCQ dans les différentes représentations sur lm =8 bits.

Coef. de filt.	Vfx		Vfl		Sdpd	
de long. : 16	D.M.C.I.	P.M.C.Q.	D.M.C.I.	P.M.C.Q.	D.M.C.I.	P.M.C.Q.
H(0) =h(15)	0	0.0078125	-0.00195312500	0.0087890625	0	0
H(1)=h(14)	0.0078125	0.0156250	0.00732421875	0.0126953125	0	0
H(2) =h(13)	0.0078125	-0.0078125	0.00439453125	-0.0068359375	0	0
H(3)=h(12)	-0.0234375	-0.0390625	-0.02734375000	-0.0429687500	0	0
H(4)=h(11)	-0.0546875	-0.0468750	-0.04687500000	-0.0429687500	0	0
H(5)=h(10)	0.0234375	0.0468750	0.02539062500	0.0429687500	0	0
H(6)=h(9)	0.1953125	0.2031250	0.18750000000	0.2031250000	0.125	0.250
H(7)=h(8)	0.3515625	0.3281250	0.34375000000	0.3437500000	0.375	0.375

Tableau 26 : Tableau des coefficients du filtre obtenu par DMCI et PMCQ à chaque représentation sur lm=8 bits.

Le tableau 25 présente le temps de calcul et les erreurs Ems des filtres conçus avec les méthodes DMC, DMCI et PMCQ dans les trois représentations binaires.

Les mêmes remarques faites aux exemples 1 et 2 sont vérifiées pour ce cas de filtre. Les filtres de DMCI sont de meilleures performances par rapport à DMC et PMCQ, sauf en représentation Vfx où Ems en DMCI est plus grande que celle en PMCQ. Cet exemple est un cas parmi d'autres où le pas uniforme de la représentation en virgule fixe peut être néfaste lors de la synthèse dans l'espace discret. Par conséquent, on est attrapé par un optimum local dont on ne peut s'en sortir. Comme remarqué dans le tableau 26, cela est du au coefficient h(7) qui est différent. Le choix de ce coefficient est déterminant dans le calcul des autres coefficients. Parmi les filtres conçus, celui de PMCQ en représentation Vfx présente la plus faible Ems à un temps de calcul très petit.Les coefficients du tableau 26 nous mènent à avoir la figure suivante :

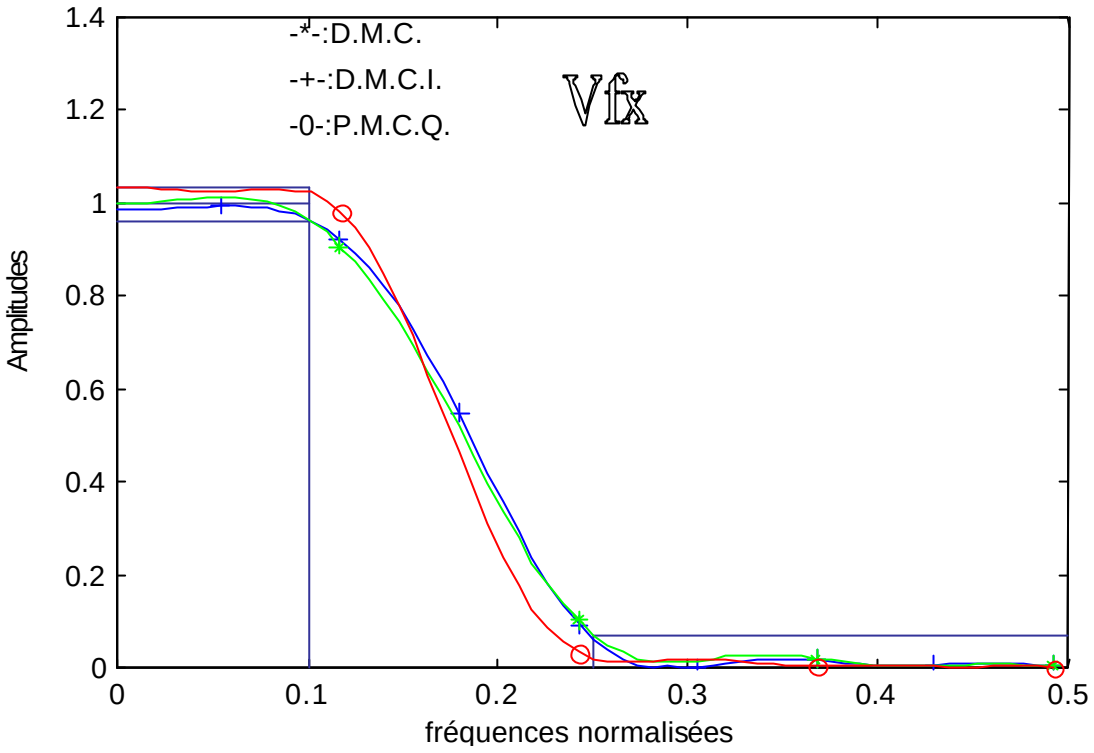


Fig.45. DMC, DMCI et PMCQ en représentation Vfx sur lm = 8 bits

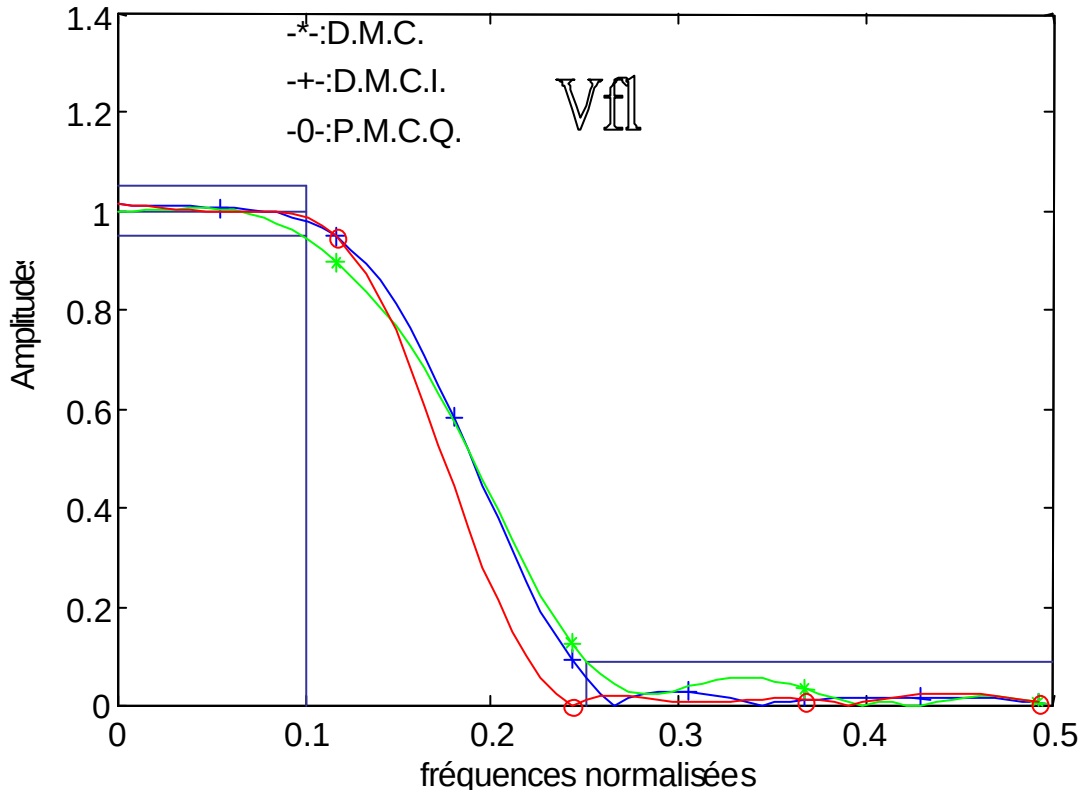


Fig.46. DMC, DMCI et PMCQ en représentation Vfl sur lm = 8 bits

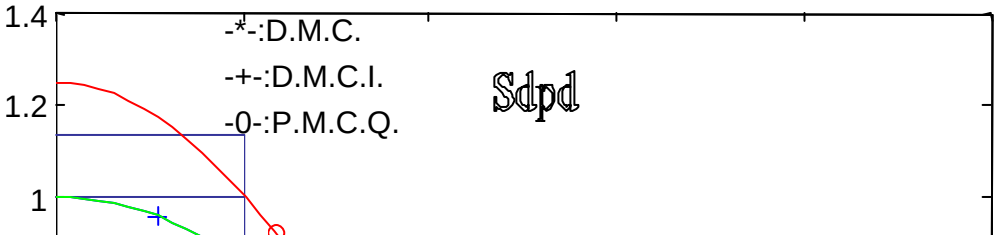


Fig.47. DMC, DMCI et PMCQ en représentation Sdpd sur lm = 8 bits

Comme dans les figures des exemples précédents, nous remarquons que le filtre de PMCQ en Sdpd sort du gabarit prescrit par celui de DMCI, et que ces deux filtres présentent les erreurs Emp et Ema les plus grandes. Tandis que les filtres de DMCI et PMCQ en vfx possèdent une erreur Ema et Emp la plus petite.

IV.4. FILTRE 4 :

N=56.
lm = 16 bits.
wp = 0.318.
ws = 0.371.

Dans cet exemple nous allons aussi présenter les résultats et les coefficients des filtres de Parks –Mc Clellan avec des coefficients à précision infinie que nous notons par PMC.

Nous avons obtenu les résultats qui sont groupés dans les tableaux suivants :

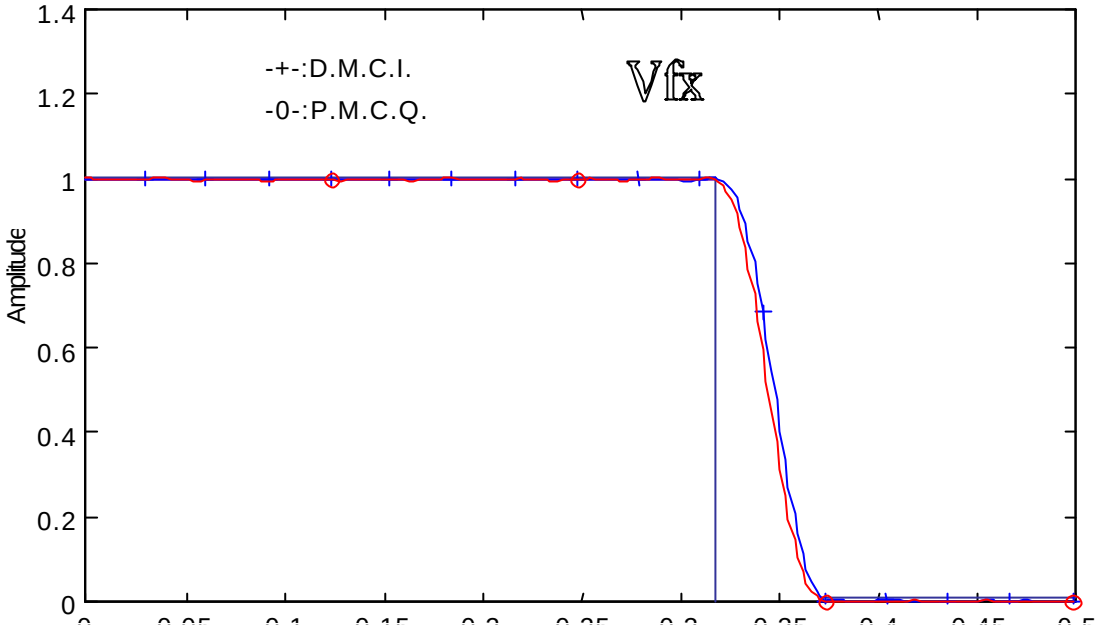
	D.M.C.I.						D.M.C.		P.M.C.Q.		P.M.C.
	Ems	Emp	Ema	Tps	v	It	Ems	Tps	Ems	Tps	Ems
Vfx	0.0013	0.0028	0.0083	164 s	40	4	0.0082	2.92 s.	0.0015	7.41 s.	0.0015
Vfl	0.0013	0.0034	0.0059	162 s	40	4	0.0083	3.07 s.	0.0015	4.18 s.	0.0015
Sdpd	0.0108	0.0602	0.0489	40 s.	10	4	0.0161	1.32 s.	0.0119	0.94 s.	0.0015

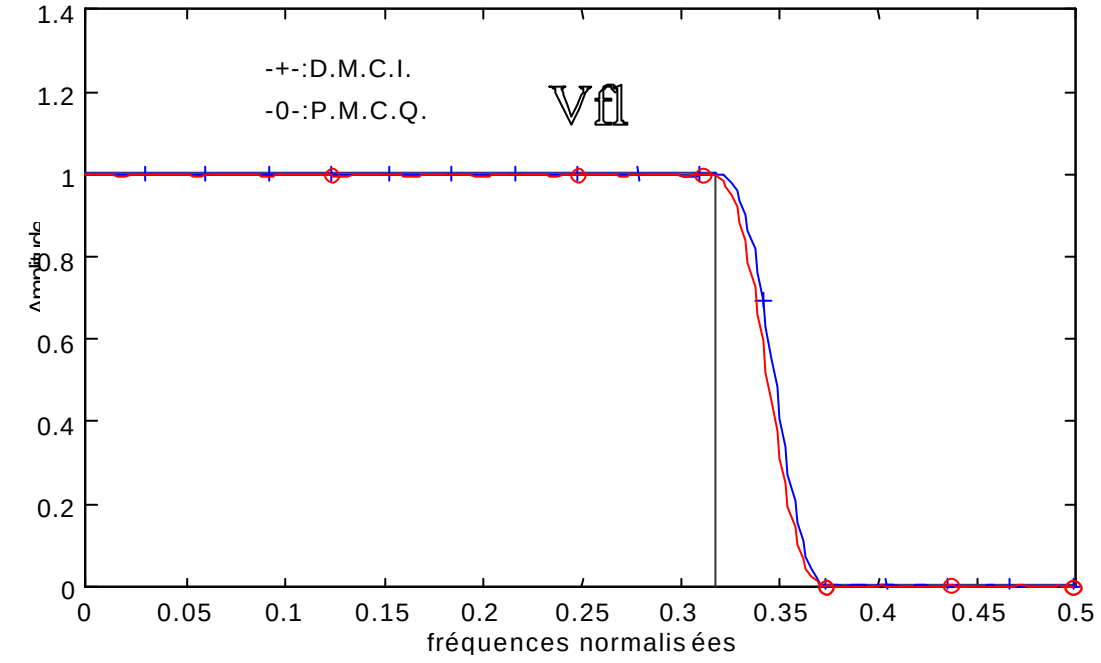
Tableau27 : Représentation de l'erreur quadratique moyenne des méthodes DMCI, DMC, PMCQ et PMC) dans les différentes représentations à lm =16 bits.

Dans cet exemple nous avons testé les capacités de la méthode DMCI pour des filtres de longueur supérieure (N=56) et dans un espace discret à mot machine relativement grand (16 bits). Nous n'avons pas représenté les coefficients des filtres conçus à cause de leur nombre important (56 coefficients). Le tableau 27 présente le temps de calcul et les erreurs Ems des filtres conçus avec les méthodes DMC, DMCI, PMC et PMCQ dans les trois représentations binaires. Les mêmes remarques faites aux exemples 1, 2 et 3 sont vérifiées pour ce cas de filtre. Les filtres de DMCI sont de meilleures performances par rapport à DMC, PMCQ et même PMC, dans les 3 représentations. Les coefficients de PMC sur précision infinie sont entachés des erreurs due à la représentation et les approximations numériques (addition, multiplication…etc), par conséquent les coefficients de DMCI obtenus dans l'espace discret sont de meilleures approximation dans le sens de Ems. L'intérêt de cet exemple n'est

pas d'assurer que la méthode D.M.C.I. est optimale, mais au moins meilleur que P.M.C. sans quantification.

A partir de ces résultats, nous avons obtenus les filtres représentés dans les figures suivantes :





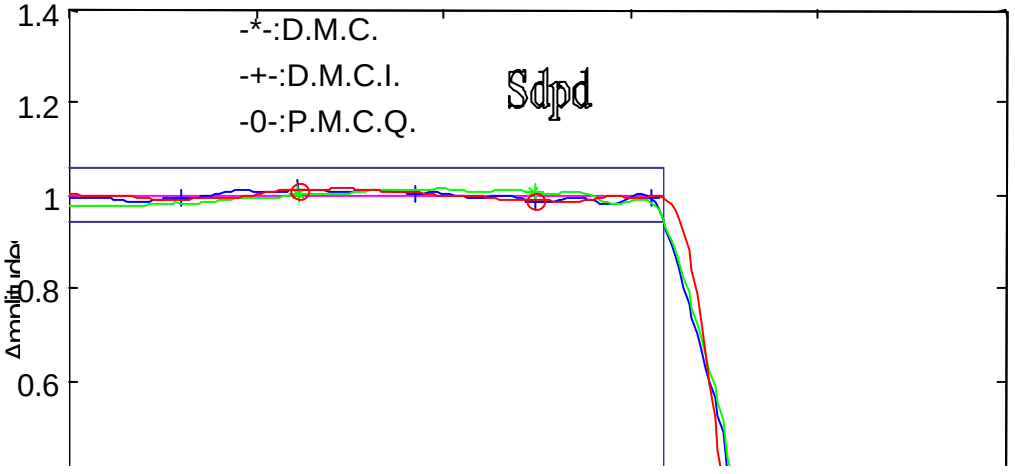


Fig.50. DMC, DMCI et PMCQ en représentation Sdpd sur lm = 16 bits

Dans les figures 48 et 49, nous n'avons pas représenté le filtre de DMC pour éviter l'encombrement des schémas, puis la distinction entre les figures s'avère difficile. Nous remarquons que les filtres conçus en Vfx et Vfl présentent la plus faible erreur minmax dans les bandes BP et BA.

Dans le paragraphe suivant, nous allons effectuer une étude comparative des résultats obtenus dans ces quatre exemples en fonction de l'utilisation de la méthode et de la représentation.

V. COMPARAISON DE LA METHODE DMCI AVEC RA, DMC ET PMCQ :

D'une manière générale, en observant les résultats des quatre exemples considérés, nous remarquons que la méthode DMCI délivre de meilleures performances dans un temps de calcul acceptable. Pour le premier exemple, nous constatons que les résultats de DMCI sont identiques à ceux de RA à un temps de calcul beaucoup plus petit. Pour les autres exemples où la conception des filtres numériques par la méthode RA est prohibitive à cause du grand nombre d'opérations nécessaires, les résultats de la méthode DMCI sont meilleurs que ceux de DMC et PMCQ.

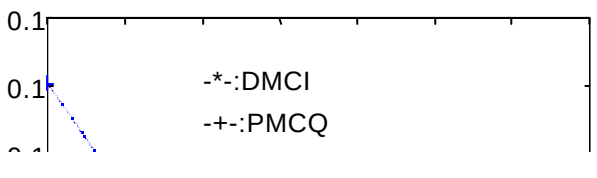
Le temps de calcul de la méthode DMCI est plus grand que celui de DMC et PMCQ, mais demeure très acceptable.

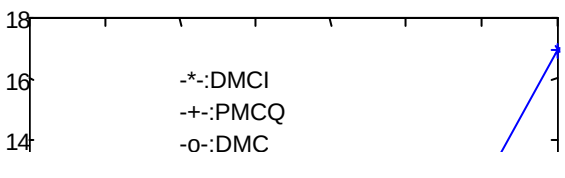
Nous constatons que dans tous les cas, la méthode DMCI a permis d'améliorer les performances de la méthode DMC, pour donner des résultats meilleurs que ceux de PMCQ. Soit 'T' le rapport qui s'exprime de la manière suivante :

$$T(\text{rapport de calcul}) = \frac{\text{temps de calcul de D.M.C.I.}}{\text{temps de calcul de P.M.C.Q.}} \qquad (64)$$

D'une manière générale, le rapport de calcul n'a pas une valeur constante, mais il dépend du nombre d'itérations adéquat 'It' et du rayon du sous espace discret 'v' nécessaire à une bonne optimisation de filtres numériques. En concerne les exemples précédents, ce rapport varie entre 2 et 100.

Afin de mieux interpréter ces résultats, nous avons effectué une étude statistique sur plusieurs centaines de filtres de longueurs différentes et dans des espaces discrets de longueurs de mot.

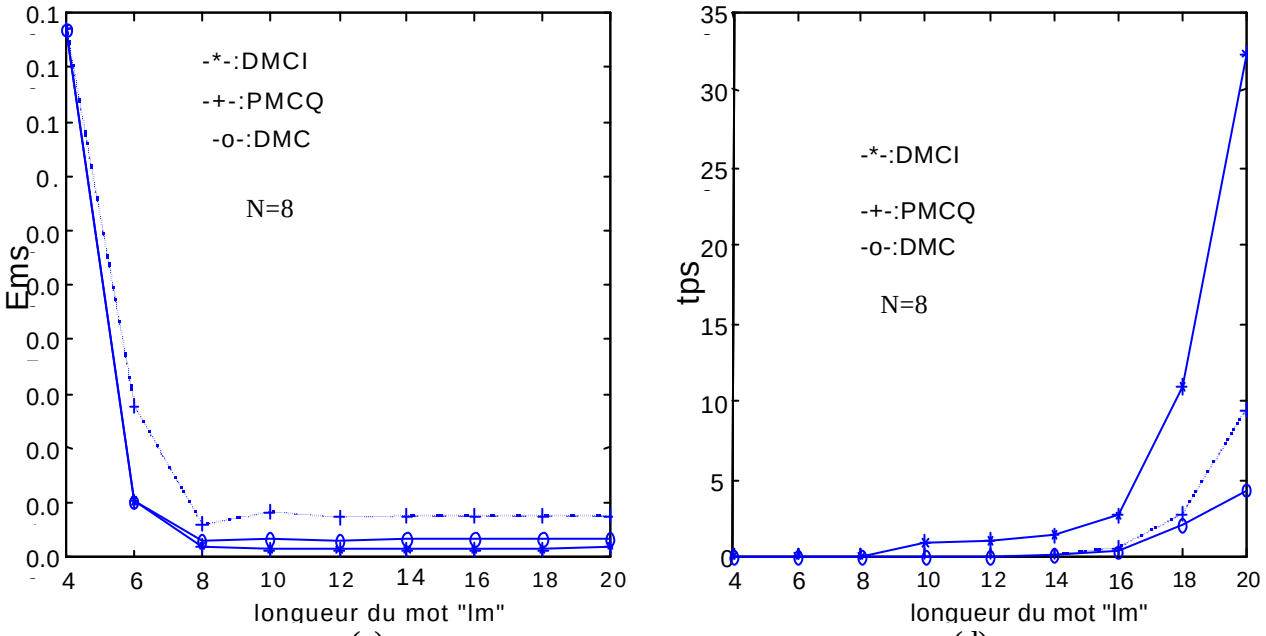




Lm=8bits

Lm=8bits

(a) (b)



(c) (d)

Fig.51. Représentation de Ems et Tps des méthodes DMCI DMC et PMCQ en Vfx en fonction de lm et N.

Les figures 51(a, b, c, et d) présentent les résultats de l'erreur Ems et le temps de calcul tps des méthodes DMC, DMCI, et PMCQ sur plusieurs filtres représentés avec la

représentation Vfx, en fonction de la longueur du filtre et de la longueur du mot lm. Dans tous les cas, la méthode DMCI a été effectuée avec

v = entier (va/10) où va est le nombre de valeurs admissibles sur lm

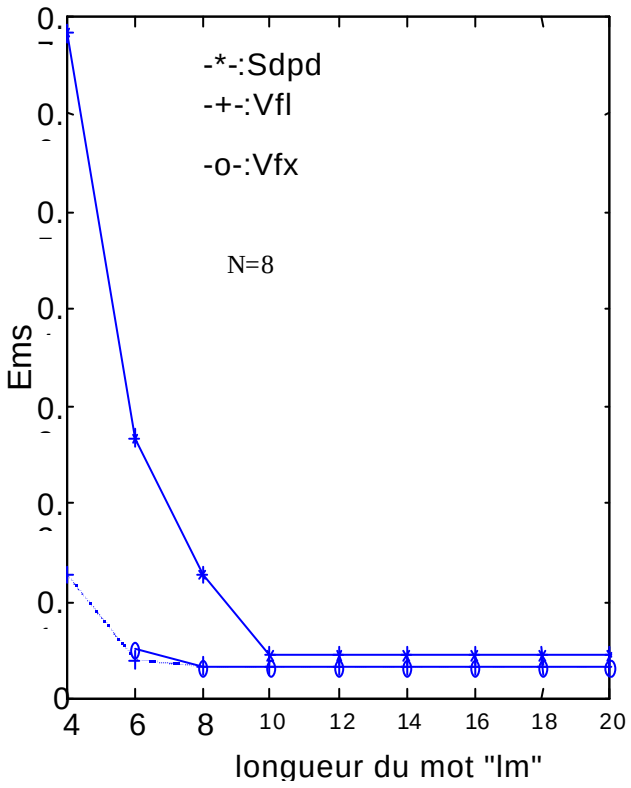
et It= 4.

Nous remarquons de ces figures que la méthode DMCI a procuré les meilleurs performances à un temps de calcul plus grand que celui de DMC et PMCQ mais demeure acceptable. Le temps de calcul augmente plus avec lm qu'avec N. tandis que Ems de DMCI diminue quand N et lm augmentent.

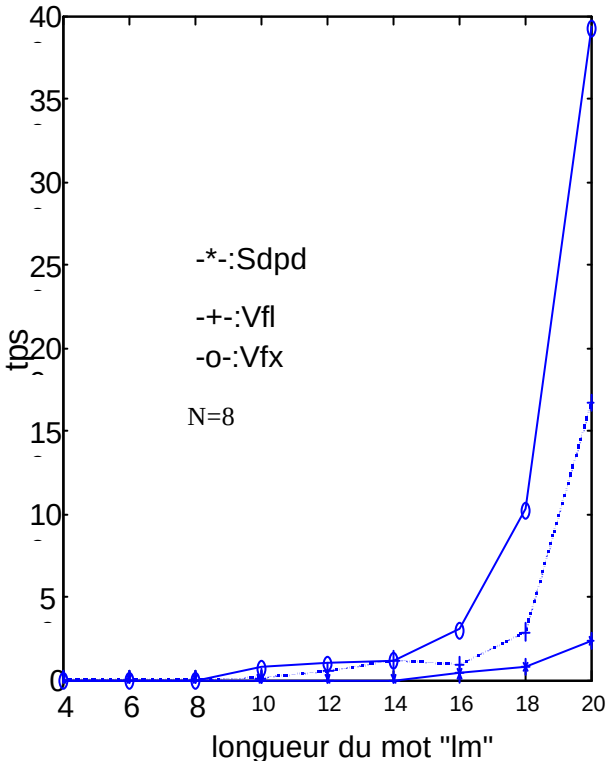
Afin de voir l'influence de la représentation sur l'utilisation de la méthode DMCI, une étude statistique a été menée sur plusieurs filtres avec chaque représentation dans le paragraphe suivant.

VI. ETUDE DU CHOIX DE LA REPRESENTATION EN UTILISANT LA METHODE DMCI :

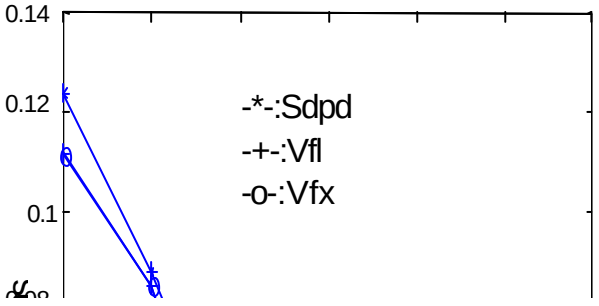
Pour une étude comparative du choix de la représentation binaire pour l'utilisation de la méthode DMCI, une étude statistique a été menée sur plusieurs centaines de filtres. Les résultats correspondants ont été relevés dans les quatre figures suivantes. Nous avons représenté les résultats de l'erreur Ems et du temps de calcul tps de l'algorithme de DMCI dans les trois représentations Vfx, Vfl et Sdpd en fonction de N et lm.

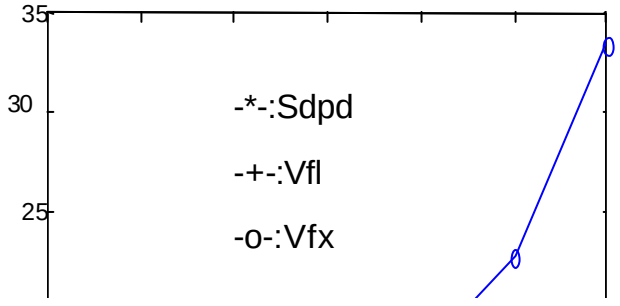


(a)



(b)





Lm=8bits Lm=8bits

(c) (d)

Fig.52. Représentation de Ems et Tps de la méthode DMCI dans les représentations Vfx, Vfl et Sdpd en fonction de lm et N.

De ces figures (52), nous remarquons que la méthode DMCI dans la représentation Vfx a procuré les meilleurs performances tandis que le temps de calcul est le plus grand. La représentation Sdpd a procuré le temps de calcul le plus petit mais de performances moindres. La représentation Vfl a présenté de bonnes performances à un temps de calcul acceptable se situant entre celui de Vfx et de Sdpd.

Le choix de la représentation dépend des spécifications (Ems|tps) requises. Puisque la méthode DMCI a une convergence algorithmique acceptable, nous recommandons l'utilisation de Soit Vfx ou Vfl.

VII. CONCLUSION :

Dans ce chapitre, nous avons proposé l'algorithme nommé DMCI qui utilise une nouvelle approche qui consiste à retrouver les performances de RA et la rapidité de DMC. A travers quelques exemples nous avons montré que les objectifs recherchés ont été atteints. Dans l'exemple 1, nous avons constaté que la conception d'un filtre à 8 coefficients par la méthode DMCI a donné les mêmes performances du filtre de la méthode RA (Ems=0.0319 en Vfx) dans un temps de calcul 80000 fois plus petit. L'étude comparative basée sur les critères Ems|tps que nous avons donné dans ce chapitre montre que les résultats obtenus peuvent être considérés comme très interressants. Dans les exemples, il a été constaté que les représentations à virgule fixe et à virgule flottante sont mieux adaptées à DMCI La méthode DMCI est plus lente que la méthode DMC avec un temps de synthèse très acceptable mais nettement plus rapide que la méthode RA, prohibitive pour des filtres de longueur élevée.

Notre recherche actuelle qui entre dans le contexte du chapitre IV, consiste à améliorer la complexité de cet algorithme RA et cela en faisant une réduction de l'espace discret en se basant sur des propriétés mathématiques du modèle numérique.

Chapitre IV.
METHODE DE RECHERCHE SEQUENTIELLE ET PROGRESSIVE 'R.S.P.'

I. INTRODUCTION :

Dans le chapitre précédent, nous avons élaboré une méthode itérative qui améliore les résultats de DMC tout en gardant la faible complexité algorithmique. Il a été montré que les performances de la méthode RA, pouvant être atteintes, et ne sont pas garanties. Par conséquent, notre propos dans ce chapitre est de réduire l'espace discret de recherche des coefficients du filtre à concevoir. Nous allons élaborer une nouvelle méthode qui préserve les performances de la méthode RA et améliore sa complexité algorithmique. Contrairement au chapitre précédent qui utilise l'espace global dans la conception des filtres en utilisant la méthode des moindres carrés, cette méthode nommée méthode de Recherche Séquentielle et Progressive 'R.S.P.' est basée sur une recherche exhaustive, dans un espace dont le nombre de valeurs est réduit, susceptible de contenir la solution de meilleure approximation dans le sens du critère choisi (Ems ou Emm). Dans le travail de ce chapitre, nous avons choisi la représentation binaire en virgule fixe à cause de son pas discret uniforme.

II. METHODE DE RECHERCHE SEQUENTIELLE ET PROGRESSIVE 'RSP' :

Le but de cette méthode est de retrouver les résultats de la méthode de la recherche arborescente dans un temps diminué (coût meilleur), ceci en réduisant l'espace de recherche des coefficients. Afin de comparer les résultats obtenus par notre méthode à ceux de Park-McClellan, nous avons été amenés à choisir le même critère d'erreur qui est celui de Tchebyshev (erreur min-max), puis nous avons étendu notre méthode en utilisant le critère de l'erreur quadratique moyenne.

II.1. IDEE GENERALE :

L'algorithme RSP se subdivise en deux étapes :

- Un filtre est obtenu avec un pas grand en utilisant R.A. dans le sens du critère choisi.
- Autour du voisinage du filtre obtenu dans l'étape précédente une autre recherche est effectuée avec un pas réduit par rapport au précédent.

Cette seconde étape est renouvelée tant que nous n'avons pas atteint le pas minimal désiré.

II.2. DESCRIPTION DE LA METHODE :

Soit à concevoir un filtre numérique RIF à phase linéaire de longueur N dont la réponse en fréquence H(f) s'écrit habituellement sous la forme (29){chap.1}. Dans le chapitre I, Il a été montré que l'amplitude de la réponse en fréquence des quatre cas de filtres à phase linéaire peut être écrite sous la forme (32).
Nous posons $a_k = \alpha(k)$: la réponse implusionnelle dépendante du cas considéré. Nous aurons

$$Pn(f) = \sum_{k=0}^{n-1} a_k . \cos 2\pi fk \qquad (65)$$

Pn(f) étant la combinaison de fonctions cosinusoïdales qui dépende de chaque cas et 'n' est nombre de termes

$$n = \begin{cases} \frac{N}{2} \text{ pour N paire} \\ \frac{N-1}{2} \text{ pour N impaire et réponse impulsionn elle symétrique} \\ \frac{N+1}{2} \text{ pour N impaire et réponse impulsionn elle antisymétr ique} \end{cases}$$

et a_k est relative à h_k est la séquence décalée résultante dépendante du cas considéré.

La fonction Pn (f) est comparée avec l'amplitude de la réponse de la fréquence désirée 'D (f)' au sens de l'erreur minmax, comme il est fait dans le cas de la conception de filtre RIF à phase linéaire à précision infinie (Eq. 44 dans le chapitre II). L'erreur pondérée d'approximation 'Emm' est donnée par

$$Emm = \min_{(coeff.a)} \max_{f \in F} W(f) \left| D(f) - P_n(f) \right| \qquad (66)$$

avec

- F: l'union disjointe des bandes de fréquence d'intérêt.
- W (f): la fonction de pondération définie sur F.
- D (f): l'amplitude de la réponse en fréquence désirée ou idéale.

Utilisant l'Eq. (65) dans l'Eq. (66) donne

$$Emm = \min_{(coeff.a)} \max_{f \in F} W(f) \left| D(f) - \sum_{k=0}^{n-1} a_k \cos 2p\ fk \right| \qquad (67)$$

Les coefficients du filtre sont restreints à des valeurs discrètes représentables dans un mot machine de lm bits binaires.
Dans ce qui suit la représentation binaire choisie est la virgule fixe. Nous pourrions montrer qu'une extension à d'autres représentations binaires telles que la virgule flottante ou la Sdpd peut être faite très facilement.

Considérons le problème de la conception de filtre avec une contrainte imposante une limite sur la longueur du mot du coefficient 'a_k', k= 0,1., N/ 2.
En utilisant la représentation en virgule fixe, nous pouvons exprimer le coefficient discret 'a_k' comme une combinaison linéaire:

$$| a_k | = \sum_{j=1}^{lm-1} y_{j,k}\, 2^{-j} \qquad k= 0, 1., N/ 2. \tag{68}$$

où 'lm' est la longueur du mot binaire permise pour la conception de filtre numérique et 'j' est l'indice du bit binaire. '$y_{j,k}$' est une variable bivalente. Elle peut prendre les valeurs '0' ou '1'.
D'ici, l'amplitude de la réponse en fréquence Pn (f) peut être exprimée sous la forme

$$Pn\,(f) = \sum_{k=0}^{n-1} s.\Big(\sum_{j=1}^{lm-1} y_{j,k}.2^{-j}\Big).\cos 2\pi fk. \tag{69}$$

où s est le signe de 'a_k', s (= - 1 ou +1).

Nous considérons que dans un mot processeur de longueur 'lm' bits, nous ne pouvons pas utiliser la méthode 'R.A.' pour la synthèse de filtre, à cause du temps de calcul prohibitif dû au grand nombre de valeurs admissibles. Par ailleurs, nous pouvons calculer le filtre numérique à coefficients discrets dans une longueur du mot plus petite, 'lmp' bits binaires avec (lmp ≤ lm) , pour un faible nombre de valeurs admissibles, avec la méthode R.A. 'a_k' peut être exprimé comme suit :

$$|a_{k(opt)}| = \sum_{j=1}^{lmp-1} y_{j,k(opt)} 2^{-j} \qquad k= 0, 1,...., N/2. \tag{70}$$

$a_{k(opt)}$: le coefficient relatif au filtre numérique optimal au sens de l'erreur minmax, calculé sur une longueur de mot 'lmp'.
$y_{j,k(opt)}$: valeur binaire du bit 'j' pour le coefficient '$a_{k(opt)}$'.

Cette solution '$a_{k(opt)}$' est prise comme point de départ (solution de départ) par la méthode que nous présenterons nommé méthode de Recherche Séquentielle et Progressive 'RSP'. Cette méthode se branche à partir cette solution pour la recherche des 'N' coefficients de filtre représentés sur la longueur de mot 'lm'. Le critère d'erreur utilisé est celui de Chebyshev 'Minmax'.

Au début, les coefficients '$a_{k(opt)}$' ont été calculés sur 'lmp' bits (lmp<lm) en utilisant la méthode RA. Puis, nous concevons ce filtre à coefficients discrets sur une longueur de 'lmp+1' bits binaires après une réduction de l'espace de recherche relatif à lmp+1 bits. Cette réduction d'espace s'effectue autour des solutions retrouvés précédemment. Graduellement, nous augmentons la longueur du mot (lmp+2, lmp+3, lmp+4, …….) en réduisant à chaque fois l'espace discret de recherche jusqu'à atteindre la longueur de mot sollicitée 'lm'. Pour chaque étape 'i,' nous définissons la fonction à plus faible borne Emm_i, qui peut être écrite comme suit :

$$Emm_i(a) = \min_{(coeff.a)} \max_{f\in F} W(f)\left|D(f) - \sum_{k=0}^{n-1} a_k \cos 2pos\right| \tag{71.a}$$

pour $a = \{a_k : k= 0, 1,...., N/2\}$,

en d'autre terme nous avons :

$$Emm_i(a) = \min\ z \tag{71.b}$$

$$\text{avec} \qquad z = \max_{f \in F} W(f) \left| D(f) - \sum_{k=0}^{n-1} a_k \cos 2pfk \right| \tag{71.c}$$

Pour tout a_k (k= 0, 1,...., N/2), $Emm_i(a)$ est définie comme la meilleure valeur de la fonction z.

$$Emm_i(a) = \min z \tag{72.a}$$

sous les contraintes

$$a_k^{lmp+1} \le a_k^{lmp} + \mu \qquad k=1,..., N/2 \tag{72.b}$$

et

$$a_k^{lmp+1} \ge a_k^{lmp} - \mu \qquad k=1,..., N/2 \tag{72.c}$$

μ : intervalle choisi pouvant contenir la solution.

L'implantation d'une valeur continue sur deux mots de longueurs différentes en représentation Vfx, ne procure pas nécessairement deux valeurs discrètes égales. Le maximum de la différence entre ces valeurs discrètes est l'erreur de quantification '±LSB' (least sided bit) dû soit à la troncature (±LSB) soit à l'arrondissement (±1/ 2.LSB) (voir chapitre I parag. III.2).

Par conséquent, nous avons surestimé le μ à

$$\mu = LSB = 2^{-(lmp-1)}. \tag{73}$$

en remplaçant Eq. (73) et Eq. (70) dans Eq. (72)

$$\sum_{j=1}^{lmp} y_{j,k} 2^{-j} \le \sum_{j=1}^{lmp-1} y_{j,k(opt)} 2^{-j} + 2^{-(lmp-1)} \tag{74.a}$$

$$\sum_{j=1}^{lmp} y_{j,k} 2^{-j} \ge \sum_{j=1}^{lmp-1} y_{j,k(opt)} 2^{-j} - 2^{-(lmp-1)} \tag{74.b}$$

en développant Eq. (74) nous obtenons

$$\sum_{j=1}^{lmp-2} y_{j,k(opt)} 2^{-j} + \sum_{j=lmp-1}^{lmp} y_{j,k} 2^{-j} \le \sum_{j=1}^{lmp-2} y_{j,k(opt)} 2^{-j} + y_{lmp-1,k(opt)} 2^{-(lmp-1)} + 2^{-(lmp-1)}. \tag{75.a}$$

$$\sum_{j=1}^{lmp-2} y_{j,k(opt)} 2^{-j} + \sum_{j=lmp-1}^{lmp} y_{j,k} 2^{-j} \ge \sum_{j=1}^{lmp-2} y_{j,k(opt)} 2^{-j} + y_{lmp-1,k(opt)} 2^{-(lmp-1)} - 2^{-(lmp-1)}. \tag{75.b}$$

après simplification, nous aurons :

$$y_{lmp-1,k}2^{-(lmp-1)} + y_{lmp,k}2^{-(lmp)} \leq y_{lmp-1,k(opt)}2^{-(lmp-1)}+2^{-(lmp-1)}. \quad (76.a)$$

$$y_{lmp-1,k}2^{-(lmp-1)} + y_{lmp,k}2^{-(lmp)} \geq y_{lmp-1,k(opt)}2^{-(lmp-1)}-2^{-(lmp-1)}. \quad (76.b)$$

d'ici, nous obtenons

$$y_{lmp-1,k}+y_{lmp,k}2^{-1} \leq y_{lmp-1,k(opt)}+1 \quad (77.a)$$
$$y_{lmp-1,k}+y_{lmp,k}2^{-1} \geq y_{lmp-1,k(opt)}-1 \quad (77.b)$$

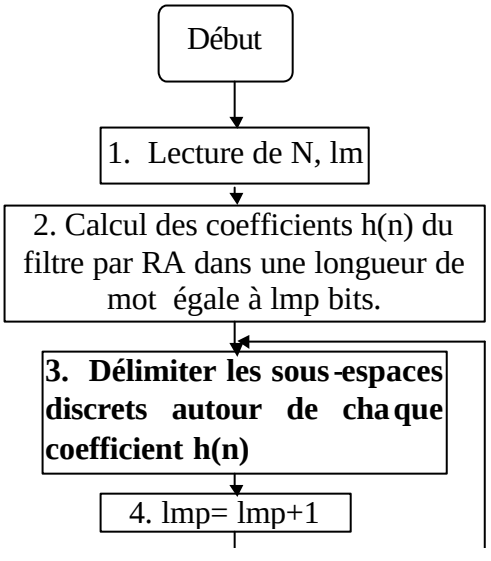
Le problème est restreint à une résolution de deux inéquations avec deux variables (x1, x2) sous la forme de

$$a1x1+a2\ x2 \leq b1 \quad (78.a)$$
$$a1x1+a2\ x2 \geq b2 \quad (78.b)$$

(a1, a2, b1, b2) sont des constantes.
a1= 1,
a2= 2^{-1},
b1= y_p+1,
b2= y_p-1.
Où y_p est la solution précédemment calculée. (dans ce cas $y_p = y_{lmp-1,k(opt)}$)
A partir de ces deux inéquations, nous avons un ensemble de valeurs discrètes admissibles dans laquelle nous choisirons le meilleur arrangement (x1, x2) permettant d'avoir des coefficients qui donnent la meilleure approximation au sens de Chebyshev.
Cette procédure est séquentiellement refaite pour chaque coefficient et à chaque étape de croissance de la longueur de mot, jusqu'à atteindre l'espace discret de longueur de mot désirée. Notons que cette méthode n'affecte pas la variable bivalente de 1 à 'lmp-2' bits obtenu par la méthode R.A.. Par conséquent, nous améliorons la précision des coefficients en ajoutant des bits binaires de 'lmp-1' à 'lm', afin de mieux définir la valeur du coefficient.

II.3. ORGANIGRAMME

L'organigramme de cette méthode R.S.P. peut être schématisé comme suit :



Non

Non

Fig. 59. Organigramme de la méthode R.S.P.

II.4. DESCRIPTION DE L'ORGANIGRAMME :

1. Lecture des spécifications du filtre (N, lm, type de filtre, fréquences de coupure).
2. Calcul des coefficients discrets de longueur de mot réduite lmp < lm par la méthode RA. lmp est choisi conventionnellement lmp ≤ 5 bits, car dépassée cette valeur l'algorithme de la méthode RA est plus lent.
3. Définition de l'intervalle de recherche correspondant à chaque coefficient précédemment calculé dans un espace discret dont la longueur de mot est plus grande de un bit.
4. Ajout à lmp un incrément de 1 bit.
5. Calcul des coefficients offrants la meilleure approximation au sens de Chebyshev en utilisant Eq. (78)
6. Test sur la longueur du mot : si la longueur désirée lm est atteinte, la recherche serait arrêtée, sinon, nous continuons à partir de 3.
7. Impression des résultats.

Les caractéristiques principales utilisées par l'algorithme sont décrites ci-dessous :

- La stratégie du branchement est relative au point de départ.
- La politique de recherche est similaire que dans [8], la technique utilisée est exhaustive dans l'espace de recherche défini, la même que dans la méthode RA [10].

Pour montrer l'efficacité de l'algorithme que nous proposons, nous allons étudier la complexité de RSP en nombre d'opérations par rapport à celle de la méthode RA.

III. NOMBRE D'OPERATIONS :

Puisque le calcul du temps d'exécution des programmes RSP et RA est relatif et dépend des spécifications matérielles et logicielles du calculateur utilisé (la fréquence du travail, le logiciel utilisé, …etc.), nous avons préféré présenter les programmes en nombre d'opérations sous la forme de fonctions d'évaluation 'N_{fe}' définies dans le chapitre II, afin que la comparaison soit plus significative.

Dans ce qui suit, nous allons faire une étude comparative sur le nombre de fonctions d'évaluations par les méthodes RSP et RA dans le cas d'un filtre RIF à phase linéaire. La symétrie

(l'antisymétrie) des coefficients est considérée. Il en découle que dans notre méthode le nombre de coefficients à retrouver constitue la moitié du nombre total.

III.1. NOMBRE D'OPERATIONS DE LA METHODE R.A. :

Le nombre de fonctions d'évaluations 'N_{fe}^{RA}' nécessaire pour le fonctionnement normal de la méthode RA est calculé comme suit :

Si 'va' est le nombre de valeurs admissibles dans un espace discret de longueur de mot 'lm' bits, et 'N' le nombre de coefficients du filtre.

$$N_{fe}^{RA} = (va)^{N/2} \tag{79}$$

La représentation binaire choisie est la représentation à virgule fixe. Le nombre de valeurs discrètes 'va' admissibles (positives et négatives) représentables sur lm bits, incluant le bit de signe, peut être calculé comme suit :

$$va = 2^{lm} - 1 \tag{80}$$

en remplaçant (80) dans (79) nous obtenons :

$$N_{fe}^{RA} = (2^{lm} - 1)^{N/2} \tag{81}$$

III.2. NOMBRE D'OPERATIONS DE LA METHODE RSP :

Le nombre de fonctions d'évaluations 'N_{fe}^{RSP}' nécessaire pour le fonctionnement normal de la méthode RSP est calculé pour le même nombre de coefficients et dans la même longueur binaire comme suit.

$$N_{fe}^{RSP} \leq (va_{lmp})^{N/2} + (va_{lmp+1}(s))^{N/2} + (va_{lmp+2}(s))^{N/2} + \ldots\ldots\ldots + (va_{lm}(s))^{N/2}. \tag{82}$$

Avec lmp : longueur du mot initiale où nous utilisons RA.

va_{lmp} : valeurs discrètes admissibles sur un mot de longueur lmp où

$$va_{lmp} = 2^{lmp} - 1. \tag{83}$$

$va_{lmp+i}(s)$: valeurs discrètes admissibles sur un mot de longueur (lmp + i) dans un sous espace de sélection 's' où i est un entier allant de 1 à 'lm –lmp'.

Puisque le sous espace discret 's' est le même pour des longueurs de mot différentes, alors le nombre de valeurs admissibles est égal.

$$va_{lmp+1}(s) = va_{lmp+2}(s) = \ldots\ldots\ldots = va_{lm}(s) \tag{84}$$

Nous avons choisi l'intervalle de recherche égal au ± LSB, qui constitue un bit. Ce bit peut prendre deux valeurs (0 et 1), elles peuvent être soit positives, soit négatives(4 valeurs) plus la valeur médiane (précédemment calculée) on aura en tout 5 valeurs admissibles.

$$va_{lmp+1}(s) = va_{lmp+2}(s) = \ldots\ldots = va_{lm}(s) = 5 \text{ valeurs admissibles.} \quad (85)$$

En remplaçant (85) et (83) dans (82) nous aurons

$$N_{fe}^{RSP} \leq (2^{lmp}-1)^{N/2} + (lm-lmp).(5)^{N/2} \quad (86)$$

III.3. ETUDE COMPARATIVE DE LA COMPLEXITE ENTRE RSP ET R.A. :

L'étude comparative de la complexité entre les deux méthodes RA et RSP nous mène à une comparaison entre le nombre d'opérations des deux méthodes d'où le nombre de fonctions d'évaluations. A partir de Eq. (81) et Eq. (86) nous avons le nombre de fonctions d'évaluations des deux méthodes avec :

$$N_{fe}^{RA} = (2^{lm}-1)^{N/2} .$$

$$\max(N_{fe}^{RSP}) = (2^{lmp}-1)^{N/2} + (lm-lmp).(5)^{N/2} \quad (87).$$

XX. Afin d'évaluer le gain en temps d'exécution nous calculons le rapport en nombre de fonctions d'évaluations nommé '$T_{RA/RSP}$'.

XXI. $$T_{RA/RSP} = N_{fe}^{RA} / \max(N_{fe}^{RSP}).$$

XXII.

$$T_{RA/RSP} = ((2^{lm}-1)^{N/2}) / ((2^{lmp}-1)^{N/2} + (lm-lmp).(5)^{N/2}) \quad (88).$$

Conventionnellement, nous avons choisi : $lm \geq 8$ bits et $lmp \leq 5$ bits.

De ce principe, nous remarquons que $T_{RA/RSP}$ est au moins de l'ordre des centaines même pour des filtres à faible longueur. Puisque Eq. (88) n'est pas vraiment significative à l'œil nue, nous donnons un exemple numérique dans le tableau suivant, avec quelques espaces discrets de longueurs de mot 'lm' différentes (8 bits et 16 bits) afin de comparer la complexité numérique des deux méthodes en calculant à chaque fois '$T_{RA/RSP}$'. Pour tous les cas, Le filtre choisi est un filtre RIF à phase linéaire de longueur 8.

lm/lmp (bits)	N_{fe}^{RA} (fonctions)	Max(N_{fe}^{RSP}) (fonctions)	$T_{RA/RSP}$
8/3	4228250625	5526	765155
8/4	4228250625	53125	79591
16/3	18445618199572250000	10526	1752386300548380
16/4	18445618199572250000	58125	317343969024899

Tableau 28. Comparaison du nombre de fonctions d'évaluations d'exécution des algorithmes RA et RSP pour un filtre de longueur 8 dans des espaces discrets de longueurs de mot différentes.

A partir de ce tableau, nous remarquons clairement le gain en fonctions d'évaluations en utilisant la méthode RSP au lieu de RA, cela s'est effectué après avoir réduit l'espace discret de recherche, d'où la diminution du nombre de valeurs admissibles à traiter. Nous donnons dans la figure 60 (a et b) l'évolution du rapport $T_{RA/RSP}$ en fonction de N (pour lm fixe) et lm (pour N fixe).

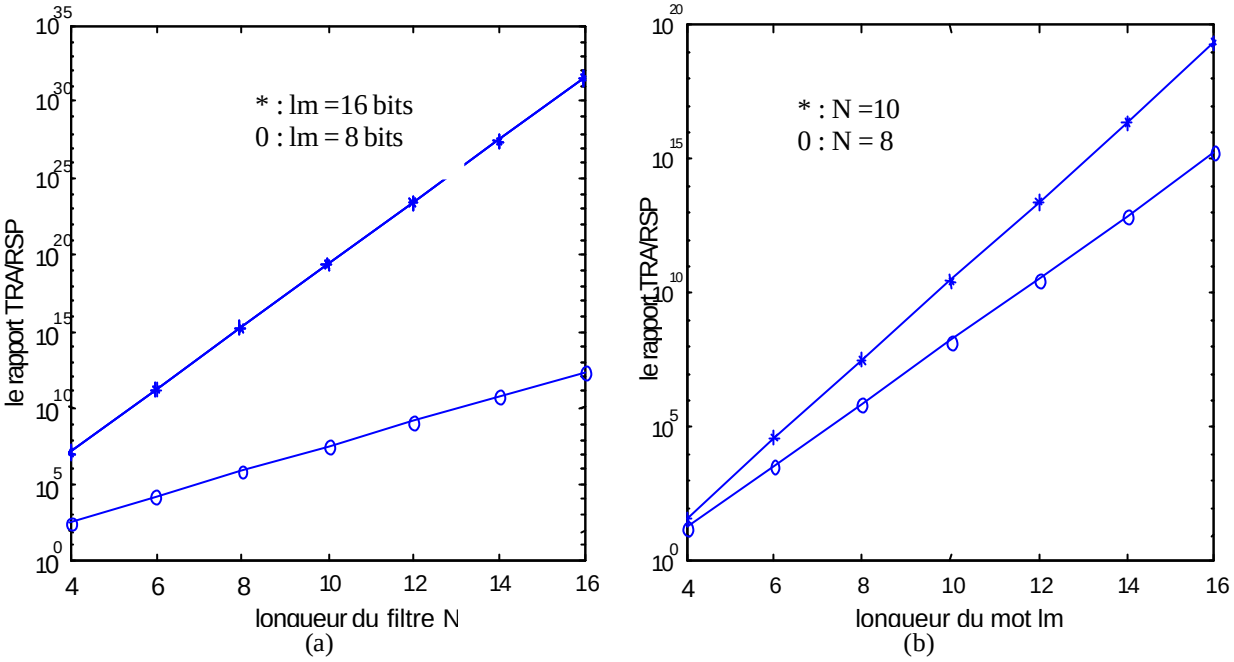


Fig. 60. Représentation du rapport $T_{RA/RSP}$ en fonction de N (a) et lm (b).

A partir des figures 60, nous remarquons que le rapport $T_{RA/RSP}$ augmente beaucoup plus avec N qu'avec lm. Ce rapport est très grand pour N=16 et lm = 16 bits, ce qui explique la réduction drastique du temps de calcul en utilisant la méthode RSP au lieu la méthode RA. Le paragraphe suivant exploite cet avantage de la méthode RSP pour des filtres de longueur égale et supérieure à 8.

IV. EXEMPLES DE FILTRES CONCUS PAR LA METHODE RSP DANS LE SENS DE L'ERREUR MINIMAX :

Un ensemble de 4 filtres repérés par les numéros de 1 à 4 choisis parmi plusieurs centaines de filtres ont été testés par la méthode RSP dans le sens minmax. Pour une étude comparative efficace, la plupart des filtres choisis ont des spécifications identiques à ceux du chapitre II, du chapitre III et de [4].

Dans tous les exemples, les filtres considérés sont des filtres R.I.F. à phase linéaire passe bas appartenants au cas2 (symétrique et longueur du filtre paire) et au cas1(symétrique et longueur du filtre impaire). Les algorithmes ont été élaborés dans un calculateur dont la fréquence CPU de travail est de 300 Mhz. Nous avons choisi de noter par :

IV.1. FILTRE 1 :

N=8.
lm = 8 bits.
wp = 0.159.
ws = 0.295.
avec Emm : erreur minmax dans BP et BA.

La synthèse du filtre choisi nous a donné les résultats qui sont groupés dans les tableaux suivants :

R.S.P.	P.M.C.Q.	P.M.C.	R.A.

Emm	tps	lmp (bits)	Em m	tps	Em m	Tps	Emm	tps
0.0635	0.06	3	0.0635	0.05 s	0.0576	0.05 s.	0.0635	26 h.

Tableau29 : Représentation de l'erreur minmax par les méthodes RSP, RA, PMC et PMCQ sur lm =8 bits.

Coefficients de filtre de longueur :8	R.S.P.	P.M.C.Q.	P.M.C.	R.A.
h(0) =h(7)	-0.0625000	-0.0625000	-0.0619896831395	-0.0625000
h(1)=h(6)	-0.0468750	-0.0468750	-0.0499878806747	-0.0468750
h(2) =h(5)	0.1718750	0.1718750	0.1723090749299	0.1718750
h(3)=h(4)	0.4140625	0.4140625	0.4109131351360	0.4140625

Tableau30: Tableau de coefficients du filtre obtenu par RSP, RA, PMC et PMCQ sur lm=8 bits.

Le tableau 29 présente la représentation de l'erreur au sens de Chebyshev pour les filtres conçus par les méthodes RSP, RA et PMC. Le tableau 30 présente les coefficients correspondants. Nous remarquons, que le filtre conçu par la méthode RSP est identique en performance à celui de PMCQ, mais plus mauvais comparé à celui de PMC dans un temps de calcul plus grand. De plus, nous constatons que la méthode R.S.P. retrouve les mêmes performances que la méthode RA sur la longueur de mot machine 'lm' correspondante dans un temps meilleur. Par conséquent, nous pouvons confirmer dans cet exemple, que le filtre de RSP présente la meilleure approximation dans le sens de Emm sur la longueur de mot lm définie. Nous avons schématisé l'allure des filtres correspondants sur la figure suivante :

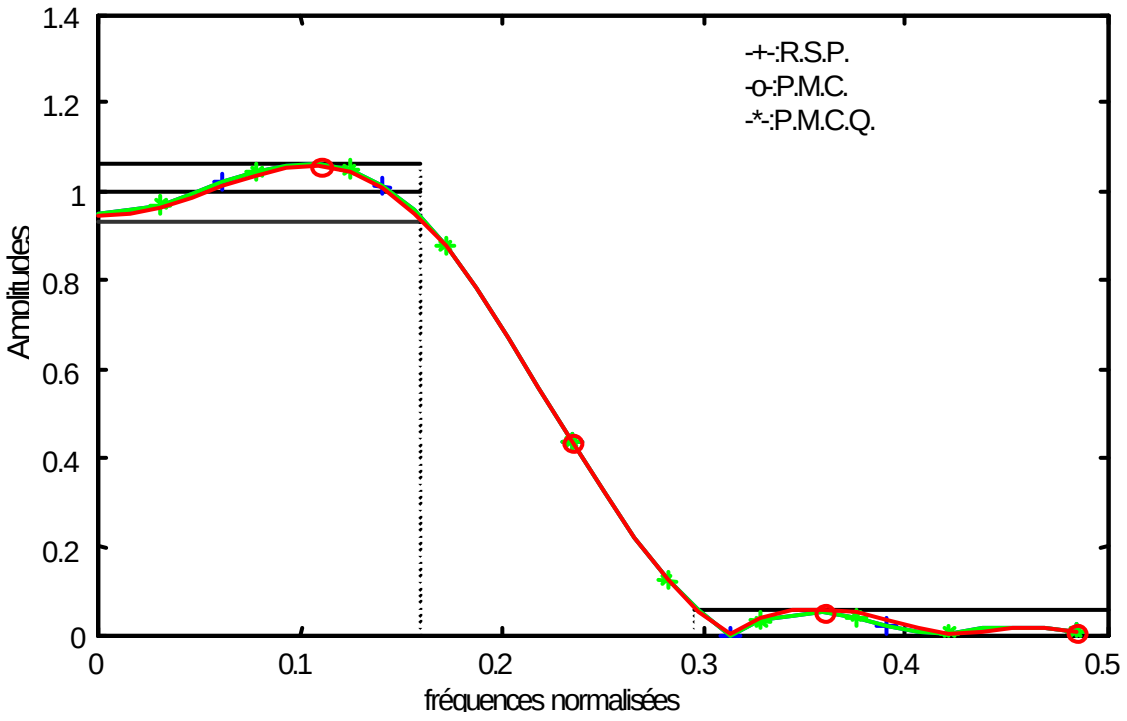


Fig. 61. Filtre de RSP, de PMC et PMCQ sur lm = 8 bits.

Cette figure montre que l'allure du filtre obtenu par la méthode R.S.P. est cohérent à celui obtenu par la méthode PMC avec une erreur minmax plus grande, due au nombre de valeurs discrètes réduit ne permettant pas une grande représentativité. Les trois filtres sont superposés à cause de la faible différence de l'erreur Emm entre eux.

Dans l'exemple suivant, nous avons étendu ce travail en utilisant les mêmes spécifications de filtre sur un mot de longueur plus grande 'lm=16 bits'.

IV.2. FILTRE 2 :

N=8.
lm = 16 bits.
wp = 0.159.
ws = 0.295.

La synthèse du filtre choisi par la méthode RSP nous a donné les résultats qui sont groupés dans les tableaux suivants :

R.S.P.			P.M.C.Q.		P.M.C.		R.A.	
Emm	tps	lmp (bits)	Emm	Tps	Emm	tps	Emm	tps
0.05759	19.6 s.	3	0.05765	2.91 s.	0.05766	0.05 s.	--------	-------

Tableau 31 : Représentation de l'erreur micmax par les méthodes RSP, RA, PMC, et PMCQ sur lm =16 bits.

Coefficients de filtre de longueur :8	R.S.P.	P.M.C.Q.	P.M.C.	R.A.
h(0) =h(7)	-0.06207275390625	-0.06201171875000	-0.06198968313957	-------
h(1)=h(6)	-0.04980468750000	-0.04998779296875	-0.04998788067471	-------
h(2) =h(5)	0.17211914062500	0.17230224609375	0.17230907492992	-------
h(3)=h(4)	0.41088867187500	0.41088867187500	0.41091313513603	-------

Tableau 32 : Tableau de coefficients du filtre obtenu par RSP, RA, PMC et PMCQ sur lm=16 bits.

Le tableau 31 présente la représentation de l'erreur au sens de Chebyshev pour les filtres conçus par les méthodes RSP, PMCQ et PMC. Le tableau 32 présente les coefficients correspondants. Dans ces tableaux nous n'avons pas pu représenter les résultats relatifs à la méthode RA, à cause de sa complexité de calcul, car cette méthode ne peut être utilisée pour des filtres dont la longueur est supérieure à huit et dans des espaces discrets à longueur de mot 'lm > 8 bits'. Nous remarquons, que le filtre conçu par la méthode RSP est meilleur en performance par rapport à celui de PMCQ et PMC dans un temps de calcul plus grand. Cela est du aux calculs numériques qu'effectue l'algorithme de PMC, ignorant la restriction de la longueur de mot dans ces calculs. A ce propos, notre méthode qui effectue la synthèse de filtres directement dans l'espace discret a permis d'obtenir une solution qui présente une meilleure approximation dans le sens de l'erreur Emm.

Nous remarquons aussi que l'erreur Emm du filtre de PMCQ est plus petite que celle de PMC malgré que ce dernier est considéré comme le filtre qui présente la meilleure approximation dans le sens de l'erreur minmax. Cette confusion est due aux approximations numériques et opérations arithmétiques qu'effectue l'algorithme de PMC dans son calcul des coefficients sur calculateur (à 64 bits). il s'avère qu'il existe des bits erronés dans le mot sur lequel les coefficients ont été calculés. Alors par quantification de ces coefficients, il se peut que l'erreur Emm par élimination de ces bits.

A partir du tableau 32, nous avons obtenu la figure suivante :

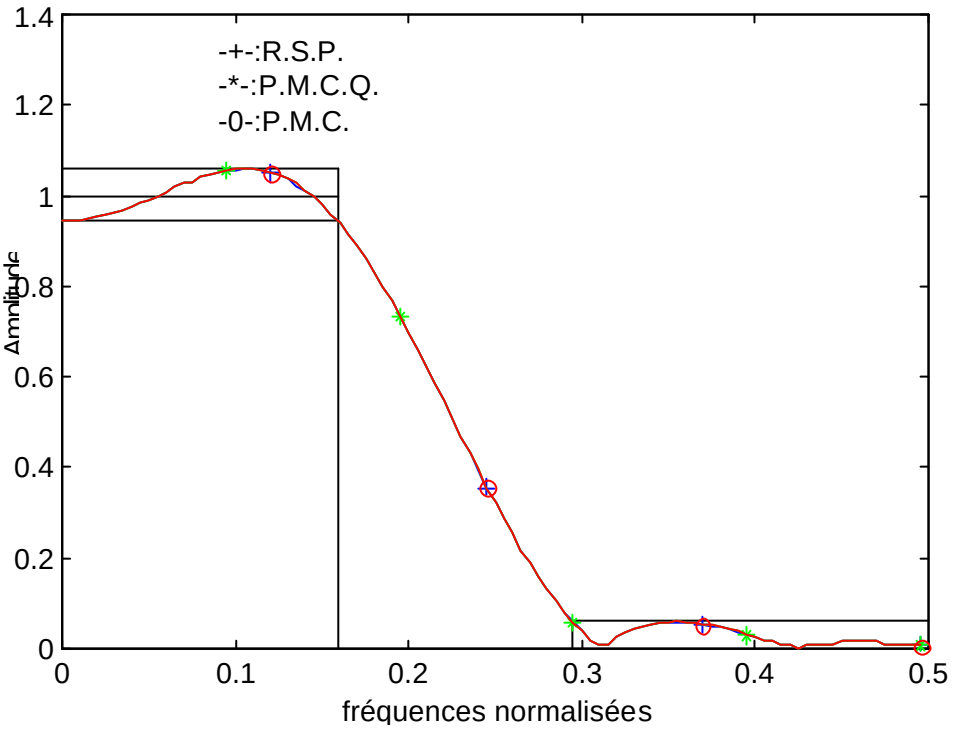


Fig. 62. Filtre de RSP, de PMC et PMCQ sur lm = 16 bits.

Cette figure montre que les trois filtres sont cohérents et superposés à cause de la faible différence de Emm entre eux.

Afin de confirmer ces résultats, nous avons éprouvé la méthode RSP avec d'autres exemples pour des filtres de longueur élevée et des mots de longueur plus grande parmi lesquels, nous présenterons deux dans les paragraphes suivants.

IV.3. FILTRE 3:

N=21.
lm = 6 bits.
wp = 0.8.
ws = 0.16.

Ces spécifications sont prises de la référence [4].

La synthèse du filtre choisi, nous a donné les résultats qui sont groupés dans les tableaux suivants :

R.S.P.			P.M.C.Q.		P.M.C.		Reference [4]	
Emm	tps	lmp(bits)	Emm	tps	Emm	tps	Emm	Tps
0.08605	79256 s.	3	0.12379	0.06 s.	0.05822	0.05 s.	0.0711	5 s.

Tableau 33 : Représentation de l'erreur Emm par les méthodes RSP, PMC, et PMCQ et celle en [4] sur lm =6 bits.

Coefficients de filtre de longueur :8	R.S.P.	P.M.C.Q.	P.M.C.
h(0) =h(20)	0	0	.00606215177645
h(1)=h(19)	0	0	-0.00393608214192
h(2) =h(18)	0	-0.03125	-0.01596945663947
h(3)=h(17)	-0.03125	-0.03125	-0.03029600895028
h(4)=h(16)	-0.03125	-0.03125	-0.03646325795463
h(5) =h(15)	-0.03125	-0.03125	-0.02259390811843
h(6)=h(14)	0	0.03125	0.01808613873788
h(7)=h(13)	0.09375	0.09375	0.08158952586834
h(8) =h(12)	0.15625	0.15625	0.15222097143151
h(9)=h(11)	0.21875	0.21875	0.20776889335317
h(10)	0.25000	0.21875	0.22883679864436

Tableau 34 : Tableau de coefficients du filtre obtenu par RSP, PMC et PMCQ sur lm=6 bits.

Le tableau 33 présente l'erreur au sens de Chebyshev pour les filtres conçus par les méthodes RSP, PMCQ, PMC et celui de la référence[4]. Le tableau 34 présente les coefficients correspondants. Nous remarquons, que le filtre conçu par la méthode R.S.P. est meilleur en performance par rapport à celui de PMCQ, mais mauvais comparé à celui de PMC à cause de lm petite qui réduit la représentativité des coefficients. Le temps de calcul est très grand comparé à celui de PMC et PMCQ. dans [4], il a été relevé que pour les mêmes spécifications de filtre, nous avons un (Emm|tps) de (0.0711|5 s.) meilleur que celui de la méthode RSP. Le PMCQ relevé dans [4] est de (0.0781) est différent du notre qui est de (0.12379), c'est pourquoi, les résultats de [4] restent à vérifier.

A partir des coefficients du tableau 34, nous avons schématisé les filtres suivants :

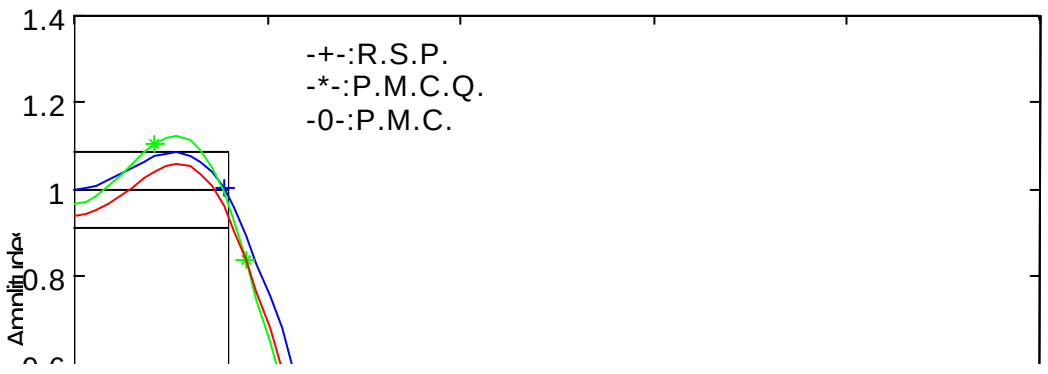


Fig. 63. Filtre de RSP, de PMC et PMCQ sur lm = 6 bits.

Nous remarquons que le filtre de la méthode PMCQ sort du gabarit prescrit par celui de la méthode RSP, tandis que le filtre de PMC présente la meilleure allure dans le sens de Emm.

IV.4. FILTRE 4 :

N=16.
lm = 20 bits.
wp = 0.307.
ws = 0.35.

La synthèse du filtre choisi, nous a donné les résultats qui sont groupés dans les tableaux suivants :

R.S.P.			P.M.C.Q.		P.M.C.	
Emm	Tps	lmp(bits)	Emm	tps	Emm	Tps
0.13588	11830 s.	3	0.13590	218 s.	0.13590	0.05 s.

Tableau 35 : Représentation de l'erreur minmax par les méthodes RSP, PMC, et PMCQ sur lm =20 bits.

Coefficients de filtre de longueur :8	R.S.P.	P.M.C.Q.	P.M.C.
h(0) =h(15)	0.02908706665039	0.02938461303711	0.02938473501292
h(1)=h(14)	0.07563781738281	0.07599258422852	0.07599166739243
h(2) =h(13)	-0.06398963928223	-0.06394004821777	-0.06393950840571
h(3)=h(12)	0.01408004760742	0.01354217529297	0.01354139770060
h(4)=h(11)	0.07048034667969	0.06973648071289	0.06973666708439
h(5) =h(10)	-0.11213493347168	-0.11265563964844	-0.11265578652321
h(6)=h(9)	0.00950050354004	0.00970077514648	0.00970095908901
h(7)=h(8)	0.54527664184570	0.54603195190430	0.54603124395824

Tableau 36 : Tableau de coefficients du filtre obtenu par RSP, PMC et PMCQ sur lm=20 bits.

A partir du tableau 35, nous remarquons que le filtre de la méthode RSP présente la plus faible erreur Emm comparée à celle de PMC et PMCQ. le temps de calcul correspondant est 50 fois plus grand que celui de PMCQ. Les filtres de PMC et PMCQ présentent des coefficients identiques au sixième chiffre significatif près.

A partir des coefficients du tableau 36, nous avons les filtres suivants :

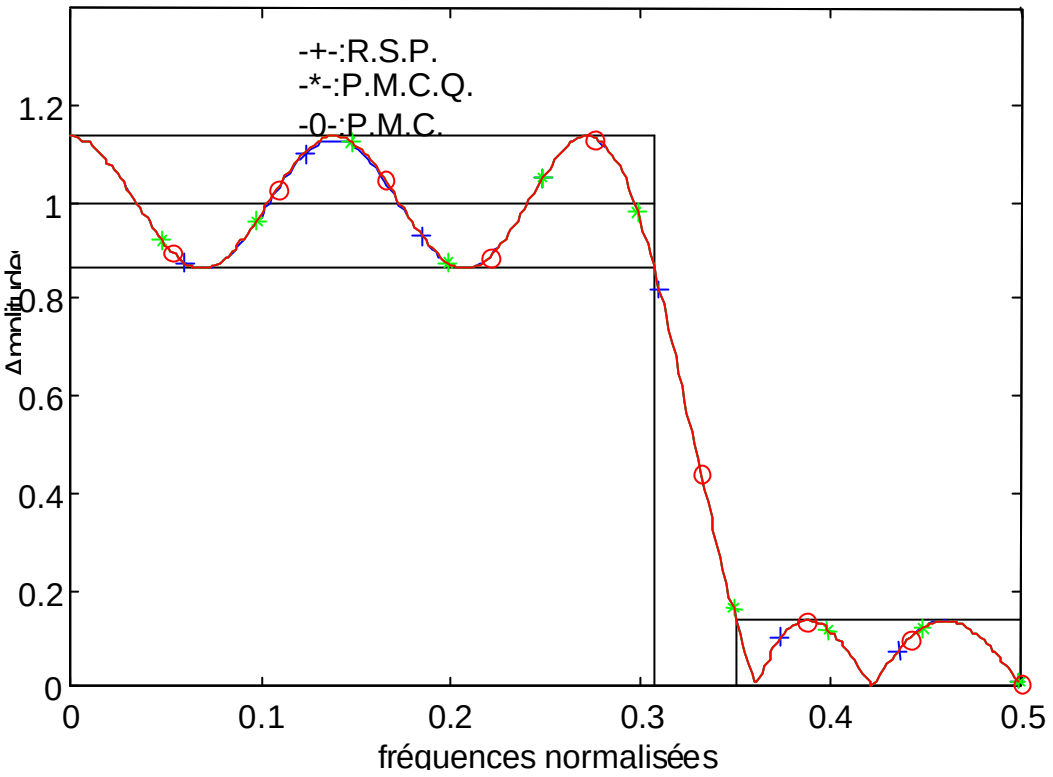


Fig. 64. Filtre de RSP, de PMC et PMCQ sur lm = 20 bits.

Cette figure montre que le filtre des méthodes RSP, PMC, et PMCQ sont superposés et cohérents. La distinction entre eux ne peut pas être faite à l'œil nu. Par conséquent, nous allons faire une étude de ses résultats dans le paragraphe suivant.

IV.5. ETUDE DES RESULTATS DE RSP :

A partir des résultats des exemples précédents , nous remarquons que les filtres de RSP sont équivalents ou meilleurs que ceux de PMCQ et mêmes que PMC pour lm > 16 bits (exemple 2 et exemple 4). Le principal problème est le temps de calcul qui reste prohibitif malgré l'espace discret qui a été drastiquement réduit par rapport à celui de RA.

Pour une comparaison significative une étude statistique a été menée sur plusieurs centaines de filtres sur les performances de la méthode RSP. Par conséquent, nous présentons les résultats correspondants (Emm | tps) en fonction de N et lm sur les figures suivantes.

Emm : étant la moyenne de tous les Emm des filtres testés.
tps : étant la moyenne de tous les tps des filtres testés.

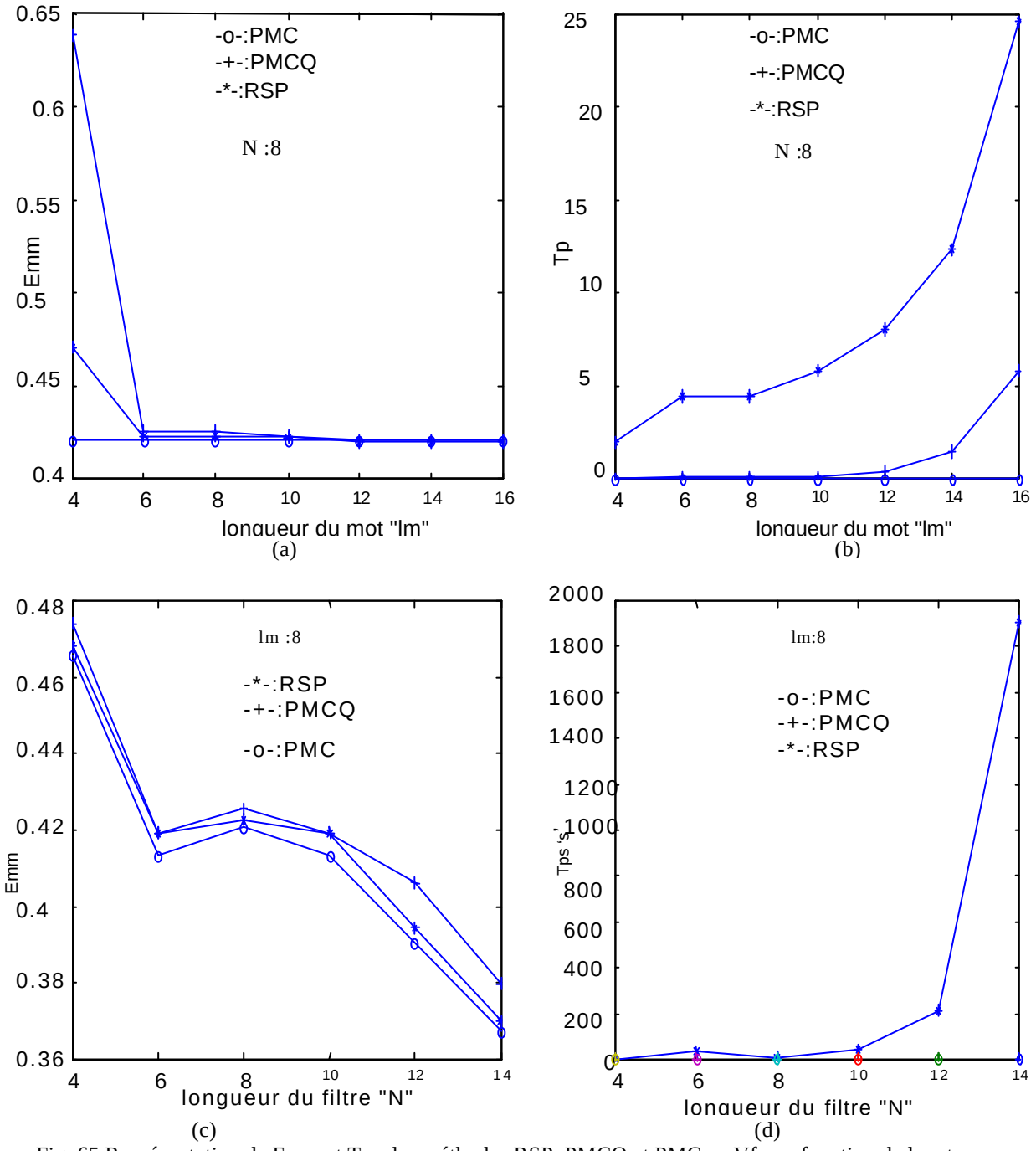


Fig. 65 Représentation de Emm et Tps des méthodes RSP, PMCQ et PMC en Vfx en fonction de lm et N.

Pour les figures 65 a et b N a été fixé à 8, tandis que pour les figures 65 c. et d lm a été fixé à 8 bits. Pour les trois méthodes nous remarquons que l'erreur Emm diminue quand N et lm augmentent tandis que le temps de calcul augmentent dans ce cas. Dans le cas de la méthode RSP, nous remarquons que le temps de calcul augmente beaucoup plus avec le N qu'avec lm. En performances nous remarquons que la courbe de l'erreur Emm de la méthode RSP se trouve entre celle de PMC et PMCQ. Dans le cas de la figure 65.a, l'erreur Emm de la méthode RSP se superpose avec celle de PMC pour lm > 10 bits. Ces résultats

montrent que la méthode RSP donnent de bonnes performances pour un nombre fini de bits meilleures que celles de PMCQ, mais le temps de calcul reste prohibitif surtout quand N augmente.
Dans ce suit, nous allons utiliser le critère de l'erreur quadratique moyenne pour une bonne évaluation de la méthode RSP.

V. EXEMPLES DE FILTRES CONCUS PAR LA METHODE RSP AU SENS DE L'ERREUR QUADRATIQUE MOYENNE :

La même étude faite dans le sens de Emm dans le paragraphe IV, sera refaite dans ce paragraphe dans le sens de Ems. Les mêmes spécifications des quatre filtres sont reportés dans ce paragraphe. Notre but dans ce paragraphe est de tester les capacités de la méthode RSP en utilisant le critère d'erreur de Ems et en relever la qualité de sortie correspondante afin de confirmer les conclusions du paragraphe IV.

V.1. FILTRE 1 :

N=8.
lm = 8 bits.
wp = 0.159.
ws = 0.295.

La synthèse du filtre choisi nous a donné les résultats qui sont groupés dans les tableaux suivants :

R.S.P.			P.M.C.Q.		P.M.C.		R.A.	
Ems	tps	lmp(bits)	Ems	tps	Ems	Tps	Ems	Tps
0.0319	0.5 s.	3	0.0367	0.05 s	0.0374	0.05 s.	0.0319	26 h.

Tableau37 : Représentation de l'erreur Ems par les méthodes RSP, RA, PMC et PMCQ sur lm =8 bits en Vfx.

Coefficients de filtre de longueur :8	R.S.P.	P.M.C.Q.	P.M.C.	R.A.
h(0) =h(7)	-0.0546875	-0.0625000	-0.06198968313957	-0.0546875

h(1)=h(6)	-0.0390625	-0.0468750	-0.04998788067471	-0.0390625
h(2) =h(5)	0.1640625	0.1718750	0.17230907492992	0.1640625
h(3)=h(4)	0.4140625	0.4140625	0.41091313513603	0.4140625

Tableau 38 : Tableau de coefficients du filtre obtenu par RSP, RA, PMC et PMCQ sur lm = 8 bits en Vfx.

Les même remarque de IV.1. sont vérifiées dans cet exemple. Nous remarquons, que le filtre conçu par la méthode RSP est identique en performance de RA dans un temps de calcul plus petit. Comparé à celui de PMCQ et de PMC, ce filtre présente une erreur Ems plus petite dans un temps de calcul plus grand. Comme dans le paragraphe IV.2., nous remarquons aussi que l'erreur Ems du filtre de PMCQ est plus petite que celle de PMC. Cela est du aux opérations arithmétiques qu'effectue l'algorithme de PMC dans son calcul des coefficients sur calculateur (à 64 bits). cela induit des bits erronés dans le mot sur lequel les coefficients ont été calculés. Alors par quantification de ces coefficients, nous avons obtenu une erreur Ems plus petite, en éliminant ces bits.

Nous avons schématisé l'allure des filtres correspondants sur la figure suivante :

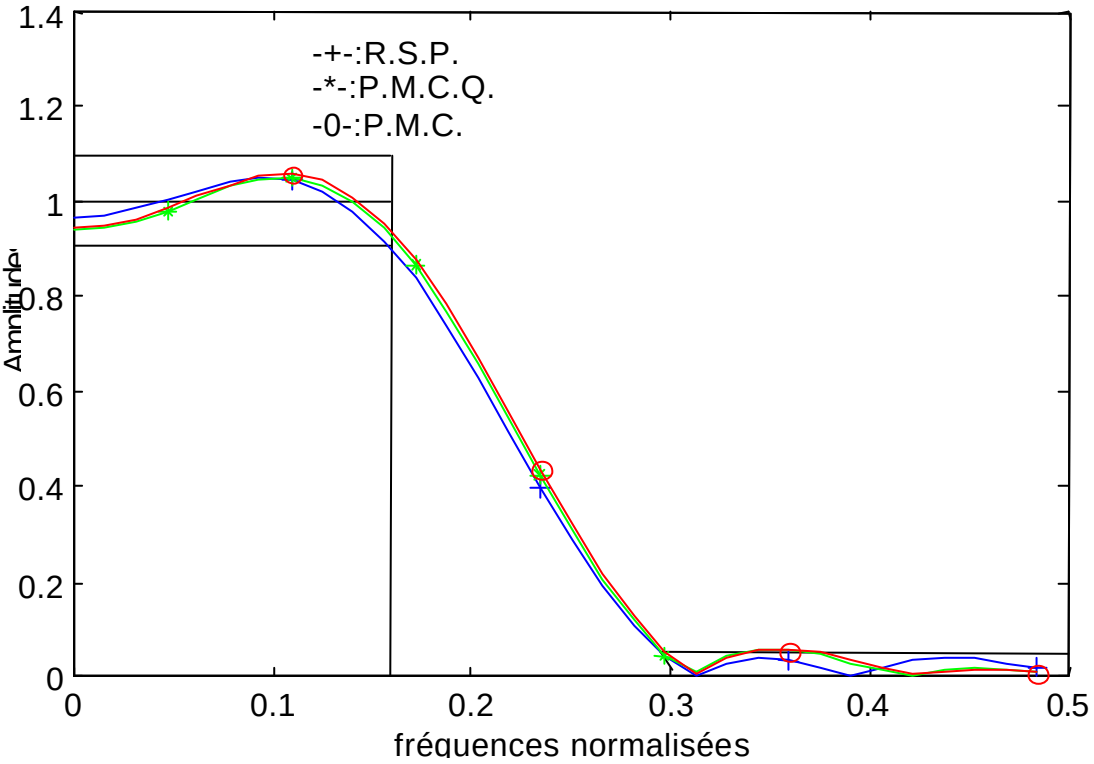


Fig. 66. Filtre de RSP, de PMC et PMCQ sur lm = 8 bits.

Cette figure montre que le filtre obtenu par la méthode RSP, comparé à celui de PMC, présente l'erreur Emm plus grande en BP et plus petite en BA.

IV.2. FILTRE 2 :

N=8.
lm = 16 bits.
wp = 0.159.
ws = 0.295.

La synthèse du filtre choisi, nous a donné les résultats qui sont groupés dans les tableaux suivants :

R.S.P.			P.M.C.Q.		P.M.C.		R.A.	
Ems	tps	lmp(bits)	Ems	tps	Ems	tps	Ems	tps
0.0314	19.8 s.	3	0.037432	2.91 s.	0.037438	0.05 s.	-------	-------

Tableau39 : Représentation de l'erreur Ems par les méthodes RSP, RA, PMC,et PMCQ sur lm =16 bits en Vfx.

Coefficients de filtre de longueur :8	R.S.P.	P.M.C.Q.	P.M.C.	R.A.
H(0) =h(7)	-0.05529785156250	-0.06201171875000	-0.06198968313957	-------
h(1)=h(6)	-0.03802490234375	-0.04998779296875	-0.04998788067471	-------
H(2) =h(5)	0.16790771484375	0.17230224609375	0.17230907492992	-------
h(3)=h(4)	0.40997314453125	0.41088867187500	0.41091313513603	-------

Tableau 40 : Tableau de coefficients du filtre obtenu par RSP, RA, PMC et PMCQ sur lm=16 bits.

Les mêmes remarques faites dans l'exemple précédent sont vérifiées pour ce cas. L'erreur Ems RSP est plus petite que celle de PMC et PMCQ dans un temps légèrement plus grand. Dans ces tableaux nous n'avons pas pu représenter les résultats relatifs à la méthode RA, à cause de la complexité algorithmique. Nous remarquons que l'erreur Ems du filtre de PMCQ s'approche plus de celle de PMC que celle de l'exemple précédent. Cela est du au 8 bits de précision que nous avons ajouté aux coefficients de PMCQ. Les bits erronés ajoutés ont induit une augmentation de la valeur de l'errer Ems du filtre de PMCQ.

A partir des coefficients du tableau 40, nous avons les filtres suivants :

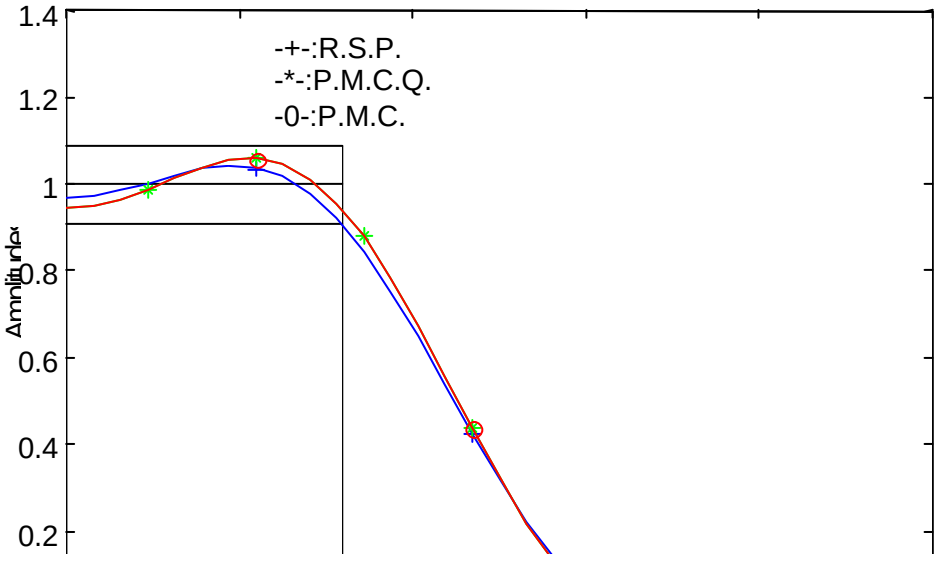


Fig. 67. Filtre de RSP, de PMC et PMCQ sur lm = 16 bits.

Cette figure montre que le filtre obtenu par la méthode RSP, comparé à celui de PMC, présente l'erreur Emm la plus grande en BP et en BA

V.3. FILTRE 3:

N=21.
lm = 6 bits.
wp = 0.08.
ws = 0.16.
Ces spécifications ont été prises de la référence [4]. La synthèse du filtre choisi, nous a donné les résultats qui sont groupés dans les tableaux suivants :

R.S.P.			P.M.C.Q.		P.M.C.	
Ems	Tps	Lmp(bits)	Ems	Tps	Ems	tps
0.03294	78956 s.	3	0.03294	0.06 s.	0.01512	0.05 s.

Tableau 41: Représentation de l'erreur Ems par les méthodes RSP, PMC, et PMCQ sur lm =6 bits en Vfx.

Coefficients de filtre de longueur :8	R.S.P.	P.M.C.Q.	P.M.C.
h(0) =h(20)	0	0	0.01525172358834
h(1)=h(19)	0	0	0.00594642537256
h(2) =h(18)	0	0	-0.00648600408011
h(3)=h(17)	-0.03125	-0.03125	-0.02541699016895
h(4)=h(16)	-0.03125	-0.03125	-0.03897848330871
h(5) =h(15)	-0.03125	-0.03125	-0.03172921328924
h(6)=h(14)	0	0	0.00682232594748
h(7)=h(13)	0.06250	0.06250	0.07426100199557
h(8) =h(12)	0.15625	0.15625	0.15299844516878
h(9)=h(11)	0.21875	0.21875	0.21646571824279
h(10)	0.25000	0.25000	0.24078398536844

Tableau 42 : Tableau de coefficients du filtre obtenu par RSP, RA, PMC et PMCQ sur lm=6 bits en Vfx.

Le tableau 41 montre que RSP a obtenu les mêmes performances de PMCQ dans un temps de calcul 1 million de fois plus grand. Le but de cet exemple est de montrer lorsque N augmente, cette méthode est coûteuse en temps de calcul pour obtenir des résultats retrouvés par PMCQ en un temps très petit. La méthode PMC a

donné le filtre a plus faible erreur Ems à cause de grande la longueur de mot sur laquelle sont représentés les coefficients(tableau 42).

A partir de ces coefficients, nous avons obtenu figure suivante :

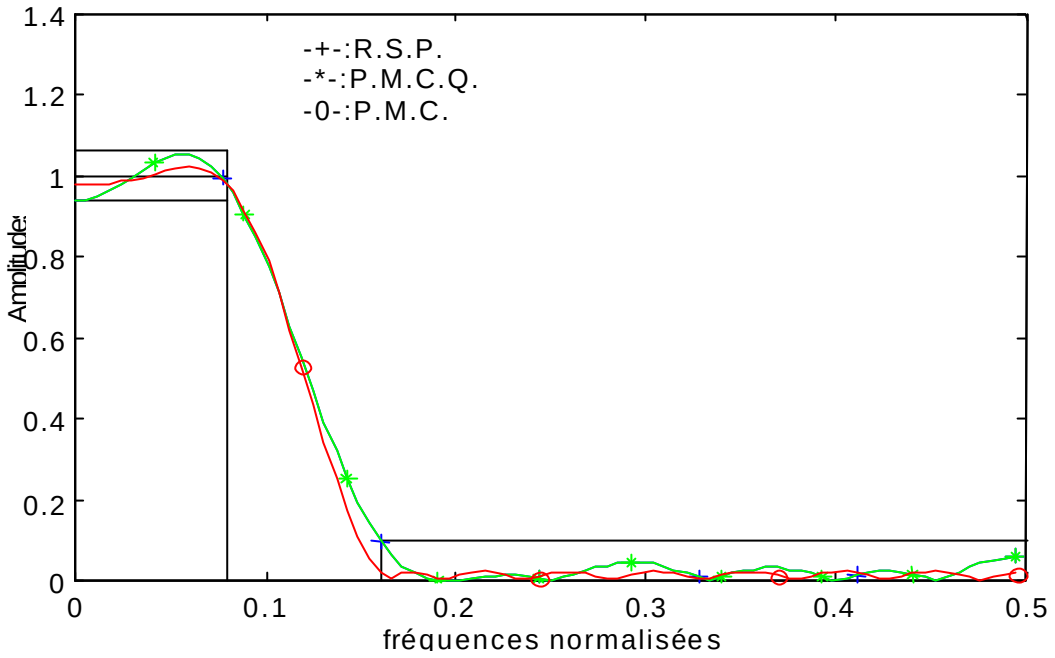


Fig. 68. Filtre de RSP, de PMC et PMCQ sur lm = 6 bits en Vfx.

Cette figure montre que les filtres de PMC et PMCQ présentent une faible erreur Emm comparée à celle de la méthode RSP et cela est visible aux frontières des fréquences de coupure.

V.4. FILTRE 4 :

N=16.
lm = 20 bits.
wp = 0.307.
ws = 0.35.

La synthèse du filtre choisi, nous a donné les résultats qui sont groupés dans les tableaux suivants :

R.S.P.			P.M.C.Q.		P.M.C.	
Ems	Tps	lmp(bits)	Ems	tps	Ems	Tps
0.05440	11896 s.	3	0.09618	219 s.	0.09618	0.05 s.

Tableau 43 : Représentation de l'erreur Ems par les méthodes RSP, PMC,et PMCQ sur lm = 20 bits en Vfx.

Coefficients de filtre de longueur :8	R.S.P.	P.M.C.Q.	P.M.C.

h(0) =h(15)	0.00599670410156	0.02938461303711	0.02938473501292
h(1)=h(14)	0.03017425537109	0.07599258422852	0.07599166739243
h(2) =h(13)	-0.04647445678711	-0.06394004821777	-0.06393950840571
h(3)=h(12)	0.00787734985352	0.01354217529297	0.01354139770060
h(4)=h(11)	0.06958389282227	0.06973648071289	0.06973666708439
h(5) =h(10)	-0.11115837097168	-0.11265563964844	-0.11265578652321
H(6)=h(9)	0.00874137878418	0.00970077514648	0.00970095908901
H(7)=h(8)	0.54604911804199	0.54603195190430	0.54603124395824

Tableau 44 : Tableau de coefficients du filtre obtenu par RSP, RA, PMC et PMCQ sur lm=6 bits en Vfx.

A partir du tableau 43, nous remarquons que le filtre de la méthode RSP présente la plus faible erreur Ems comparée à celle de PMC et PMCQ. le temps de calcul correspondant est 50 fois plus grand que celui de PMCQ. Comme dans le paragraphe IV.4, les filtres de PMC et PMCQ présentent des coefficients identiques au sixième chiffre significatif près, comme dans le paragraphe IV.4.. A partir de ces coefficients, nous avons schématisé l'allure des filtres correspondants.

Nous avons obtenu la figure suivante qui présente les réponses en amplitude des filtres dont les coefficients sont notés dans le tableau 44 :

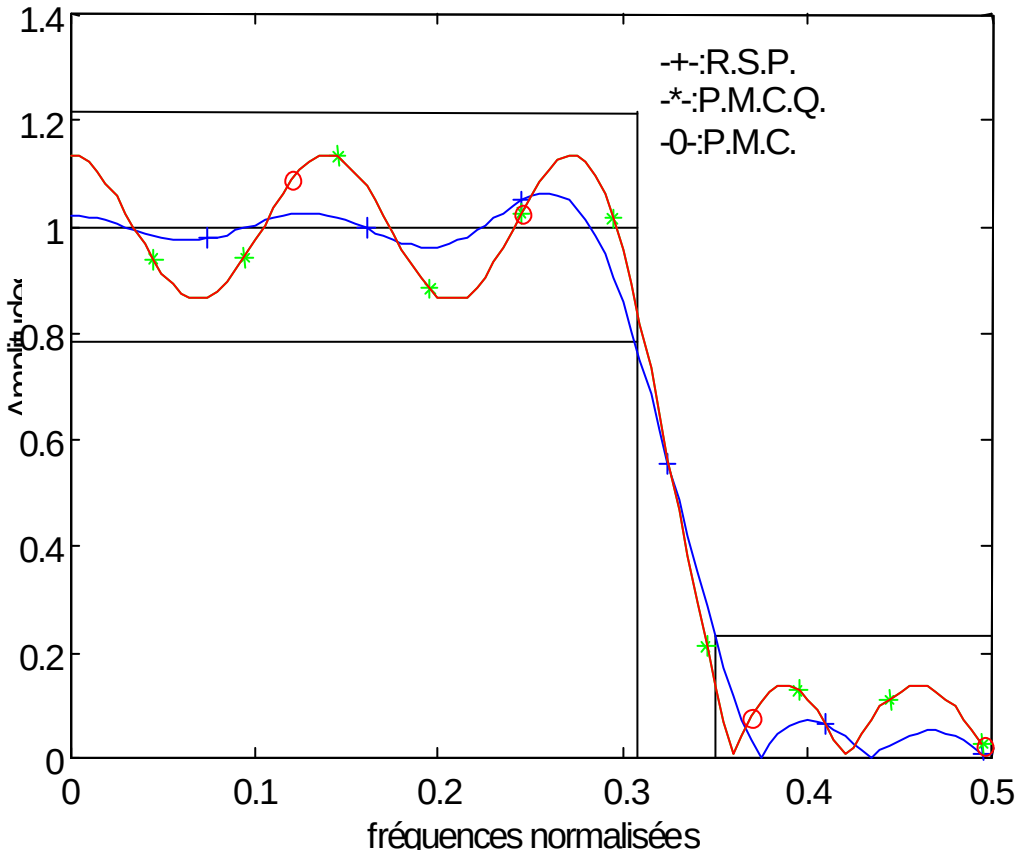


Fig. 69. Filtre de RSP, de PMC et PMCQ sur lm = 20 bits.

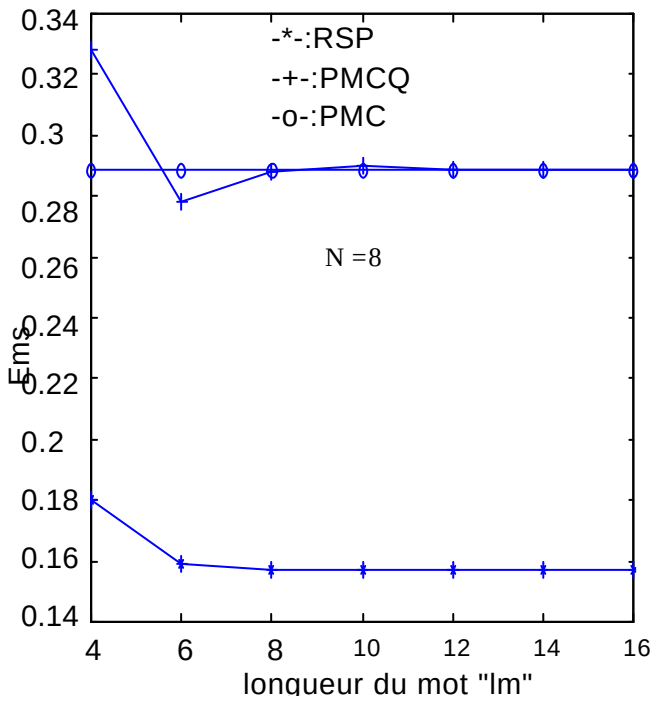
Cette figure montre que l'allure du filtre obtenu par la méthode RSP présente une erreur maximale plus petite que celle du filtre de PMC dans la BP et dans la BA à l'exception des frontières des fréquences de coupure.

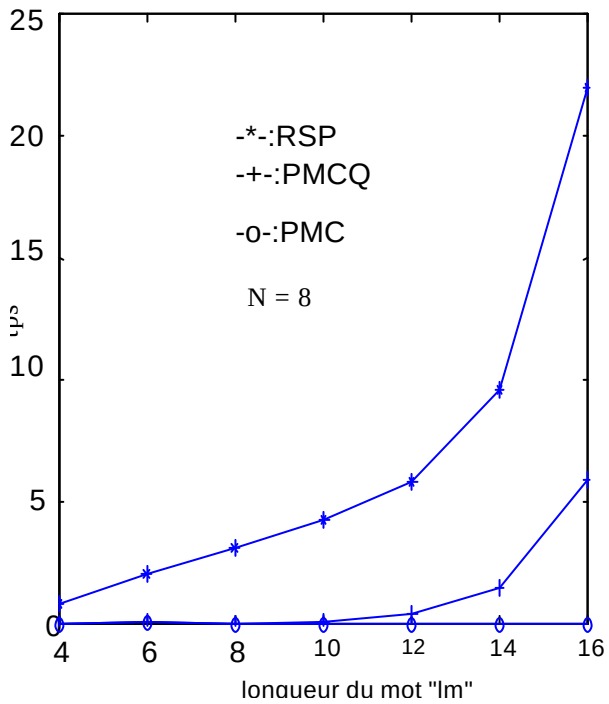
Pour une interprétation significative de ces résultats, nous avons fait une étude statistique dans le paragraphe suivant.

V.5. ETUDE DES RESULTATS DE RSP DANS LE SENS DE L'ERREUR Ems :

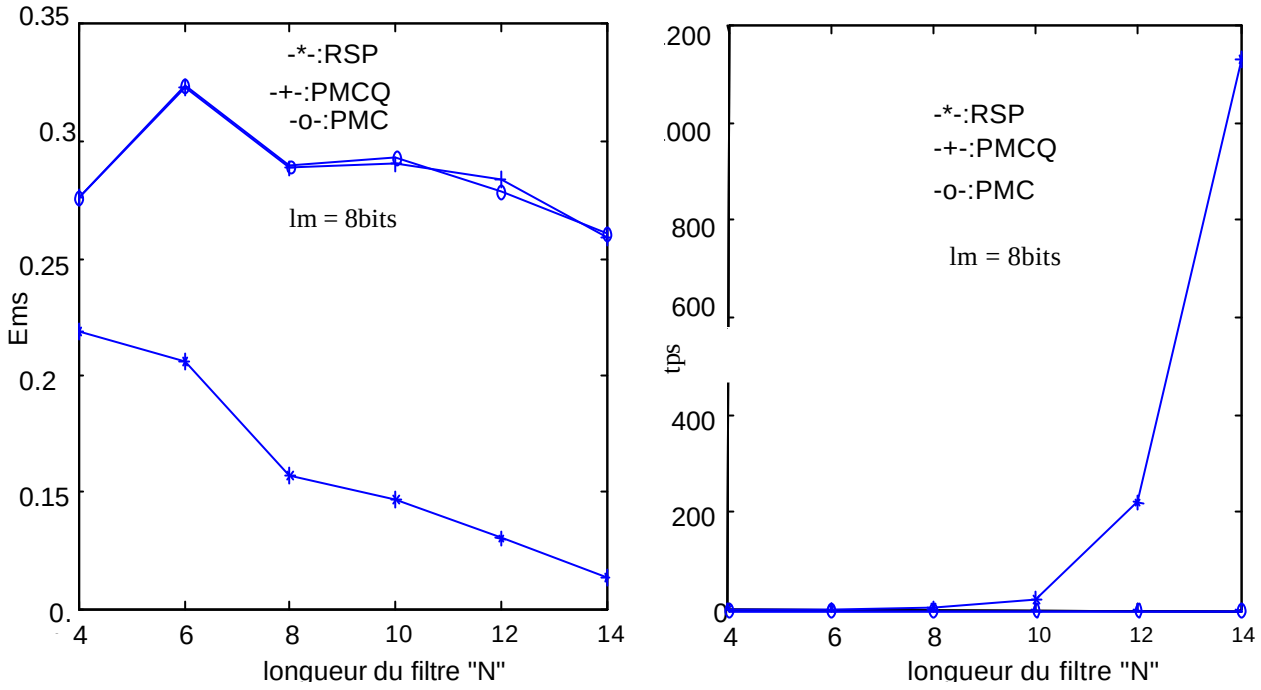
A partir des résultats des exemples précédents , nous remarquons que les filtres de RSP sont équivalents ou meilleurs que ceux de PMCQ et mêmes que PMC pour certains cas. Le principal problème est le temps de calcul (exemple 3) qui reste prohibitif malgré l'espace discret qui a été drastiquement réduit comparé à celui de RA, où parfois nous retrouvons les résultats de PMCQ avec un temps très grand.

Pour une comparaison significative une étude statistique a été menée sur plusieurs centaines de filtres sur la qualité de sortie de la méthode RSP. Nous donnons les résultats obtenus sous forme de moyenne des (Ems | tps) en fonction de N et lm sur les figures suivantes.





(a) (b)



(c) (d)

Fig. 70 Représentation de Ems et Tps des méthodes RSP, PMCQ et PMC en Vfx en fonction de lm et N.

Pour les figures 70 a et b N a été fixé à 8, tandis que pour les figures 70 c. et d lm a été fixé à 8 bits. Pour les trois méthodes nous remarquons que l'erreur Ems diminue quand N et lm augmentent tandis que le temps de calcul augmentent dans ce cas. Comme dans le cas de l'utilisation de l'erreur Emm, nous remarquons que le temps de calcul augmente beaucoup plus avec le N qu'avec lm. En performances nous remarquons que la courbe de l'erreur Ems de la méthode RSP se trouve toujours au dessous des celles de PMC et PMCQ. Ces résultats montrent que la méthode RSP donnent les meilleures performances dans le sens de l'erreur Ems comparées à celles de PMC et PMCQ, mais le temps de calcul est plus grand surtout quand N augmente.

Dans ce suit, nous allons donner une étude comparative de l'utilisation des deux critères de l'erreur Ems et Emm pour la méthode RSP.

VI. DISCUSSION COMPARATIVE DES RESULTATS OBTENUS A L'AIDE DES DEUX CRITERES EMS ET EMM :

D'après l'étude théorique de la méthode R.S.P., nous confirmons à l'aide des exemples pris précédemment que cette méthode retrouve, au pire des cas, les performances de la méthode de PMC sinon mieux. D'une façon générale, nous pouvons considérer que la méthode R.S.P. fournit des filtres meilleurs dans les deux sens (Ems et Emm) que ceux de PMCQ.

Dans le cas de l'erreur Ems, nous avons remarqué que la plupart des filtres de la méthode R.S.P. sont meilleurs en performances que ceux de PMC et de PMCQ. par conséquent, le temps de calcul est prohibitif quand N>24..

Dans ce cas, nous avons remarqué que la méthode R.S.P. retrouve, au pire des cas, les performances du filtre de Parks-McClellan sinon mieux, dans tous les exemples choisis. Parmi les problèmes que nous avons rencontré pour la méthode RSP a été l'utilisation du critère d'erreur minmax pour l'évaluation du filtre. Le filtre

considéré comme celui de meilleure approximation dans le sens de Emm est celui qui suit vérifie l'éq. 66 ; en d'autres termes, c'est le filtre dont la réponse en amplitude présente les erreurs maximales les plus faibles dans la bande passante et dans la bande atténuée à la fois. Dans notre recherche progressive, nous avons rencontré des filtres avec une erreur maximale dans la bande passante plus petite que celle du filtre précédemment retrouvé (considéré provisoirement comme le meilleur dans le sens de Tchebyshev), et une erreur maximale dans la bande atténuée plus grande. Le choix entre ces deux filtres est difficile. Dans le cas où le filtre provisoire est celui de PMC, nous avons utiliser la fonction poids afin de mettre dans un niveau requis les erreurs dans la BP et la BA. Tandis que dans un cas général, la précédente solution est maintenue.

VII. CONCLUSION

Dans ce chapitre, une nouvelle approche de la synthèse de filtres numériques dans l'espace discret des coefficients qui utilise une nouvelle stratégie de branchement de la méthode de recherche arborescente est présentée. Le résultat fructueux de cette approche est l'application de cette méthode nommée RSP pour des filtres dans un espace discret de longueur de mot très grande. Le temps du calcul avec un tel mot, utilisant la méthode de recherche arborescente est prohibitif. Les résultats obtenus, comparés à ceux d'autres méthodes telles que DMCI et PMCQ, sont meilleurs ou égaux dans tous les cas en performances. Le temps de calcul correspondant à un mot de grande longueur est plus petit que celui de DMCI. Dans les exemples la limitation du domaine de recherche ne semble pas dégrader les performance de l'algorithme. le problème relatif à cette méthode est lorsque la longueur de filtre est supérieure à 24, l'utilisation de la méthode RSP est prohibitif.

En perspective, l'amélioration du temps de calcul de l'algorithme pour les filtres de grande longueur ouvre un champ à explorer.

CONCLUSION

Ce travail a permis de mettre en évidence les problèmes liés au filtrage numérique et spécialement ceux concernant la mise en œuvre des coefficients du filtre sur un processeur de signaux de longueur de mot finie. A ce propos, nous avons proposé des approches permettant d'évaluer la méthode adéquate de l'optimisation de la mise en œuvre des filtres sur un processeur.

Dans ce contexte, nous avons étudié les effets de la quantification des coefficients, du choix de la représentation binaire, du choix de critère d'évaluation et du choix de la méthode de synthèse de filtre sur la qualité de sortie (précision et temps de calcul) du filtre. Comme nous l'avons montré, les méthodes existantes dans la littérature scientifique concernant la synthèse de filtres, telles que celles de Parks-McClellan avec quantification et celles du laboratoire Signaux et Systèmes (méthode RA et méthode DMC) donnent soit un temps de calcul prohibitif, soit des résultats peu performants. Notre recherche dans ce contexte a été de mettre en œuvre trois approches permettant d'améliorer la qualité de sortie des filtres mis en œuvre.

Chronologiquement, la première approche consiste au choix de la représentation binaire entre celle en virgule fixe, en virgule flottante ou en SDPD, procurant les meilleures performances dans un temps de calcul acceptable. A ce propos, nous avons choisi d'étudier la qualité de sortie des deux méthodes de synthèse de filtres proposées par le laboratoire Signaux et Systèmes, la méthode RA et la méthode DMC, pour respectivement leur meilleures performances et leur convergence rapide. Pour des raisons de complexité algorithmique le laboratoire Signaux et Systèmes a utilisé la représentation SDPD. Notre principale contribution a été d'étendre les travaux de celui-ci afin de voir la perte de performance comparativement au gain de complexité découlant de l'utilisation de la représentation SDPD. Pour la méthode RA, nous avons montré que l'utilisation de cette représentation a permis de réduire drastiquement le temps de calcul pour une grande perte en performance comparativement à l'utilisation des représentations en virgule fixe ou en virgule flottante. Alors que pour la méthode DMC, nous avons constaté qu'il y a une grande perte de performance en utilisant la représentation SDPD pour un temps de calcul, équivalent à celui des représentations en virgule fixe et en virgule flottante.

La deuxième approche consiste à améliorer les performances de la méthode DMC qui possède une convergence rapide, mais les résultats ne sont pas, dans tous les cas, équivalents à ceux de la méthode RA. A ce propos, nous avons élaboré une méthode itérative annexée à la méthode DMC, consistant à améliorer les performances de cette dernière en plusieurs itérations. Cette méthode nommée DMCI a été testée dans les trois représentations binaires. Nous avons montré que les représentations en virgule fixe et en virgule flottante sont mieux adaptées à DMCI et procurent les meilleurs résultats dans le sens de l'erreur quadratique moyenne. La méthode DMCI est plus lente que la méthode DMC, mais nettement plus rapide que la méthode RA, son temps demeure acceptable.

La troisième approche consiste à accélérer la convergence de la méthode RA tout en gardant ses performances. Contrairement à DMCI qui utilise l'ensemble de l'espace discret, cette méthode nommée RSP procède séquentiellement à la réduction de l'espace discret de recherche et de cibler les régions susceptibles de contenir la solution. Pour cette méthode, nous avons choisi d'utiliser seulement la représentation en virgule fixe, à cause de son pas uniforme. Cette méthode RSP a été testée en utilisant d'abord le critère de l'erreur de Chebyshev, ensuite le critère de l'erreur

quadratique moyenne. Les résultats obtenus sont satisfaisants. Le temps de calcul a été drastiquement réduit comparé à celui de la méthode RA. La limitation du domaine ne semble pas dégrader les performances de l'algorithme.

Références

[1] D.M. Kodek, '' Design of Optimal Finite Word length FIR Digital Filters Using Integer Programming Techniques,'' IEEE Trans. Acoust., Speech, Signal Processing, vol. 28, pp. 304-308, June 1980.

[2] Ioannis PITAS, '' Optimisation and Adaptation of Discrete-Valued Digital Filter Parameters by Simulated Annealing ,'' IEEE Trans. Signal Processing, vol. 42, NO. 4, pp. 860-866, April 1994.

[3] N. Benvenuto, M. Marchesi, ''Digital Filters Design by Simulated Annealing,'' IEEE Trans. Circuits Syst., vol. 36, pp. 459-460, March 1989.

[4] T. Ciloglu and Z. Unver, '' , '' A New Approach to Discrete Coefficient FIR Digital Filter Design by Simulated Annealing,'' IEEE of Int. Conf. on ASSP. Minnisota 1993.

[5] Lawrence R. Rabiner, Bernard Gold, '' Theory and Application of Digital Signal Processing,'' PRENTICE-HALL, INC. 1975.

[6] Alan V. Oppenheim, Rinald W. Schafer, '' Digital Signal Processing,'' PRENTICE-HALL INTERNATIONAL, 1975.

[7] T. Ciloglu and Y.H. Lee, '' Efficient Allocation of Power-of-Two Terms in Complex FIR Filter Design,'' EUSIPCO Proc. Trieste, Italy, September 1996.

[8] J. P. Vidal, ''Traitement numérique du signal,'' Cours polycopiés, INT Paris 1987.

[9] J. H. Mc Clellan, T. W. Parks, and L. R. Rabiner, '' A Computer Program for Designing Optimum FIR Linear Phase Digital Filters,'' IEEE Trans. Audio Electroacoust., vol. AU-21, pp. 506-526, Dec. 1973.

[10] M. Minoux, ''Programmation mathématique, théorie et algorithmes,'' tomes 1 et 2, Dunod, 1983.

[11] B. Jaumard, M. Minoux, and P. Siohan '' Finite Precision Design of FIR Digital Filters Using a Convexity Property,'' IEEE Trans. Acoust., Speech, Signal Processing, vol. 36, pp. 407-411, March 1988.

[12] Kurt Arbenz, et Alfred Wohlhauser, ''Analyse numérique,'' PRESSES POLYTECHNIQUES ROMANDES, 1980.

[13] Yong C. Lim, Sydney R. Parker, and A. G. Constantinides, '' Finite Word Length FIR Filter Design Using Integer Programming Over a Discrete Coefficient Space,'' IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-30, pp. 661-664, Aug. 1982.

[14] Y. C. Lim, S. R. Parker, ''FIR Filter Design Over a Discrete Powers-of-Two Coefficient Space,'' IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-31, pp. 583-591, June 1983.

[15] Y. C. Lim, S. R. Parker, '' Discrete Coefficient FIR Digital Filter Design Based Upon an LMS Criteria,'' IEEE Trans. Circuits Syst., vol. CAS-30, pp. 723-739, Oct. 1983.

[16] Y. C. Lim, '' Design of Discrete-Coefficient-Value Linear Phase FIR Filters with Optimum Normalised Peak Ripple Magnitude,'' IEEE Trans. Circuits Syst., vol. 37, NO. 12 pp. 1480-1486, December 1990.

[17] Quangfu Zhao and Yoshiaki Tadokoro, ''A Simple Design of FIR Filters with Powers-of-Two Coefficients,'' IEEE Trans. Circuits Syst., vol. CAS-35, NO.5, may 1988.

[18] T. Q. Nguyen, '' The Design of Arbitrary FIR Digital Filters Using the Eigenfilter Method,'' IEEE Trans. Signal Processing, vol. 41, pp. 1128-1139, March 1993.

[19] N. Benvenuto, L. E. Franks and F. S. Hill, '' On the Design of FIR Filters with Powers-of-Two Coefficients,'' IEEE Trans. On Communications, vol. COM-32, NO. 12, Dec. 1984.

[20] D.M. Kodek, ''Limits of Finite Word length FIR Digital Filter Design,'' Proc. ICASSP München, April 1997.

[21] R. S. Garfinkel & G. L. Nemhauser, ''Integer Programming '' New York: Wiley.

[22] M. Lang, '' Design of Nonlinear Phase FIR Digital Filters Using Quadratic Programming,'' Proc. ICASSP München, April 1997.

[23] J. G. Wade, '' Codage et traitement du signal,'' Masson, 1991.

[24] M. Kunt, et al.,''Techniques modernes de traitement numérique des signaux,'' vol.1, PRESSES POLYTECHNIQUES et UNIVERSITAIRES ROMANDES, 1991.

[25] M. Bellanger, ''Traitement numérique du signal, théorie et application,'' MASSON 1987.

[26] R. Boite , H.Leich, ''Les filtres numériques,'' MASSON, 1982.

[27] Murat Kunt, ''Traitement numérique des signaux,'' Vol. XX EDITIONS GEORGI, 1980.

[28] Li Lee & Alan V. Oppenheim, ''Properties of Approximate Parks-Mc Clellan Filters,'' Proc. ICASSP München, April 1997.

[29] Henri Samueli, ''A Improved Search Algorithm for the Design of Multiplierless FIR Filters with Powers-of-Two Coefficients,'' IEEE Trans. Circuits Syst., vol. 36, NO.7, July 1989.

[30] H. Shaffeu, M. M. Jones, H. D. Griffiths, J. T. Taylor, ''Improved Design Procedure for Multiplierless FIR Digital Filters,'' Electronics Letters, vol. 27, NO.13, June 1991.

[31] B. Boulerial, ''Synthèse de Filtres RIF à Phase Linéaire dans l'Espace Discret des Coefficients'', Mémoire de Magister, Institut des Télécommunications d'Oran, Algérie, novembre 1998.

[32] B. Boulerial, M. F. Belbachir, ''Une Méthode Directe au Sens des Moindres Carres D.M.C. pour la Conception de Filtres R.I.F. à Phase Linéaire dans l'Espace Discret des Coefficients,'' Proc. I.M.C.E.S.99, Sidi Bel Abbes, Algeria, April 1999.

[33] B. Boulerial, M. F. Belbachir, ''Filtres RIF : Synthese Directe dans l'Espace Discret des Coefficients,'' Proc. NWSIP'98, Sidi Bel Abbes, Algeria, December 1998.

[34] A.N. Belbachir, B. Boulerial, M. F. Belbachir, ''Synthèse des Filtres RIF,'' soumis et accepté au Séminaire National sur l'Automatique et les Signaux, SNAS'99, 9-10 Novembre, Annaba, 1999.

[35] A.N. Belbachir, B. Boulerial, M. F. Belbachir, ''Une Approche Itérative pour la Conception de Filtres Numériques RIF à Phase Linéaire dans l'Espace Discret des Coefficients,'' Conférence Maghrébine en Génie Electrique, CGME'99, 4-6 Décembre, Institut d'Electronique, Université Mentouri Constantine, 1999.

[36] B. Boulerial, A.N. Belbachir, M. F. Belbachir, ''A New Approach to Finite Wordlength Coefficient FIR Digital Filter Design Using the Branch and Bound Technique," EUSIPCO'00,'' 05-08 September, Tampere-Finland, 2000.

[37] A.N. Belbachir, M. F. Belbachir, A. Fanni, S. Bibbò, B. Boulerial ''A New Approach to Digital FIR Filter Design Using the Tabu Search," IEEE NORdic SIGnal Processing Symposium, NORSIG'00, 13-15 June, Kolmarden-Sweden, 2000.

[38] A.N. Belbachir, M. F. Belbachir, ‘’Rapport sur le Traitement de l’Information,” Rapport Interne, Dipartimento di Ingegneria Elettrica ed Elettronica, Universita’ degli Studi di Cagliari, Italia, October 1999.

[39] A.N. Belbachir, M. F. Belbachir, A. Fanni ‘A Sequential Robust Method to Finite Wordlength Coefficient FIR Digital Filter Design,” IEEE NORdic SIGnal Processing Symposium, NORSIG'00, 13-15 June, Kolmarden Sweden, 2000.

Annexe.

METHODE D'OPTIMISATION DIRECTE A MOINDRE CARRE 'D.M.C.'

La méthode DMC calcule séquentiellement les coefficients du filtre recherché. A part le premier coefficient $h(\frac{N}{2}-1)$, qui est déterminé en fonction des spécifications données du filtre, la recherche des autres coefficients suivants est effectuée en prenant en compte l'erreur due à la quantification des coefficients précédemment calculés.

a) calcul du coefficient $h(\frac{N}{2}-1)$

On calcule $h(\frac{N}{2}-1)$ tel que la distance entre la courbe correspondante à 2.h(N/2 - 1).cos(w/2) et le gabarit du filtre donné soit minimale.

Nous définissons par :

S_1 : La surface, dans la bande passante [0 w_p], limitée par le gabarit du filtre et la courbe d'amplitude correspondante à 2.h(N/2 -1).cos(w/2).

$$S_1 = \int_0^{wp} [1 - 2.h(N/2 - 1).\cos(w/2)].dw = w_p - 4.h(N/2 - 1).\sin(w_p/2). \quad (89)$$

et S_2 : La surface, dans la bande atténuée [w_c 0.5], limitée par le gabarit du filtre et la courbe d'amplitude correspondante à 2.h(N/2 -1).cos(w/2).

$$S_2 = \int_{wc}^{\pi} [0 - 2.h(N/2 - 1).\cos(w/2)].dw = 4.h(N/2 - 1).[\sin(w_c/2) - 1]. \quad (90)$$

Nous appellerons par Ω la quantité suivante :

$$\Omega = S_1^2 + S_2^2 = \{ w_p - 4.h(N/2 - 1).\sin(w_p/2).\}^2 + \{4.h(N/2 - 1).[\sin(w_c/2) - 1].\}^2 \quad (91)$$

Ω est l'erreur quadratique résiduelle.

Pour minimiser la distance entre la courbe correspondante à 2.h(N/2-1).cos(w/2) et le gabarit du filtre donné, on optimise Ω, pour cela il faut résoudre:

$$\partial\Omega/\partial\, h(\frac{N}{2}-1) = 0 . \quad (92)$$

La résolution de cette équation, nous donne:

$$h(\frac{N}{2}-1) = \frac{w_p \sin(\frac{w_p}{2})}{4\left\{\sin^2(\frac{w_p}{2}) + (\sin(\frac{w_c}{2}) - 1)^2\right\}} \quad (93)$$

La valeur du coefficient, ainsi déterminée par cette formule, est une valeur continue. La valeur discrète est obtenue en effectuant une quantification par arrondissement dans une des trois représentations.

$h_d(N/2 - 1) = \{ h(\frac{N}{2}-1) \}$ quantifié dans une des 3 représentations .

On remarque que la fonction Ω étant quadratique, la quantification par arrondi des coefficients ne donne pas toujours la bonne solution discrète. En remplacement de la quantification par arrondi, nous avons ajouté le test sur les deux valeurs discrètes adjacentes (inférieure et supérieure) à la valeur continue. La valeur discrète retenue, est celle qui donne la plus petite valeur d'erreur quadratique moyenne résiduelle.

La détermination analytique des autres coefficients est difficile à mettre en équation, les coefficients sont alors calculés numériquement.

b) Calcul des autres coefficients

L'erreur résiduelle Err_0 entre la courbe correspondante à $2.h(N/2 -1).\cos(w/2)$ et le gabarit du filtre, due à l'approximation faite avec un seul terme de la somme $\{h(\frac{N}{2}-1)$ pour le filtre de cas 2$\}$, ne peut qu'augmenter en faisant la quantification du coefficient.

A une étape n, le coefficient suivant $h(\frac{N}{2}-n-1)$ doit donc être déterminé en tenant compte de deux erreurs qui sont dues à:

a) La somme finie de termes.

b) La quantification du coefficient h(N/2 - n), calculé à l étape précédente n-1.

Pour optimiser une grandeur, nous disposons de deux méthodes (la méthode du simplexe, et la méthode par les moindres carrés). Le choix de l'optimisation des coefficients par la méthode des moindres carrés (M.C.) est fait sur la base de la rapidité: M.C. étant non itérative et donc est plus rapide que la méthode du simplexe.

Pour le calcul du coefficient $h(\frac{N}{2}-n-1)$ à l'étape n (n = 1 .. N/2), on forme tout d'abord le vecteur $\mathbf{E_n}$, en calculant ses éléments par l'équation :

$$E_n(w_i) = (Err_{n-1} + Erd_{n-1})(w_i) = E_{n-1}(w_i) - 2\, h_d(N/2-n)\cos(n-1/2)w_i \quad (94)$$

où $\mathbf{E_n}$: erreur d'amplitude résiduelle effective obtenue à la fin de l'étape n-1

$\mathbf{Err_{n-1}}$: erreur d'amplitude résiduelle avant quantification du coefficient

$h(\frac{N}{2}-n)$. à l'étape n.-1

$\mathbf{Erd_{n-1}}$: erreur d'amplitude de quantification du coefficient $h(\frac{N}{2}-n)$.

Les fréquences w_i sont prises à des intervalles sensiblement égaux, à l'intérieur des deux bandes du filtre.

$h(\frac{N}{2}-n-1)$ est ensuite déterminé par optimisation (dans le sens des moindres carrés), en minimisant l'écart entre l'erreur résiduelle $\mathbf{E_n}$ et la courbe correspondante à $2.\, h(\frac{N}{2}-n-1).\cos((n+1/2)w)$

En admettant des erreurs Err_{ni}, on obtient, en notation matricielle, le système d'équations suivant:

$$\mathbf{h}.\mathbf{A} = \mathbf{E_n} + \mathbf{Err_n} \quad (95)$$

où

$$\mathbf{A} = 2\begin{bmatrix} \cos(n+\frac{1}{2})w_1 \\ \cos(n+\frac{1}{2})w_2 \\ \\ \cos(n+\frac{1}{2})w_{4N} \end{bmatrix} \quad ; \quad \mathbf{E_n} = \begin{bmatrix} E_n(w_1) \\ E_n(w_2) \\ \\ E_n(w_{4N}) \end{bmatrix} ; \quad \mathbf{Err_n} = \begin{bmatrix} Err_{n1} \\ Err_{n2} \\ \\ Err_{n4N} \end{bmatrix}$$

Le vecteur des inconnues **h** ne contient qu'un seul élément $h(\frac{N}{2}-n-1)$.

La somme Ω des carrés des erreurs $\mathbf{Err_n}$ est égale à :

$$\Omega = \mathbf{Err_n}^{\mathbf{T}}.\mathbf{Err_n} = (\mathbf{h}.\mathbf{A} - \mathbf{E_n})^{\mathbf{T}}.(\mathbf{h}.\mathbf{A} - \mathbf{E_n}) \quad (96)$$

la minimisation de Ω par rapport à $h(\frac{N}{2}-n-1)$ est obtenue par:

$$\partial\Omega/\partial\, h(\frac{N}{2}-n-1) = 0. \quad (97)$$

on déduit que $h(\frac{N}{2}-n-1) = (\mathbf{A}^{\mathbf{T}}.\mathbf{A})^{-1}.\mathbf{A}^{\mathbf{T}}.\mathbf{E_n} = \mathbf{A}^{\mathbf{T}}.\mathbf{E_n} / \mathbf{A}^{\mathbf{T}}.\mathbf{A}$ (98.a)

ou $$h(\frac{N}{2}-n-1) = \frac{\left\{E_n(w_1)\cos(n+\frac{1}{2})w_1 + ... + E_n(w_{4N})\cos(n+\frac{1}{2})w_{4N}\right\}}{2\left\{\cos^2(n+\frac{1}{2})w_1 + ... + \cos^2(n+\frac{1}{2})w_{4N}\right\}} \quad (98.b)$$

Le coefficient ainsi déterminé en continu, est ensuite quantifié à sa plus proche valeur dans la représentation binaire choisie.

$h_d(\frac{N}{2}-n-1) = \{\ h(\frac{N}{2}-n-1)\ \}$ quantifié.

ARTICLES

Article N° 1
“Une Approche Itérative pour la Conception de Filtres Numériques RIF à Phase Linéaire dans l'Espace Discret des Coefficients”.

Publié dans:
Conférence Maghrébine en Génie Electrique, CMGE'99, Constantine, Algeria, December 1999.

UNE APPROCHE ITERATIVE POUR LA CONCEPTION DE FILTRE NUMERIQUE R.I.F. A PHASE LINEAIRE DANS L'ESPACE DISCRET DES COEFFICIENTS

A.N. Belbachir, B. Boulerial et M. F. Belbachir

Laboratoire Signaux et Systèmes, Institut d'Electronique,
Université Mohammed BOUDIAF 'U.S.T.O.' Oran, B.P. 1505 El Mnaouer 31000 Oran, Algérie
E-mail : belbachiran@yahoo.com.

Résumé :

La méthode de recherche arborescente est optimale pour la conception des filtres numériques à coefficients de longueur limitée. Le problème est le temps de calcul prohibitif nécessaire pour la conception des filtres de longueur supérieure à 8 avec une représentation des coefficients sur 8 bits. A ce propos, une nouvelle technique ''méthode d'optimisation directe par moindre carré itérative'' est présentée afin d'approcher les résultats obtenus par la méthode de recherche arborescente dans le sens de l'erreur quadratique moyenne avec un temps de calcul plus petit.

Mots clés :

Méthode de Recherche Arborescente (R.A.), Méthode Directe par Moindre Carré (D.M.C.), Méthode Directe par Moindre Carré Itératives (D.M.C.I.), Optimum Local (O.L.), Critère d'Erreur Quadratique Moyenne 'E.Q.M.'.

1. Introduction

Lorsqu'un filtre numérique est implanté sur un processeur de signaux de longueur de mot 'lm' bits, chaque coefficient du filtre doit être représenté, par un nombre fini 'lm' de bits. L'approche habituellement utilisée consiste à quantifier les coefficients. Ceci engendre une détérioration importante des caractéristiques du filtre. On montre d'ailleurs qu'il existe d'autres coefficients de même longueur de mot finie qui donnent une meilleure approximation. Afin de retrouver ces coefficients, nous proposons donc d'inclure la limitation de la longueur de mot dans la procédure de la conception de filtre.

Pour concevoir des filtres numériques à coefficients de longueur finie, il est souvent désirable d'utiliser des algorithmes dont la qualité de sortie peut être ajustée suivant la disponibilité des ressources telles que, la précision et le temps de calcul. Dans ce cas le problème devient plus complexe, où une investigation générale de la solution optimale nécessite un temps de calcul prohibitif. Afin de remédier à ce problème, plusieurs méthodes d'optimisation ont été élaborées pour la conception des filtres numériques à coefficients discrets. La technique du gradient simulée 'Simulated Annealing Technique' (S.A.) [2]-[4] est efficace dans plusieurs cas, mais nécessite un grand nombre de fonctions d'évaluations impliquant un coût de calcul élevé. Ce nombre de fonctions d'évaluations dépend des températures de départ.

La programmation linéaire en nombre entier a été appliquée dans [1], [7]-[9] comme méthode d'optimisation discrète dans le sens minmax. Bien qu'elle permet d'obtenir des résultats optimaux, le temps de calcul nécessaire même avec les super ordinateurs actuels, prohibe l'application de ces techniques pour des filtres d'ordre élevé.

Les techniques d'optimisation dans l'espace discret des coefficients, et en particulier la méthode de recherche arborescente, ont été élaborées afin de remédier à ce problème d'optimum discret. Ces méthodes basées sur des techniques d'énumérations implicites nécessitent aussi un coût de calcul très élevé [8], [10], [14], et [15].

Dans plusieurs méthodes de recherche locale basées sur la méthode de recherche arborescente, tels que 'la méthode de recherche en profondeur d'abord' ou 'la méthode de séparation et d'évaluation progressive' (SEP), les solutions retrouvées sont meilleures que celles obtenues à partir d'une quantification directe. L'objectif majeur de ces méthodes est la détermination de stratégies de branchement et de syntonisation. Cependant, la solution optimale n'est pas assurée [10], [14], [15].

La convergence de la méthode d'optimisation directe par moindre carré 'D.M.C.' proposée dans [15], [16] est très rapide, tandis que la solution finale dépend du choix du premier coefficient à déterminer ainsi que de l'ordre dans lequel sont ensuite considérés les autres coefficients.

Dans cet article, nous présentons une technique nommée 'D.M.C.I.' permettant d'exploiter l'optimalité de la méthode de recherche arborescente 'R.A.' et la vitesse de convergence de la méthode D.M.C.

Dans la section 2, nous définissons le problème et les caractéristiques du critère d'erreur quadratique moyenne. Dans la section 3, la méthode directe à moindre carré itérative 'DMCI' est décrite et les formulations mathématiques sont données. Une étude comparative des résultats que nous avons obtenus avec ceux obtenus dans [15] et [16] seront donnés dans la section 4.

2. Position du Problème

Soit à concevoir un filtre numérique RIF à phase linéaire de longueur N dont la réponse en fréquence s'écrit sous la forme :

$$H(e^{j\omega}) = \sum_{k=0}^{N-1} h_k e^{-j\omega k} \quad (1)$$

Il a été montré dans [5], que l'amplitude de la réponse en fréquence, pour les quatre cas de

filtre à phase linéaire, peut s'écrire sous la forme :

$$Pn(\omega) = \sum_{k=0}^{n-1} ak\cos\omega k \qquad (2)$$

où n est le nombre de termes :

$$n = (N/2 \text{ ou } (N-1)/2 \text{ ou } (N+1)/2) \qquad (3)$$

et ak, relative à hk, est la séquence décalée dépendant du cas considéré.
La réponse en amplitude Pn(ω) est comparée avec l'amplitude de la réponse en fréquence idéale D(ω), à l'aide du critère de l'erreur quadratique moyenne 'EQM'.
L'erreur d'approximation en amplitude suivant le critère de l'erreur quadratique moyenne est donnée par la forme suivante :

$$enam = \frac{1}{Nw}\sum_{i=0}^{Nw-1}\left\| D(w_i) - P_n(w_i)\right\| \qquad (4)$$

où i = 1, 2, .., Nw-1
Nw : nombre de points fréquentiels sélectionnés.

Si on considère un filtre passe bas idéal, l'amplitude de sa réponse en fréquence s'écrit comme suit :

D(ωi) = 1 pour ωi ∈ bande passante.
D(ωi) = 0 pour ωi ∈ bande atténuée.

L'expression (4) devient alors :

$$enam = \frac{1}{Nw}\left\{ \sum_{i=0}^{k-1}\left\|1 - P_n(w_i)\right\| + \sum_{i=k}^{Nw-1}\left|P_n(w_i)\right|\right\} \qquad (5)$$

Les coefficients des filtres sont restreints à des valeurs discrètes définies par la longueur du mot machine disponible à 'lm' bits.
Les méthodes que nous présentons dans ce qui suit sont valables pour les quatre cas décrits précédemment. Nous nous limitons au cas 2 (N pair et la symétrie positive). L'extension aux autres cas peut se faire facilement.

3. Méthode Directe à Moindre Carré Itérative 'D.M.C.I.'

Il a été montré dans [15], et [16] que la méthode 'D.M.C.' est plus rapide que la méthode 'R.A.'. La méthode détermine par un calcul séquentiel les valeurs des coefficients du filtre dans un ordre bien déterminé. Il a été aussi montré que les performances des filtres conçus par cette méthode 'D.M.C.' ne garantit pas l'optimalité des filtres conçus, la solution obtenue peut être un optimum local 'O.L.'.
Notre propos dans cet article est de présenter une nouvelle approche qui exploite la rapidité de convergence de la méthode 'D.M.C.' et permet de concevoir des filtres approchant les performances de ceux obtenus par la 'R.A.'.
Notre objectif est de palier aux deux inconvénients suivants :

- Le problème de la méthode 'D.M.C.' est le choix du coefficient de départ sur les performances du filtre obtenu ainsi que l'ordre dans lequel sont calculés les coefficients.

- Le problème lié à la méthode 'R.A.' est le temps de calcul prohibitif lorsque la longueur du filtre est supérieure à 8 avec une longueur de mot machine supérieure à 8 bits [15]. Ceci explique que dans la section 4., on ne trouvera qu'un seul filtre conçu par la méthode R.A. de longueur N=8 et lm=7 bits pour l'étude comparative.

A ce propos, nous avons élaboré la méthode Directe par Moindre Carré Itérative 'D.M.C.I.' qui permet d'améliorer la précision des résultats de la méthode 'D.M.C.' en effectuant en plusieurs itérations une recherche exhaustive autour de la solution obtenue au moyen de la méthode 'D.M.C.'.

3.1. Description de la Méthode :

Pour expliquer la méthode, nous choisissons l'exemple simple suivant :

Considérons un filtre défini par deux coefficients {h(0), h(1)}, dans un espace discret de longueur de mot 'lm' bits, avec 'va' valeurs admissibles.
La méthode 'D.M.C.I.' se déroule comme suit :

Nous utilisons la méthode 'D.M.C.' [16] pour calculer les coefficients hdmc(0) et hdmc(1). L'erreur 'E.Q.M.' du filtre calculé par rapport au filtre idéal est notée par Edmc.

Posons Er=Edmc.

Puis, nous fixons le coefficient hdmc(0) et nous varions le coefficient hdmc(1) dans un sous espace de valeurs

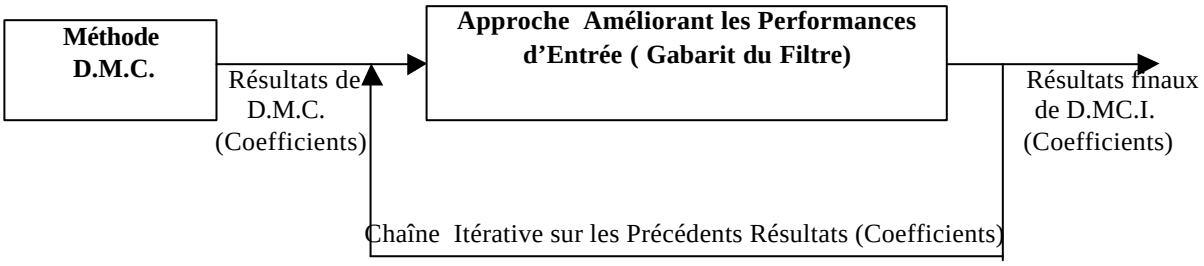


Fig.1. Schéma Représentatif de la Méthode D.M.C.I.

discrètes de rayon 'v' centré autour du coefficient hdmc(1). ('v' est le nombre de valeurs admissibles inférieur à 'va').
Par ailleurs, à chaque solution discrète se situant dans l'espace de rayon 'v', nous calculons l'erreur 'E.Q.M.' notée par Edmci.
Si Edmci < Er, nous sauvegardons les coefficients correspondants soit {hdmc(0), hr(1)} et Er devient égale à Edmci.

Posons Er = Edmci.

De la même façon précédente, nous fixons cette fois-ci le coefficient hr(1) et nous varions le coefficient hdmc(0) dans un sous espace de valeurs discretes de rayon 'v' centré autour du coefficient hdmc(0). ('v' est le nombre de valeurs admissibles inférieur à 'va'). A chaque solution discrète, nous calculons l'erreur 'E.Q.M.' Edmci.
Si Edmci < Er, nous sauvegardons les coefficients correspondants soit {hr(0), hr(1)} et Er devient égale à Edmci.
Cette recherche est refaite sur plusieurs itérations 'It', jusqu'à ce que la variation de l'erreur ΔEr soit nulle.

Où
ΔEr=Er(itération actuelle)–Er(itération précédente). (6)

Dans ce cas nous obtenons les coefficients finaux hf(0) et hf(1). Ces coefficients sont nommés {hdmci(0), hdmci(1)} obtenus par la méthode 'D.M.C.I.' après 'It' itérations.

3.2. Description de l'algorithme:

L'algorithme de cette méthode 'D.M.C.I.' se subdivise en deux parties (Fig.1) :
La première partie représente l'algorithme de la méthode 'D.M.C.' définie dans [15], [16], où les valeurs des coefficients sont obtenues par un calcul séquentiel.
La deuxième partie représente une méthode itérative qui est basée sur l'amélioration des résultats trouvés dans la première partie (méthode DMC) après 'It' itérations, en faisant le balayage des valeurs discrètes dans un espace de rayon 'v' centré autour du coefficient de départ (coefficient calculé par 'D.M.C.').

'v' étant fixé suivant le nombre de valeurs admissibles 'va' et la longueur du mot du processeur 'lm'.

4. Résultats des Expériences

L'algorithme a été testé en utilisant des cas rapportés dans la littérature. Dans tous les exemples, le calculateur utilisé est Microsoft Intel Pentium II avec une fréquence CPU de travail de 300 MHZ. Les résultats obtenus

sont présentés et comparés aux algorithmes dans [15] et [16]. Le numéro de référence indique où le filtre a été pris. Un filtre de longueur 8 et une quantification à 7 bits, excluant le bit de signe, est noté par '8/7'. L'algorithme a été testé dans les trois représentations binaires en virgule fixe, en virgule flottante et S.D.P.D. (Somme de Deux de Puissance de Deux) [10] respectivement dans les tableaux 1, 2, 3.

Nous notons par :

N : longueur du filtre.

lm : longueur du mot du processeur en bits.

PMCQ : 'EQM' relative au filtre de Parks Mc. Clellan et où les coefficients sont quantifiés dans une longueur de mot de 'lm' bits.

RA : 'EQM' relative au filtre calculé au moyen de la méthode 'R.A.'.

DMC : 'EQM' relative au filtre calculé au moyen de la méthode 'D.M.C.'.

DMCI : 'EQM' relative au filtre calculé au moyen de la méthode 'DMCI'.

tp : temps de calcul correspondant à chaque méthode.

N/lm	PMCQ/tp	RA/tp	DMC/tp	DMCI/tp
8/7	.035/.05s	.033/46h	.059/.05s	.035/1.9s
8/19	.035/25s	----------	.048/6.2s	.031/60s
24/15	.0010/13s	----------	.0099/4s	.0008/85s
16/15	.070/9s	----------	.043/2s	.039/45s
50/15	.002/20s	----------	.007/8s	.001/117s
56/13	.004/19s	----------	.007/7.2s	.003/123s

Tableau 1. Comparaison des résultats obtenus par les méthodes 'P.M.C.Q.', 'R.A.'[15], 'D.M.C.'[16] et 'D.M.C.I.' dans la représentation en virgule fixe.

N/lm	PMCQ/tp	RA/tp	DMC/tp	DMCI/tp
8/7	.035/.05s	.032/49h	.059/.05s	.032/1.8s
8/19	.035/25s	----------	.031/5.6s	.031/51s
24/15	.0010/10s	----------	.0099/4s	.0009/80s
16/15	.070/7.5s	----------	.043/1.8s	.039/60s
50/15	.002/17s	----------	.007/9s	.001/130s
56/13	.004/18s	----------	.007/5s	.003/139s

Tableau 2. Comparaison des résultats obtenus par les méthodes 'P.M.C.Q.', 'R.A.'[15], 'D.M.C.'[16] et **'D.M.C.I.' dans la représentation en virgule** flottante.

N/lm	PMCQ/tp	RA/tp	DMC/tp	DMCI/tp
8/7	.127/.05s	.127/78s	.148/.05s	.127/.88s
8/19	.070/17s	----------	.064/3.9s	.051/41s
24/15	.065/2.2s	----------	.053/1.3s	.016/65s
16/15	.078/1.1s	----------	.063/.9s	.040/8s
50/15	.023/4s	----------	.020/2.5s	.013/96s
56/13	.025/8s	----------	.028/4.2s	.017/95s

Tableau 3. Comparaison des résultats obtenus par les méthodes 'P.M.C.Q.', 'R.A.'[15], 'D.M.C.'[16] et 'D.M.C.I.' dans la représentation S.D.P.D.

Dans tous les exemples, les filtres considérés sont des filtres R.I.F. à phase linéaire appartenants au cas2 (symétrie positive et longueur du filtre paire) [6]. Nous présentons dans ces trois tableaux les résultats de six filtres. Les trois premiers exemples sont des filtres de bande passante [0 0.159] et de bande atténuée [0.295 0.5]. Les trois autres exemples sont de bandes passantes [0 0.318], [0 0.05], [0 0.31] et de bandes atténuées respectivement [0.371 0.5], [0.104 0.5], [0.35 0.5]. Tous les filtres sont conçus avec une fonction poids égale dans la bande passante et dans la bande atténuée. Autour de chaque coefficient calculé par la méthode DMC, nous avons défini un sous espace discret de recherche de rayon 'v' expérimentalement fixé en référence à la longueur du mot machine 'lm'. Le nombre d'itérations 'It' dépend de l'ordre du filtre. Dans la colonne de la méthode R.A. nous n'avons représenté que le premier filtre parce que cette méthode ne peut pas être utilisée pour la conception des filtres d'ordre supérieur à 7 et de longueur de mot supérieure à 7 bits excluant le bit de signe, à cause du temps de calcul prohibitif.

Le nombre d'itérations 'It' varie de 5 à 40 suivant l'ordre du filtre, tandis que le rayon 'v' du sous espace discret varie entre 5 à 20 suivant la longueur du mot 'lm'.

Dans tous les exemples les résultats obtenus sont meilleurs que ceux obtenus dans la référence indiquée. Dans les tableaux 1, 2 et 3 les résultats sont donnés en fonction de l'erreur et du temps de calcul. Il est remarqué que l'algorithme de la méthode D.M.C.I. peut être utilisé pour le calcul de filtres de grande longueur avec un temps de calcul accessible (N=56 pour exemple 6) et peuvent approximer des

filtres dans l'espace discret des coefficients à longueur de mot grande (lm=20 bits pour l'exemple 2).

5. Conclusion

Dans cet article, nous avons proposé l'algorithme nommé 'D.M.C.I.' qui utilise une nouvelle approche qui consiste à exploiter les performances de 'R.A.' et la rapidité de 'D.M.C.'. A travers quelques exemples nous avons montré que les objectifs recherchés ont été atteints. A titre d'exemple, pour la conception d'un filtre de longueur N=8, le filtre obtenu par 'D.M.C.I.' possède les performances du filtre 'R.A.' (E.Q.M.=.032 ou .127 selon la représentation utilisée) et un temps de calcul très acceptable. L'étude comparative basée sur les critères E.Q.M./tp que nous avons donné dans cet article montre que les résultats obtenus peuvent être considérés comme meilleurs que ceux de [15], [16]. Dans les exemples, il est constaté que les représentations en virgule fixe et en virgule flottante sont mieux adaptées à D.M.C.I. La méthode D.M.C.I. est plus lente que la méthode D.M.C. avec un temps de synthèse très acceptable mais nettement plus rapide que R.A.
Notre recherche actuelle consiste à améliorer la complexité de cet algorithme et cela en faisant un choix approprié du nombre nécessaire d'itérations.

Références

[1] D.M. Kodek, ''Design of Optimal Finite Word length FIR Digital Filters Using Integer Programming Techniques,'' IEEE Trans. Acoust., Speech, Signal Processing, vol. 28, pp. 304-308, June 1980.

[2] Ioannis PITAS, ''Optimisation and Adaptation of Discrete-Valued Digital Filter Parameters by Simulated Annealing,'' IEEE Trans. Signal Processing, vol. 42, NO. 4, pp. 860-866, April 1994.

[3] N. Benvenuto, M. Marchesi, ''Digital Filters Design by Simulated Annealing,'' IEEE Trans. Circuits Syst., vol. 36, pp. 459-460, March 1989.

[4] T. Ciloglu and Z. Unver, ''A New Approach to Discrete Coefficient FIR Digital Filter Design by Simulated Annealing,'' IEEE of Int. Conf. on ASSP. Minnisota 1993.

[5] Lawrence R. Rabiner, Bernard Gold, ''Theory and Application of Digital Signal Processing,'' PRENTICE-HALL, INC. 1975.

[6] J. H. Mc. Clellan, T. W. Parks, and L. R. Rabiner, ''A Computer Program for Designing Optimum FIR Linear Phase Digital Filters,'' IEEE Trans. Audio Electroacoust., vol. AU-21, pp. 506-526, Dec. 1973.

[7] M. Minoux, ''Programmation mathématique, théorie et algorithmes,'' tomes 1 et 2, Dunod, 1983.

[8] B. Jaumard, M. Minoux, and P. Siohan ''Finite Precision Design of FIR Digital Filters Using a Convexity Property,'' IEEE Trans. Acoust., Speech, Signal Processing, vol. 36, pp. 407-411, March 1988.

[9] Yong C. Lim, Sydney R. Parker, and A. G. Constantinides, ''Finite Word Length FIR Filter Design Using Integer Programming Over a Discrete Coefficient Space,'' IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-30, pp. 661-664, Aug. 1982.

[10] Y. C. Lim, S. R. Parker, ''FIR Filter Design Over a Discrete Powers-of-Two Coefficient Space,'' IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-31, pp. 583-591, June 1983.

[11] Y. C. Lim, S. R. Parker, ''Discrete Coefficient FIR Digital Filter Design Based Upon an LMS Criteria,'' IEEE Trans. Circuits Syst., vol. CAS-30, pp. 723-739, Oct. 1983.

[12] Y. C. Lim, ''Design of Discrete-Coefficient-Value Linear Phase FIR Filters with Optimum Normalised Peak Ripple Magnitude,'' IEEE Trans. Circuits Syst., vol. 37, NO. 12 pp. 1480-1486, December 1990.

[13] Li Lee & Alan V. Oppenheim, ''Properties of Approximate Parks Mc. Clellan Filters,'' Proc. ICASSP München, April 1997.

[14] B. Boulerial, M. F. Bel Bachir, ''Filtres RIF: Synthèse Directe dans l'Espace Discret des coefficients'', 1st National Workshop on Signal & Image Processing, NWSIP'98, 1 Décembre 1998, Sidi Bel-Abbes, Algérie.

[15] B. Boulerial, ''Synthèse de Filtres RIF à Phase Linéaire dans l'Espace Discret des Coefficients'', Thèse de Magister, Institut des Télécommunications d'Oran, Algérie, novembre 1998.

[16] B. Boulerial, M. F. Bel Bachir, ''Une Méthode Directe au Sens des Moindres Carrés D.M.C pour la Conception de Filtres R.I.F à Phase Linéaire dans l'Espace Discret des coefficients'', International Meeting on Components and Electronic Systems IMCES'99, 17-18 mai 1999, Sidi Bel-Abbes, Algérie.

Article N° 2

“A Sequential Robust Method to Finite Wordlength Coefficient FIR Digital Filter Design”.

Publiée dans:

IEEE NORdic SIGnal Processing Symposium, NORSIG'00, KolmardenSweden, June 2000.

A SEQUENTIAL ROBUST METHOD TO FINITE WORDLENGTH COEFFICIENT FIR DIGITAL FILTER DESIGN

A.N. Belbachir[1,2], B. Boulerial[1] and M.F. Belbachir[1]

[1]Signal and System Laboratory, Electronic Institut U.S.T.O.
XXIII. B.P. 1505 El Mnouer Oran - ALGERIA
[2]Vienna University of Technology, Pattern Recognition and Image Processing Group
Favoritenstr. 9/1832, A-1040 Vienna – AUSTRIA, nabil@prip.tuwien.ac.at

ABSTRACT

This paper describes a novel branching strategy using the branch and bound technique for the design of finite word length optimal digital filters. Using an extrapolated solution, the presented technique offers good performances in a low algorithmic complexity for large processor wordlength. It also provides a large flexibility to different error criterion. The details of the algorithm and many examples are given and compared to the other methods.

I. INTRODUCTION

One of the most known optimisation methods in the coefficients discrete space is the Branch and Bound Technique (BBT). This was used to solve the discrete optimum problem. This method based on implicit enumeration techniques also requires an expensive computing cost [7,8,10,15,16]. Many exhaustive approaches are performed on the (BBT) basis. The Depth First Search (DFS) and Breath First Search (BFS) are the famous approaches. Their major issue of concern is the strategy of branching. (DFS) approach performs its search in the whole available discrete space. Therefore, the obtained solution is optimal. However, before optimisation, (BFS) performs an estimation of the discrete space susceptible to contain the optimal solution. Hence, the performance are not guaranteed. Furthermore, the computing cost is prohibitive in a large processor wordlength.

The rounding of infinite precision coefficients for linear phase FIR digital filters design [6] is the most widely used technique when low algorithmic complexity is required. However, the frequency response amplitude of the obtained filter do not usually fulfill the passband and/or the stopband frequency tolerances.

For discrete optimisation, it is often desirable to use algorithms whose output quality can be adjusted depending on the availability of resources such as computing time and precision. To solve this problem, many optimisation methods has been applied to discrete coefficients FIR digital filters design. Simulated annealing technique [2,3,4] has proven to be effective in many cases, but requires a large number of function evaluations and does not guarantee the optimal solution. Furthermore, this method is based on heuristic assumptions, which may lead to bad performance.

The linear integer programming formulation [1],[7]-[9] was applied as a discrete optimisation method on the minmax criterion. Although, it is possible to obtain an optimum result, the computing time required even with the high-speed supercomputer of today, prohibits the application of these techniques for high order filters.

In this paper, we present a novel concept to design finite wordlength coefficients filter. The performed method named Sequential and Progressive Search (SPS) is based on a robust branching strategy. Its aim is to reduce the discrete space using an extrapolation method and perform the optimisation in the same time. Compared to the Depth First Search method (DFS), the number of function evaluations is smaller and it depends on the filter length without degrading the performance of the algorithm [14]. It will be shown (SPS) and (DFS) algorithmic complexity related to the processor wordlength.

In section II, we present the problem statements and the characteristics of the error criterion chosen. In section III, (SPS) method description is given. The algorithm modules are depicted in section IV. The results reported on section V deal with conventional minmax optimisation of FIR digital filters. A comparison of the algorithm performance with other reference algorithm is also given.

II. PROBLEM STATEMENTS

The frequency response of N-1 order linear phase FIR digital filter is usually written as

$$H(f) = \sum_{k=0}^{N-1} h_k e^{-j2\pi fk} \qquad (1)$$

In [5], It was shown that the frequency response amplitude of the four cases of linear phase (FIR) filters could be written in the form

$$P_n(f) = \sum_{k=0}^{n-1} a_k \cos 2\pi fk \qquad (2)$$

where the number of terms, n, is:

$$n(= N/2 \text{ or } (N-1)/2 \text{ or } (N+1)/2)$$

and a_k is the resulting shifted sequence depending on the considered case. The function $P_n(f)$ is compared with a desired frequency response amplitude D(f) using a minmax criterion. The weighted approximation error e_n is given by

$$e_n = \min_{(coeff.a)} \max_{f \in F} W(f)\left|D(f) - P_n(f)\right| \qquad (3)$$

- F : the disjoint union of all the frequency bands of interest.
- W(f): a weighting function defined on F.
- D(f) : the desired frequency response amplitude.

Using Eq. (2) in Eq. (3) gives

$$e_n = \min_{(coeff.a)} \max_{f \in F} W(f)\left|D(f) - \sum_{k=0}^{n-1} a_k \cos 2pfk\right| \qquad (4)$$

The filter coefficients are restricted to the discrete values allowed by (bwl) bit binary word length.

III. OPTIMIZATION METHOD

'SEQUENTIAL AND PROGRESSIVE SEARCH' (SPS)

Let us consider $\{a_k, k=0,1,...., N/2\}$ the filter discrete coefficients designed with an extra constraint imposing a limit on the processor wordlength.

Using the fixed point representation, we can express the discrete coefficient a_k as a linear combination:

$$|a_k| = \sum_{j=1}^{bwl-1} y_{j,k} 2^{-j} \quad k = 0, 1,...., N/2. \qquad (5)$$

Where (bwl) is the binary bit allowed for the filter discrete design and 'j' is the binary bit indication. $Y_{j,k}$ is a bivalent variable only fixed to only take the values '0' or '1'. Hence, the frequency response amplitude $P_n(f)$ could be expressed as

$$P_n(f) = \sum_{k=0}^{n-1} s\left(\sum_{j=1}^{bwl-1} y_{j,k} 2^{-j}\right) \cos 2\pi fk. \qquad (6)$$

where s is the sign of 'a_k' , s (= -1 or +1).

It is not possible to find the optimal filter coefficients at 'bwl' bits wordlength processor using the DFS method, owing to the long computing time required. But we can calculate these filter coefficients in a lower wordlength, (bbN) binary bit (bbN ≤ bwl) with such a method. a_k could be expressed as

$$|a_{k(opt)}| = \sum_{j=1}^{bbN-1} y_{j,k(opt)} 2^{-j} \quad k = 0, 1,...., N/2. \qquad (7)$$

$a_{k(opt)}$: the coefficients related to the optimal digital filter at (bbN) wordlength.

$Y_{j,k(opt)}$: 'j' binary bit value of the $a_{k(opt)}$ coefficient.

The proposed method, named Sequential and Progressive Search (SPS) takes this optimal solution as a starting point (starting solution) for its branching strategy in order to design filters with a higher wordlength discrete coefficients. Using the (DFS) optimal solution in the minmax sense in the bbN binary bits wordlength, we calculate with the SPS algorithm the solution at bwl bits (bbN ≤ bwl).

First, the coefficients are found at the (bbN+1) binary bits on the minimax sense using a local investigation in a reduced discrete search space. This reduced space is defined using the previous solution. Then we increase gradually the wordlength and calculate the new solution till reaching the (bwl) (the required word length). For each step 'i', we define a lower bounding function $e_{n\,i}$, which can be written as

$$e_{n\,i}(a) = \min_{(coeff.a)} \max_{f \in F} W(f) \left| D(f) - \sum_{k=0}^{n-1} a_k \cos 2\pi fk \right| \qquad (8)$$

$e_{n\,i}(a)$ is defined as the lowest value of the error function for a_k solving the following program :

$$a_k^{bbN+1} \le a_k^{bbN} + \mu \quad k=1,..., N/2 \qquad (9a)$$

$$a_k^{bbN+1} \ge a_k^{bbN} - \mu \quad k=1,..., N/2 \qquad (9b)$$

a_k^{bbN} : is the coefficients a_k represented by bbN bits

μ: interval chosen to contain the solution.

The implementation of a continuous value in two different wordlengths of the fixed-point representation does not offer two equal discrete values. The maximum difference between these discrete values is the quantization error '±LSB' (least sided bit) due to both truncation (±LSB) and rounding (±1/2.LSB). Hence, we have overestimate the μ value between the bbN and bwl wordlengths to

$$\mu = LSB = 2^{-(bbN-1)}. \qquad (10)$$

Substituting Eq. (10) and Eq. (7) in Eq. (9)

$$\sum_{j=1}^{bbN} y_{j,k} 2^{-j} \le \sum_{j=1}^{bbN-1} y_{j,k(opt)} 2^{-j} + 2^{-(bbN-1)} \qquad (11a)$$

$$\sum_{j=1}^{bbN} y_{j,k} 2^{-j} \ge \sum_{j=1}^{bbN-1} y_{j,k(opt)} 2^{-j} - 2^{-(bbN-1)} \qquad (11b)$$

Developing Eq. (11) we obtain

$$\sum_{j=1}^{bbN-2} y_{j,k(opt)} 2^{-j} + \sum_{j=bbN-1}^{bbN} y_{j,k} 2^{-j} \le \sum_{j=1}^{bbN-2} y_{j,k(opt)} 2^{-j} + y_{bbN-1,k(opt)} 2^{-bbN} + 2^{-(bbN-1)}. \qquad (12a)$$

$$\sum_{j=1}^{bbN-2} y_{j,k(opt)} 2^{-j} + \sum_{j=bbN-1}^{bbN} y_{j,k} 2^{-j} \ge \sum_{j=1}^{bbN-2} y_{j,k(opt)} 2^{-j} + y_{bbN-1,k(opt)} 2^{-bbN} - 2^{-(bbN-1)}. \qquad (12b)$$

After simplifications, we have

$$y_{bbN-1,k} \cdot 2^{-bbN+1} + y_{bbN,k} \cdot 2^{-bbN} \le y_{bbN-1,k(opt)} \cdot 2^{-bbN} + 2^{-(bbN-1)} \qquad (13a)$$

$$y_{bbN-1,k} \cdot 2^{-bbN+1} + y_{bbN,k} \cdot 2^{-bbN} \ge y_{bbN-1,k(opt)} \cdot 2^{-bbN} - 2^{-(bbN-1)} \qquad (13b)$$

Hence,

$$y_{bbN-1,k} + y_{bbN,k} 2^{-1} \le y_{bbN-1,k(opt)} + 1 \qquad (14a)$$

$$y_{bbN-1,k} + y_{bbN,k} 2^{-1} \ge y_{bbN-1,k(opt)} - 1 \qquad (14b)$$

The problem is restricted as resolving two equations with two variables (x1,x2) on the form of

$$a1x1 + a2\ x2 \le b1 \qquad (15a)$$

$$a1x1 + a2\ x2 \ge b2 \qquad (15b)$$

(a1, a2, b1, b2) are constants.

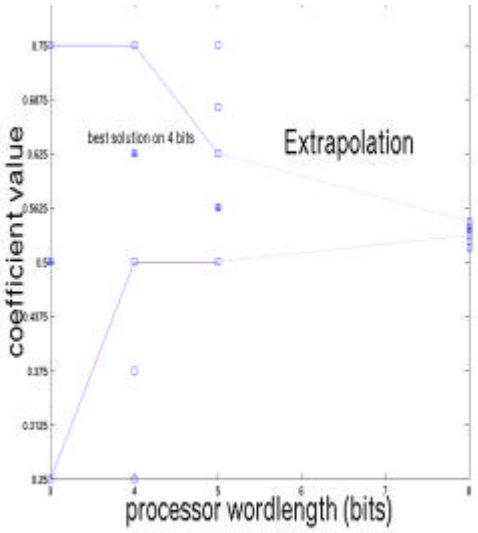
a1=1, a2= 2^{-1}, b1=y_p+1, b2=y_p-1,

Figure1. SPS Procedure on one coefficient

and where y_p is the previous calculated optimal solution. (in this case $y_p = y_{bbN-1,k(opt)}$)

Hence, we obtain a small grid containing admissible values, from which we choose the solution sequence (x1, x2) which gives the smallest value of maximal weighted error. This procedure is iterated for each coefficient increasing the word length, until reaching the desired word length, in which the design of discrete coefficients digital FIR filter was required. Figure 1 represents on one coefficients the (SPS) procedure. The x-axis is and the y-axis are respectively the processor wordlength and coefficient value. The square is the admissible values in the reduced discrete space. The cross the coefficient after optimisation using (SPS).

In this paper, we have chosen the fixed point transformation as shown in Eq. 5 We can show that an extension to the other binary representation could be easily done.



IV. THE SEARCH STRATEGY

The SPS algorithm begins from the discrete optimal starting coefficients designed by (DFS) in a lower word length. Also, we determine the interval for each coefficient in the upper word length as in Figure 1.
Therefore, the formula Eq. (15) is resolved each wordlength incrementation using a local investigation at defined nodes (reduced discrete space).
The algorithm flowchart is described in Figure2. Its main characteristics, which are used for computational experiments, are described below.

- The branching strategy depends starting solution found by (DFS).
- The function evaluation is in the minmax sense. Using (DFS) algorithm, all the defined nodes are investigated.

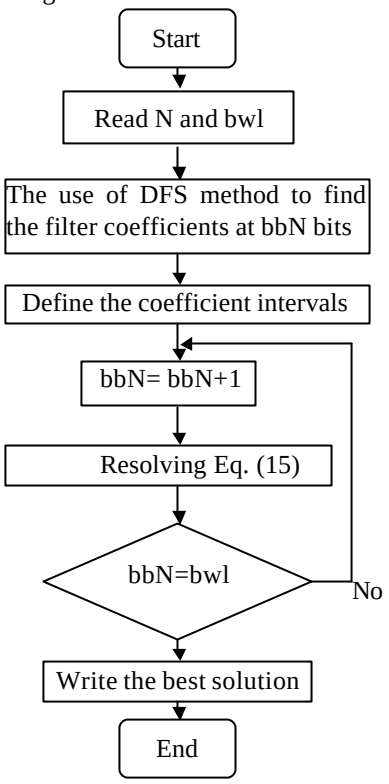


Figure2. The Strategy Search Flowchart

V.RESULTS

The (SPS) software algorithm was developed on MATLAB and tested on 300 MHz Pentium machine using cases reported in literature. The results obtained are presented and compared to those of algorithms in [4,15,16]. The reference numbers indicate where the filters are taken, (DFS), (BFS) and (SA). A filter with length 8 with 15 bits in quantization, excluded the sign bit is denoted by (8/15). The starting points are found by (DFS) in 3 bits processor wordlength. Filters in Table 2 have the frequency specifications represented in Table1. All filters have equal weight in passbands and stopbands. In Table 2 the results are given both in the weighted approximation error and the design times. All SPS performances are better than those obtained in the indicated references. Figure 3 represents the algorithmic complexity ratio between (DFS) and (SPS) algorithms according to the processor wordlength increase. It is shown the drastic time reduction using (SPS) when (bwl) increases.

N/bwl	Passband edges	Stopband edges
16/15	0 : 0.151	0.2050 : 0.5
8/15	0 : 0.411	0.4780 : 0.5
21/6	0 : 0.100	0.1125 : 0.5
12/19	0 : 0.178	0.2500 : 0.5
14/19	0 : 0.268	0.3750 : 0.5
16/19	0 : 0.358	0.4250 : 0.5

Table 1. Filters Frequency Specifications

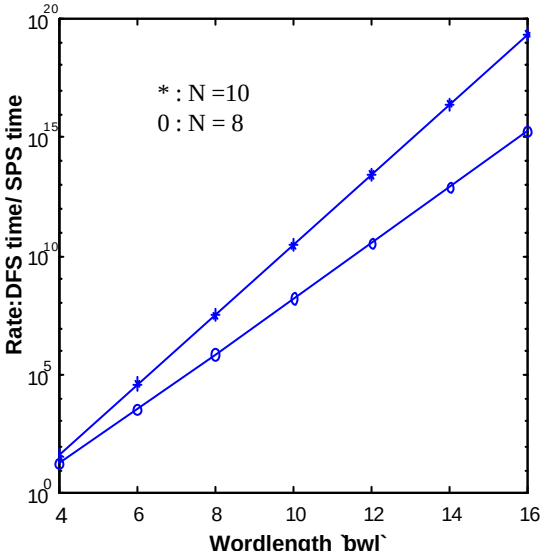


Figure3. Representation of the DFS time/SPS time according to different wordlengths for N:8 and 10

N/bwl	N	bbN	bwl	Inf. Precision	Rounded	[ref]/ Time (sec)	SPS/ Time (sec)
16/15[15]	16	3	15	0.0871	0.0872	0.0762/ 650	0.0712/509
8/19[16]	8	3	19	0.0850	0.0999	0.0975/ 54	0. 0968/ 16
21/6[4]	21	3	6	0.0209	0.0722	0.0711/ 5	0.0468/ 79256
12/19[15]	12	3	19	0.0930	0.0930	0.0911/ 154	0.0895/ 112
14/19 [15]	14	3	19	0.0323	0.0323	0.0305/ 139	0.0294/132
16/19[15]	16	3	19	0.0766	0.0766	0.0747/ 1276	0.13580/10567

Table 2. Results & Comparison for Filter Design Cases Using the SPS Method.

V. CONCLUSION

In this paper, the procedure of a new digital filter design method (SPS) in the discrete space is presented. An evaluation of algorithm performance compared to other algorithms (DFS), (BFS) and (SA) is given. The main feature of this approach is its applicability to the design of filter in a processor with a large wordlength. The computing time in such processor wordlength would be prohibitive using the Depth First Search (DFS). Therefore, it is shown the algorithmic complexity reduction using (SPS) algorithm when processor wordlength increases. In the examples, the limitation of the search domain does not seem to degrade the performance of the algorithm. The forthcoming target is the procedure extension to long filter order.

REFERENCES

[1] D.M. Kodek, '' Design of Optimal Finite Word length FIR Digital Filters Using Integer Programming Techniques,'' IEEE Trans. on ASSP, vol. 28, pp. 304-308, Jun. 1980.

[2] Ioannis PITAS, '' Optimisation and Adaptation of Discrete-Valued Digital Filter Parameters by Simulated Annealing ,'' IEEE Tr. on SP, Ap 1994

[3] N. Benvenuto, M. Marchesi, ''Digital Filters Design by Simulated Annealing,'' IEEE Trans. CAS, vol. 36, pp. 459-460, March 1989.

[4] T. Ciloglu and Z. Unver, ''A New Approach to Discrete Coefficient FIR Digital Filter Design by Simulated Annealing,'' Int. ICASSP. USA 1993.

[5] Lawrence R. Rabiner, Bernard Gold, '' Theory and Application of Digital Signal Processing,'' PRENTICE-HALL, INC. 1975.

[6] J. H. Mc Clellan, T. W. Parks, and L. R. Rabiner, ''A Computer Program for Designing Optimum FIR Linear Phase Digital Filters,''IEEE Tr. Audio Electroac., vol. AU-21, pp. 506-526, Dec. 1973.

[7] M. Minoux, ''Programmation mathématique, théorie et algorithmes,'' tomes 1-2, Dunod, 1983.

[8] B. Jaumard, M. Minoux, and P. Siohan '' Finite Precision Design of FIR Digital Filters Using a Convexity Property,''IEEE Tr. ASSP, Mar. 1988.

[9] Yong C. Lim, Sydney R. Parker, and A. G. Constantinides, '' Finite Word Length FIR Filter Design Using Integer Programming Over a Discrete Coefficient Space,'' IEEE ASSP, vol. 30, pp. 661-664, Aug. 1982.

[10] Y. C. Lim, S. R. Parker, ''FIR Filter Design Over a Discrete Powers-of-Two Coefficient Space,'' IEEE Tr. on ASSP, vol31, pp.583-591, Jun. 1983.

[11] Y. C. Lim, S. R. Parker,''Discrete Coefficient FIR Digital Filter Design Based Upon LMS Criteria,'' IEEE Tr. CAS, vol. 30, pp.723-739, Oct. 1983.

[12] A. N. Belbachir, ''Conception des Filtres Numériques RIF à Phase Linéaire dans l'Espace Discret des Coefficients,'' thesis, University of Technology and Science of Oran, Algeria, 2000.

[13] Li Lee & Alan V. Oppenheim, ''Properties of Approximate Parks-Mc Clellan Filters,'' Proc. ICASSP München, April 1997.

[14] A. N. Belbachir, B. Boulerial, M. F. Belbachir, ''A New Approach to Finite Wordlength Coefficient FIR Digital Filter Design Using the Branch and Bound Technique'' EUSIPCO´00, Tampere, Finland, 2000

[15] B. Boulerial, M. F. Belbachir, ''Filtres RIF : Synthese Directe dans l'Espace Discret des Coefficients,'' NWSIP'98, Algeria, Dec. 1998.

[16] B. Boulerial, ''Filtres RIF : Synthèse Directe dans l'Espace Discret des Coefficients,'' thesis, University of Oran, Algeria, Nov. 1998.

Article N°3

“A New Approach to Digital Filter Design Using the Tabu Search”.

Publiée dans

IEEE NORdic SIGnal Processing Symposium, NORSIG'00, Kolmarden, Sweden, June 2000.

A NEW APPROCH TO DIGITAL FIR FILTER DESIGN USING THE TABU SEARCH

A.N. Belbachir[1,2], M.F. Belbachir[1], A. Fanni[3], S. Bibbò[3], B. Boulerial[1]

[1]Signal and System Laboratory, Electronic Institut U.S.T.O.
XXIV. B.P. 1505 El Mnouer Oran - ALGERIA
[2]Vienna University of Technology, Pattern Recognition and Image Processing Group
Favoritenstr. 9/1832, A-1040 Vienna – AUSTRIA, nabil@prip.tuwien.ac.at
[3]Electrical and Electronic Engineering Department, University of Cagliari
Piazza d´Armi, 09123 Cagliari – ITALY, fanni@diee.unica.it

ABSTRACT

The Tabu Search method has proved to be efficient in many cases applied to digital filter design. However, the starting point chosen can affect the output quality, the computing time and the performance. In this paper, we present a new strategy to improve the output quality of the Tabu Search algorithm using as starting point, that obtained with the Sequential and the Progressive Search method. We will also show that the choice of the fixed-point binary representation provides better results than those of the power of two.

Keywords : **Tabu Search ´TS´, Depth First Search 'DFS', Sequential and Progressive Search 'SPS', Simulated Annealing 'SA'.**

1. INTRODUCTION

The Tabu Search (TS) tool [1],[2] has proved to be both versatile and easy to use, thus rapidly allowing its customisation to different optimisation applications. It exploits some of the most effective search techniques taken from the literature, as well as some new search strategies. In the case of digital FIR filter design where the coefficients are represented by a finite number of bits, TS algorithm has also provided better results than those of the literature such as ´Simulated Annealing` (SA)[13]. SA finds its best application in the design of special filters, such as Nyquist and cascade form FIR filters, where we may have numerous or conflicting constraints. Therefore, the computing cost is expensive.

The Sequential and Progressive Search (SPS)[3] method based on an extrapolation of the final solution from a starting one is also prohibitive for high filter order. The starting solution is found using the Depth First Search (DFS)[12] in a 3 bits word length.

The performance of TS can be affected by the choice of the starting solution [1][2][11]. Although the results obtained with TS are better than those of the literature, we will show that we can further improve them using a studied starting point such as SPS starting point. We will also show that we can also improve these results using an appropriate binary representation, i.e., fixed point representation instead of power of two.

This paper is structured as follows. In section 2, we present the problem statements and the characteristics of the error criterion chosen. In section 3, the descriptions of the Tabu Search method (TS), the sequential and progressive search method (SPS) and the binary representation are given. The results reported on section 4 deal with conventional minmax optimisation of FIR digital filter and are compared to those of other methods.

2. PROBLEM STATEMENTS

Let us consider the design of N-1 order linear phase FIR digital filter with a frequency response H(f), usually written as

$$H(f) = \sum_{k=0}^{N-1} h_k e^{j2\pi fk} \qquad (1)$$

There exists four cases of linear phase FIR filter depending on the filter order, even or odd, and on the kind of symmetry of its impulse response `$h_k$`, positive or negative. In [5], it was shown that the frequency response amplitude of the four cases of linear phase filters can be written in the form

$$P_n(f) = \sum_{k=0}^{n-1} a_k \cos 2\pi fk \qquad (2)$$

where the number of terms, n, is:

$$n = N/2 \text{ or } (N-1)/2 \text{ or } (N+1)/2$$

and a_k, related to h_k, is the resulting shifted sequence depending on the considered case.

The function $P_n(f)$ is compared with a desired frequency response amplitude $D(f)$ using a minmax criterion, as done in the usual optimal linear phase FIR filter design with infinite precision [6]. During the optimisation, the objective function to minimise $f(a)$ is

$$f(a,G) = \max_{f \in F} W(f)\left|D(f) - G^{-1}P_n(f)\right| \qquad (3)$$

- F : the disjoint union of all the frequency bands of interest.
- G: filter gain
- $W(f)$: a weighting function defined on F.
- $D(f)$: the desired frequency response amplitude.

Using Eq. (2) in Eq. (3) gives

$$f = \max_{f \in F} W(f)\left|D(f) - G^{-1}\sum_{k=0}^{n-1} a_k \cos 2\pi fk\right| \qquad (4)$$

The filter coefficients are restricted to the

discrete values allowed by 'b' bit binary word length.

3. METHODS

3.1.Tabu Search method (TS)

Tabu Search [7]-[10] is a metaheuristic method that leads the search for the good solution on the minmax sense making use of flexible memory

systems which exploit the history of the search. TS consists on the systematic prohibition of some solutions to prevent cycling and to avoid the risk of being trapped in local minima. New solutions are searched in the neighbourhood of the current one. The neighbourhood is defined as the set of the points reachable with a suitable sequence of local perturbations, starting from the current solution.
One of the most important features of TS is that a new configuration may be accepted even if the value of the objective function f(a) is greater than that of the current solution. In this way it is possible to avoid being trapped in local minima.
Among all the visited solutions the best one is chosen. This strategy can lead to cycling on previously visited solutions. To prevent this effect, the algorithm marks as "tabu" certain moves for a number of iterations. To do this, a so-called tabu list T of length TT=|T|, named *tabu-tenure*, which can be fixed or variable, is introduced.
Some aspiration criteria which allow overriding of tabu status can be introduced if that move is still found to lead to a better cost with respect to the cost of the current optimum.
This is a characteristic aspect of TS methods, whose main novelty is the use of flexible memory systems for taking advantage of the history of the search.
The previously described memory is the so-called 'short term memory'. A second kind of memory called 'long term' can also be implemented.
Two main important long term memory concepts, which should be evaluated, are intensification and diversification strategies. Intensification strategies are based on the idea of encouraging move combinations and solution features historically found to be good. Diversification strategies, on the other hand, are designed to drive the search into new promising regions.
Summing up, the performance of a TS algorithm depends on the proper choice of the neighbourhood of a solution, on the number of iterations for which a move is kept as tabu, on the aspiration criteria, on the best combination of short and long term memory and on the best balances of intensification and diversification strategies.
These choices are closely linked to the problem at hand and often require expensive "trial and error" processes.
In [1],[2] the TS method uses as starting point Parks- Mc Clellan coefficients or random coefficients. In these cases the results quality, the performance and the computing time, depend on these starting points. Our aim in this paper is to provide a suitable starting point. Our starting point is that of the SPS method described below.

3.2. Sequential and Progressive Search method (SPS)

The SPS method is based on the DFS method. Its major issue of concern is the determination of a good branching strategy. This strategy is detected after a study about the effect of the quantization error on the frequency response, and an examination of the DFS characteristics.
The DFS cannot be applied for a large processor word length 'b', due to the high number of discrete admissible values. Therefore, we use this technique to found an optimal solution in a lower word length 'lb' (lb < b). After, we perform an extrapolation to reach a final solution on 'b' bits wordlength under a set of constraints.
The SPS begins from the discrete optimal starting solution found by the DFS algorithm in a lower wordength, 3 bits in our case. Also, we increase the precision of the coefficients after defining the search interval for each one in the upper wordlength.
In the following sections, the starting solution of the SPS algorithms is chosen as TS starting solution.

3.3. The binary representation

In this paper, we apply two binary representation : the power of two and the fixed point representation. The power of two is used in order to compare the TS results to those of the literature such as SA, and TS with Parks-Mc Clellan starting solution. The fixed-point representation is used in order to provide an improvement of the TS results quality.
The coefficient space in both representations is defined in a ´b´ bits wordlength as follow:

- Power of two:

$$D=\left\{\begin{array}{l} a: a=\sum_{k=1}^{2} c_k \cdot 2^{-g_k}, \quad c_k \in \{-1,0,1\}, \\ g_k \in \left\{1,2,\ldots,2^{\frac{b}{2}-2}\right\} \end{array}\right\}$$

- Fixed point:

$$D=\left\{\begin{array}{l} a: a=s.\sum_{k=1}^{b-1} c_k \cdot 2^{-k}, \quad c_k \in \{0,1\}, \\ s=\{-1,1\} \end{array}\right\}$$

4. RESULTS

The TS algorithm using different starting point was developed in ANSI C and tested on 400 MHz Pentium machine. The results obtained are presented and compared to algorithms in [13] and in [2]. A filter with length 21 and 6 bits processor wordlength, excluded the sign bit is denoted by '21/6'. Firstly, we will compare the TS

performance with that of SA reported in [13]. To better compare the two algorithms, in Table 1 we present both the maximum weighted error given by (3) and the normalised peak weighted ripple d/B, where d is the peak weighted ripple and B is the mean value of the passband gain.

In this table the results of five low-pass filters design are reported, with a passband cut-off normalised frequency of 0.15 and a stopband edge of 0.25. The number of sampling frequencies in (3) is 512, gain G varies in the range 0.5-1.0,

The overall quality of the filters designed with TS and SA is almost the same (up to 0.1 dB) in all cases. However, the two algorithms are remarkable different in terms of computational costs expressed by the total number of function evaluations. In fact, TS requires from 10% to 50% less calculation than SA method.

Filter length	Starting point		Simulated Annealing (optimum point)		Universal Tabu Search (optimum point)		
	Maximum error value	δ/B [dB]	δ/B [dB]	Number of function evaluations	Maximum error value	δ/B [dB]	Number of function evaluations
27/9	0.0126	-33.5	-41.3	849000	0.008498	-41.4	710000
29/9	0.0153	-33.5	-43.1	939000	0.007016	-43.1	630000
31/9	0.0153	-33.5	-43.1	1060000	0.007016	-43.1	627000
33/9	0.0129	-35.7	-44.7	1026000	0.005797	-44.7	725000
35/9	0.0118	-33.5	-44.7	1105000	0.005797	-44.7	546000

Table 1. Results of 5 filter designs and comparison with simulated annealing. Normalised cut-off frequencies are 0.15 and 0.25.

N/b	Infinite precision	Rounded (Power of two)	TS with Parks-Mc Clellan starting solution		TS with random starting solution		TS with SPS starting solution	
			f(a)	Time s	f(a)	Time s	f(a)	Time s
8/15	0.057863	0.125033	0.099016	2.03	0.1536214	32.05	0.0979985	2.03
21/6	0.001989	0.046875	0.0157447	10.32	0.0658742	257.56	0.0153654	6.24
8/7	0.057863	0.125663	0.099016	1.04	0.112546	38.54	0.0987965	0.66
20/7	0.003429	0.0621487	0.0120481	10.17	0.0126542	336.5	0.0120481	2.65
16/19	0.136302	0.167227	0.140713	6.53	0.152633	59.50	0.1365435	4.56

Table 2. Results of TS algorithm with different starting solutions compared to reference method in power of two

N/b	Infinite precision	Rounded (Power of two)	TS with Parks-Mc Clellan starting solution		TS with random starting solution		TS with SPS starting solution	
			f(a)	Time s	f(a)	Time s	f(a)	Time s
8/15	0.057863	0.057856	0.057681	1.27	0.059980	36.68	0.05709	1.22
21/6	0.001989	0.051301	0.031250	5.88	0.037500	289	0.03125	3.25
8/7	0.057863	0.063677	0.0636766	1.32	0.0758421	85.56	0.06355	1.01
20/7	0.003429	0.021929	0.015625	5.49	0.018750	268.65	0.015625	3.54
16/19	0.136302	0.136303	0.135806	5	0.145610	78.65	0.13569	4.1

Table 3. Results of the Tabu Search algorithm with different starting solutions in the fixed point representation

Secondly, it is shown how a proper choice of the starting solution can greatly affect the goodness of the solutions. The three first filters in the Table 2 and 3 have the passband edges (0,0.159) and the stopband edges (0.295,0.5), the fourth filter has the passband edges (0,0.307) and the stopband edges (0.35,0.5) and the last filter has the passband edges (0,0.08) and the stopband edges (0.16,0.5). All the filters have equal weights in passbands and stopbands and unitary gain. In all examples the results of TS with SPS starting solution are better than others in the same case. As seen in Table 2, the computing time for TS using a random starting solution is the biggest for a less performance. However, the results of TS using Parks- Mc Clellan solution requires a quasi-equal time to that of TS using SPS starting solution for a less performance. In table 3, we improve the performance of TS algorithm using a more representative binary representation fixed point`. We can see that the results obtained are better in most cases to those of table 2. The reduction of computing time and improvement of performance are due to the high number of admissible values in the fixed-point representation. This provides a fast and a good solution.

5. CONCLUSION

The Tabu Search algorithm combines the most interesting and effective search techniques designed by several authors with new ideas and strategies aiming to satisfy simplicity and versatility. In this paper, an improvement of TS algorithm for digital FIR filter design using a new starting solution and a new binary representation is presented. The main target was to choose an independent starting solution from that of Parks-Mc Clellan one, which uses rounding to infinite precision solution. Therefore, we use a discrete starting solution directly chosen from the processor discrete space. The obtained results when compared to those of other strategies are better in all cases.

As a second improvement is the use of fixed-point representation which provides a larger coefficients space than that of power of two for the same wordlength. This yields better performance.

As a future work, the application of the Tabu Search algorithm for the IIR filter will be studied.

REFERENCES

[1] S. Bibbo, A. Fanni, A. Giua, A. Matta, '' A General Purpose Tabu Search Code: an Application to Digital Filters Design'' IEEE Int. Conf. On Sys., Man and Cybernitcs, San Diego (CA), Oct. 11-14 1998.

[2] A. Fanni, M. Marchesi, F. Pilo, A. Serri, "Tabu search metaheuristic for designing digital filters," COMPEL, International Journal for Computation and Mathematics in Electrical and Electronic Engineering, vol. 17, No. 5/6, pp. 789-796, 1998.

[3] A. N. Belbachir, B. Boulerial, M. F. Belbachir, '' A New Approach to Finite Wordlength Coefficient FIR Digital Filter Design Using the Branch and Bound Technique'' EUSIPCO´00, Tampere, Finland, 2000 (accepted).

[4] T. Ciloglu and Z. Unver, '' , '' A New Approach to Discrete Coefficient FIR Digital Filter Design by Simulated Annealing,'' IEEE of Int. Conf. on ASSP. Minnisota 1993.

[5] Lawrence R. Rabiner, Bernard Gold, '' Theory and Application of Digital Signal Processing,'' PRENTICE-HALL, INC. 1975.

[6] J. H. Mc Clellan, T. W. Parks, and L. R. Rabiner, '' A Computer Program for Designing Optimum FIR Linear Phase Digital Filters,'' IEEE Trans. Audio Electroacoust., vol. AU-21, pp. 506-526, Dec. 1973.

[7] F. Glover and M. Laguna, "Tabu search," Modern Heuristic Techniques for Combinatorial Problems, Blackwell Scientific Publications, Oxford, pp. 70-150, 1993.

[8] F. Glover, "Tabu search fundamentals and uses," unpublished technical report, University of Colorado, Boulder, 1994.

[9] F. Glover and M. Laguna, Tabu Search, Kluwer A. P., 1997.

[10] F. Glover, "Tabu search and adaptive memory programming - advances, applications and challenges," printed in Barr, Helgason and Kennington eds. "Interfaces in Computer Science and Operations Research", Kluwer A. P., 1996.

[11] A. Fanni, M. Marchesi, A. Manunza, F. Pilo, "Tabu Search methaeuristics for global optimization of electromagnetic problems," IEEE Trans. on Magnetics, vol. 34, no. 5, Sept. 1998, pp. 2960-2963.

[12] B. Boulerial, ''Filtres RIF : Synthèse Directe dans l'Espace Discret des Coefficients,''thesis, University of Science and Technology of Oran, Algeria, November 1998.

[13] N. Benvenuto, M. Marchesi and A. Uncini, "Applications of simulated annealing for the design of special digital filters," IEEE Transactions on Signal Processing, vol. 40, no. 2, pp. 323-332, February 1992.

[14] A. N. Belbachir, ''Conception des Filtres Numériques RIF à Phase Linéaire dans l'Espace Discret des Coefficients,'' thesis, University of Oran, Algeria, 2000.

Article N°4

“A New Approach to Finite Wordlength Coefficient FIR Digital Filter Design Using the Branch and Bound Technique”. Publiée dans

EUSIPCO'00, Tampere - Finland, September

A NEW APPROACH TO FINITE WORDLENGTH COEFFICIENT FIR DIGITAL FILTER DESIGN USING THE BRANCH AND BOUND TECHNIQUE

A.N. Belbachir[1,2], B. Boulerial[1] & M.F. Belbachir[1]

[1]Signal and System Laboratory, Electronic Institute U.S.T.O.
XXV. B.P. 1505 El Mnouer, Oran –ALGERIA
[2]Vienna University of Technology, Pattern Recognition and Image Processing Group
Favoritenstr. 9/1832, A-1040 Vienna – AUSTRIA
E-mail: (belbachiran@yahoo.com) or (nabil@prip.tuwien.ac.at)

ABSTRACT

It has been shown that the branch and bound technique is effective for the design of finite wordlength optimal digital filters. This technique is however expensive in computing time. In this paper, we present a robust branch and bound branching strategy named Sequential and Progressive Search, improving the design of filters on a large wordlength processor in a reasonable computing cost. The details of the algorithm and many examples are given and compared to the other methods.

I. INTRODUCTION

For finite wordlength coefficients digital filter design, it is often desirable to use algorithms whose output quality can be adjusted depending on the availability of resources such as computing time and precision.

Remez Exchange Algorithm is usually applied for the design of infinite precision linear phase (FIR) filters [6]. When these filters are implemented on a Digital Signal Processor with a special purpose-hardware, each filter coefficient has to be represented by a finite number of bits (bwl) smaller than that used on a computer. The simplest and the most widely used approach to the problem are the rounding of the optimal infinite precision coefficients to its (bwl) bits representation. However, the filters obtained are degraded and in most case there exists another set of finite word length coefficients, which gives the best Chebyshev approximation to the desired frequency response. To find these coefficients, it is necessary to include the finite word length restriction into the filter design. In this case, the optimisation problem becomes a complex problem, where a general investigation of the optimal solution requires a prohibitive computing time. To solve this problem, many optimisation methods have been applied to discrete coefficients FIR digital filters design. Simulated annealing technique [2,3,4] has proven to be effective in many cases, but requires a large number of function evaluations and does not guarantee the optimal solution.

The linear integer programming formulation [1], [7]-[9] was applied as a discrete optimisation method on the minmax criterion. Although, it is possible to obtain an optimum result, the computing time required even with the high-speed supercomputer of today, prohibits the application of these techniques for high order filters.

Optimisation technique in the coefficients discrete space, and in particular branch and bound method, was used to solve this discrete optimum problem. This method based on implicit enumeration techniques, also requires an expensive computing cost [7,8,10,14,15].

In many local search methods based on the branch and bound technique, such as the Depth First Search (DFS) and the Breadth First Search (BFS), the solution found is better than that obtained from the direct quantizing from infinite precision filter. However, the computing cost is expensive for large wordlength processor and high order filters [10], [14], [15]. Even for (BFS) the solution could not be optimal. This is related to the estimation accuracy of the branch susceptible to contain the best solution.

In this paper, we present a different view on the branching strategy and present a new method named « Sequential and Progressive Search » (SPS), based on the branch and bound technique in the minmax sense. Compared to the Depth First Search method (DFS), the number of function evaluations is smaller and it depends on the filter length without degrading the performance of the algorithm.

In section II, we present the problem statements and the characteristics of the error criterion chosen. In section III, the description of the proposed optimisation method, the sequential and progressive search method (SPS) is given. The results reported on section IV deal with conventional minmax optimisation of FIR digitals filters and are compared to those of other methods.

II. PROBLEM STATEMENTS

Let us consider the design of N-1 order linear phase FIR digital filter with a frequency response H (f) usually written as

$$H(f) = \sum_{k=0}^{N-1} h_k e^{-j2\pi fk} \qquad (1)$$

In [5], It was shown that the frequency response amplitude of the four cases of linear phase (FIR) filters could be written in the form of:

$$P_n(f) = \sum_{k=0}^{n-1} a_k \cos 2\pi fk \qquad (2)$$

Where the number of terms, n, is:

$$n = N/2 \text{ or } (N-1)/2 \text{ or } (N+1)/2$$

and a_k is the resulting shifted sequence depending on the considered case. The function $P_n(f)$ is compared with desired frequency response amplitude D(f) using a minmax criterion, as done in the usual optimal (FIR) filter design with infinite precision [6]. The weighted approximation error e_n is given by

$$e_n = \min_{(coeff.a)} \max_{f \in F} W(f)\left|D(f) - P_n(f)\right| \qquad (3)$$

- F: the union of all the frequency bands of interest.
- W (f): a weighting function defined on F.
- D (f): the desired frequency response amplitude.

Using Eq. (2) in Eq. (3) gives

$$e_n = \min_{(coeff.a)} \max_{f \in F} W(f)\left|D(f) - \sum_{k=0}^{n-1} a_k \cos 2pfk\right| \qquad (4)$$

The filter coefficients are restricted to the discrete values allowed by (bwl) bit binary word length.

III. OPTIMIZATION METHOD 'SEQUENTIAL AND PROGRESSIVE SEARCH' (SPS)

We consider the problem of filter design with an extra constraint imposing a limit on the word length of the coefficients a_k, k=0,1,...., N/2. Using the fixed point representation, we can express the discrete coefficient a_k as a linear combination:

$$|a_k| = \sum_{j=1}^{bwl-1} y_{j,k} 2^{-j} \quad k = 0, 1,...., N/2. \qquad (5)$$

Where (bwl) is the binary bit allowed for the filter discrete design and 'j' is the binary bit indication. $y_{j,k}$ is a bivalent variable only fixed to only take the values '0' or '1'. Hence, the frequency response amplitude $P_n(f)$ could be expressed as

$$P_n(f) = \sum_{k=0}^{n-1} s\left(\sum_{j=1}^{bwl-1} y_{j,k} 2^{-j} \right) \cos 2\pi f k. \qquad (6)$$

Where s is the sign of 'a_k' , s (= -1 or +1).
It is not possible to find the optimal filter coefficients at (bwl) bits wordlength processor using the DFS method, owing to the long computing time required. But we can calculate these filter coefficients filter in a lower wordlength, (bbN) binary bit (bbN ≤ bwl) with such a method. a_k could be expressed as

$$|a_{k(opt)}| = \sum_{j=1}^{bbN-1} y_{j,k(opt)} 2^{-j} \quad k = 0, 1,...., N/2. \qquad (7)$$

$a_{k(opt)}$: the coefficients related to the optimal digital filter at (bbN) wordlength.
$y_{j,k(opt)}$: 'i' binary bit value of the $a_{k(opt)}$ coefficient.

The proposed method, named Sequential and Progressive Search (SPS) takes this optimal solution as a starting point (starting solution) for its branching strategy in order to design filters with a higher wordlength discrete coefficients. Using the optimal solution in the minmax sense found by the (DFS) method in the (bbN) binary bits wordlength, we calculate with the SPS algorithm the solution at (bwl) bits (bbN ≤ bwl).
First, the coefficients are found at the (bbN+1) binary bits on the minimax sense using a local investigation in a reduced discrete search space. This reduced space is defined using the previous solution. Then we increase gradually the wordlength and calculate the new solution till reaching the (bwl) (the required word length). For each step 'i', we define a lower bounding function $e_{n\,i}$, which can be written as

$$e_{n\,i}(a) = \min_{(coeff.a)} \max_{f \in F} W(f) \left| D(f) - \sum_{k=0}^{n-1} a_k \cos 2\pi f k \right| \qquad (8)$$

$e_{n\,i}(a)$ is defined as the lowest value of the error function for a_k solving the following program which satisfy (8) under the conditions:

$$a_k^{bbN+1} \le a_k^{bbN} + \mu \quad k=1,..., N/2 \qquad (9a)$$
$$a_k^{bbN+1} \ge a_k^{bbN} - \mu \quad k=1,..., N/2 \qquad (9b)$$

a_k^{bbN} : is the coefficients a_k represented by bbN bits
μ interval chosen to contain the solution.

The implementation of a continuous value in two different wordlengths of the fixed point representation does not offer two equal discrete values. The maximum difference between these discrete values is the quantization error '±LSB' (least sided bit) due to both truncation (±LSB) and rounding (±1/2.LSB).
Therefore, we have overestimate the μ value between the (bbN) and (bwl) wordlengths to

$$\mu = LSB = 2^{-(bbN-1)}. \qquad (10)$$

Substituting Eq. (10) and Eq. (7) in Eq. (9)

$$\sum_{j=1}^{bbN} y_{j,k} 2^{-j} \le \sum_{j=1}^{bbN-1} y_{j,k(opt)} 2^{-j} + 2^{-(bbN-1)} \qquad (11a)$$
$$\sum_{j=1}^{bbN} y_{j,k} 2^{-j} \ge \sum_{j=1}^{bbN-1} y_{j,k(opt)} 2^{-j} - 2^{-(bbN-1)} \qquad (11b)$$

Developing Eq. (11) we obtain

$$\sum_{j=1}^{bbN-2} y_{j,k(opt)} 2^{-j} + \sum_{j=bbN-1}^{bbN} y_{j,k} 2^{-j} \le$$
$$\sum_{j=1}^{bbN-2} y_{j,k(opt)} 2^{-j} + y_{bbN-1,k(opt)} 2^{-bbN} + 2^{-(bbN-1)}. \qquad (12a)$$
$$\sum_{j=1}^{bbN-2} y_{j,k(opt)} 2^{-j} + \sum_{j=bbN-1}^{bbN} y_{j,k} 2^{-j} \ge$$
$$\sum_{j=1}^{bbN-2} y_{j,k(opt)} 2^{-j} + y_{bbN-1,k(opt)} 2^{-bbN} - 2^{-(bbN-1)}. \qquad (12b)$$

After simplifications, we have

$$y_{bbN-1,k} \cdot 2^{-bbN+1} + y_{bbN,k} \cdot 2^{-bbN} \le y_{bbN-1,k(opt)} \cdot 2^{-bbN} + 2^{-(bbN-1)} \qquad (13a)$$
$$y_{bbN-1,k} \cdot 2^{-bbN+1} + y_{bbN,k} \cdot 2^{-bbN} \ge y_{bbN-1,k(opt)} \cdot 2^{-bbN} - 2^{-(bbN-1)} \qquad (13b)$$

hence,

$$y_{bbN-1,k} + y_{bbN,k} 2^{-1} \le y_{bbN-1,k(opt)} + 1 \qquad (14a)$$
$$y_{bbN-1,k} + y_{bbN,k} 2^{-1} \ge y_{bbN-1,k(opt)} - 1 \qquad (14b)$$

The problem is restricted as resolving two equations with two variables (x1,x2) on the form of

$$a1x1+a2\ x2 \le b1 \quad (15a)$$
$$a1x1+a2\ x2 \ge b2 \quad (15b)$$

(a1, a2, b1, b2) are constants.

$a1=1$, $a2= 2^{-1}$, $b1=y_p+1$, $b2=y_p-1$,

and where y_p is the previous calculated optimal solution. (in this case $y_p= y_{bbN-1,k(opt)}$)

Hence, we obtain a small grid containing admissible values, from which we choose the solution sequence (x1, x2) which gives the smallest value of maximal weighted error.

This procedure is iterated for each coefficient increasing the word length, until reaching the desired word length, in which the design of discrete coefficients digital FIR filter was required.

Denoting that this method does not affect the bivalent variable from 1 to (bbN-2) bits obtained by the optimal method 'DFS'. Therefore, it improves the coefficient precision, adding required bits from (bbN-1) to (bwl), related to the optimal sequence, in order to well define the coefficient value.

In this paper, we have chosen the fixed point transformation as shown in (Eq5). We can show that an extension to the other binary representation such that floating point or power of two could be easily done.

VI. RESULTS

The algorithm has been tested using cases reported in literature. The software algorithm was developed in Matlab and tested on 300 MHZ Pentium machine. The results obtained are presented and compared to algorithms in [3,15]. The reference numbers indicate where the filters are taken. A filter with length 24 with 9 bits in quantization, excluded the sign bit is denoted by '24/9'.

The starting points are calculated by (DFS) in discrete coefficients wordlength of bbN=3 bits. The two first filters in the Table 1. have the passband edges (0,0.1) and the stopband edges (0.1125,0.5), the third filter has the passband edges (0,0.08) and the stopband edges (0.16,0.5), the fourth filter has the passband edges (0,0.159) and the stopband edges (0.295,0.5) and the last filter has the passband edges (0,0.307) and the stopband edges (0.35,0.5). All filters have equal weights in passbands and stopbands. In all examples the results are better than those obtained in the indicated references. The reference algorithms are Simulated Annealing (SA) [4], the Breath First Search (BFS) [15] and the Depth First Search (DFS) [15]. In table 1, the results are given both in the approximation error and the design times. A comparison measures the number of function evaluations between (DFS) and (SPS) algorithms of all filters of Table 2. The reduction in the number of function evaluations is at least in the order of 4500.

N/bwl	-I-	-II-
15/5	$8.52891033 .10^{11}$	1171875
21/6	$6.20506086 .10^{19}$	195312500
15/7	$6.76752340 .10^{16}$	1953125
20/7	$1.18059162 .10^{21}$	48828125
16/19	$1.46149048 .10^{48}$	12405426
24/9	$3.16993821 .10^{32}$	1708984375

Table 2. Number of the Function of Evaluation of the SPS Method Compared to DFS method [15].

I- ***Number of function evaluations by `DFS` [15].***
II- ***Number of function evaluations by 'SPS'.***

Filter length/ Word length	N	bbN	bwl	Infinite Precision	Rounded	[ref]/ Time `s` seconds	SPS/ Time seconds
8/15[15]	8	3	15	0.05766	0.05765	0.05762/ 50	0.05759/19
21/6[4]	21	3	6	0.02099	0.07223	0.07115/ 5	0.04687/ 79256
8/7[15]	8	3	7	0.05760	0.06356	0.06355/ 93600	0.06355/ 0.06
20/7 [15]	20	3	7	0.00344	0.02191	0.02005/ 10	0.016321/20256
16/19[15]	16	3	19	0.13590	0.13590	0.13961/ 1180	0.13580/11830

Table 1. Results & Comparison for Filter Design Cases Use the SPS Method.

V. CONCLUSION

In this paper, a new approach to finite wordlength coefficient (FIR) digital filter design using the branch and bound technique is presented. The main feature of this approach is its applicability to the design of filter in a processor with a large wordlength. The computing time in such processor wordlength would be prohibitive using the Depth First Search (DFS). The obtained results when compared to the other algorithms and local search methods [4], [15] and [16] are better in all cases. In the examples, the limitation of the search domain does not seem to degrade the performance of the algorithm. As a future work, the improvement of the algorithm for long filter order will be studied.

ACKNOWLEDGEMENT

The authors acknowledge Professor Michele MARCHESI of the Electrical and Electronical Engineering Department, University of Cagliari-Italy, for his valuable suggestions to improve this paper quality.

REFERENCES

[1] D.M. Kodek, 'Design of Optimal Finite Word length FIR Digital Filters Using Integer Programming Techniques' IEEE Tr.ASSP, pp.304-308, June 1980

[2] I. PITAS, 'Optimisation and Adaptation of Discrete-Valued Digital Filter Parameters by Simulated Annealing' IEEE Trans. SP, pp. 860-866, April 1994.

[3] N. Benvenuto, M. Marchesi, ''Digital Filters Design by Simulated Annealing,'' IEEE Trans. Circuits Syst., vol. 36, pp. 459-460, March 1989.

[4] T. Ciloglu and Z. Unver, 'A New Approach to Discrete Coefficient FIR Digital Filter Design by Simulated Annealing,' IEEE of Int. Conf. on ASSP Minnisota 93.

[5] L. R. Rabiner, B. Gold, 'Theory and Application of Digital Signal Processing,' Prentice-Hall, INC. 1975.

[6] J. H. Mc Clellan, T. W. Parks, and L. R. Rabiner, '' A Computer Program for Designing Optimum FIR Linear Phase Digital Filters,'' IEEE Trans. Audio Electroacoust., vol. AU-21, pp. 506-526, Dec. 1973.

[7] M. Minoux, ''Programmation mathématique, théorie et algorithmes,'' tomes 1 et 2, Dunod, 1983.

[8] B. Jaumard, M. Minoux, and P. Siohan, 'Finite Precision Design of FIR Digital Filters Using a Convexity Property,' IEEE Trans. ASSP, pp. 407-411, Mar. 1988.

[9] Yong C. Lim, S. R. Parker, and A. G. Constantinides, '' Finite Word Length FIR Filter Design Using Integer Programming Over a Discrete Coefficient Space,'' IEEE Trans. ASSP, vol. -30, pp. 661-664, Aug. 1982.

[10] Y. C. Lim, S. R. Parker, ''FIR Filter Design Over a Discrete Powers-of-Two Coefficient Space,'' IEEE Trans. ASSP, vol. ASSP-31, pp. 583-591, June 1983.

[11] Y. C. Lim S. R. Parker,'Discrete Coefficient FIR Digital Filter Design Based Upon an LMS Criteria,'' IEEE Trans. Circ. Syst., vol. CAS-30, pp. 723-739, Oct 1983.

[12] Y. C. Lim,'Design of DC-Value Linear Phase FIR Filters with Optimum Normalised Peak Ripple Magnitude,' IEEE Tran. CAS pp. 1480-1486, Dec 1990

[13] Li Lee & A.V. Oppenheim, 'Properties of Approximate Parks-Mc Clellan Filters,'' Proc. ICASSP München, April 1997.

[14] B. Boulerial, M. F. Belbachir, ''Filtres RIF : Synthese Directe dans l'Espace Discret des Coefficients,'' Pro. NWSIP'98, Sidi Bel Abbes, Algeria, December 1998.

[15] B. Boulerial, ''Filtres RIF : Synthèse Directe dans l'Espace Discret des Coefficients,''thesis, Technology University of Oran, Algeria, November 1998.

[16] A. N. Belbachir, M. F. Belbachir, "Information Processing," Internal Report, Dipartimento di Ingegneria Elettrica ed Elettronica, Universita' degli Studi di Cagliari, Italy, October 1999.

[17] A. N. Belbachir, ''Conception des Filtres Numériques RIF à Phase Linéaire dans l'Espace Discret des Coefficients,'' thesis, University of Oran, Algeria, 2000.

Article N°5

“Evaluation of the Iterative Least Square Method on Digital Filter Design”.

Publiée dans

9th DSP Workshop & 1st Signal Processing Education Workshop, DSP-SPE‘00
Hunt-Texas, USA October, 2000.

EVALUATION OF THE ITERATIVE LEAST SQUARE METHOD ON DIGITAL FILTER DESIGN

A.N. Belbachir[1,2], S. Bibbò[3], A. Fanni[3], B. Boulerial[1], M.F. Belbachir[1]

[1]Signal and System Laboratory, Electronic Institut U.S.T.O.
XXVI. B.P. 1505 El Mnouer Oran - ALGERIA
[2]Vienna University of Technology, Pattern Recognition and Image Processing Group
Favoritenstr. 9/1832, A-1040 Vienna – AUSTRIA, nabil@prip.tuwien.ac.at
[3]Electrical and Electronic Engineering Department, University of Cagliari
Piazza d´Armi, 09123 Cagliari – ITALY, fanni@diee.unica.it

ABSTRACT

In this paper we outline the experiments of the Iterative Least Square Direct algorithm on several filters. The experiments of these methods, compared to standard methods, demonstrate the feasibility and the good performance of this novel approach. It is also shown the small algorithmic complexity of this approach.

I. INTRODUCTION

Many large-volume electronic consumer products are based on digital signal processing (DSP). Digital filters are integral parts of many digital processing systems, including communication systems, control systems, systems for audio and image processing and systems for medical applications. In DSP systems, the signal that contains the information of interest is represented by a sequence of numbers, so called samples.

The DSP system operates on this input sequence of numbers to form an output sequence. In general, the overall aim of a signal processing system is to reduce the information content, or to modify it so that it can be efficiently stored or transmitted over a transmission channel.

Examples of such DSP algorithms are digital filters, fast Fourier transform (FFT), discrete cosine transformation (DCT), and wavelet transforms.

In this paper, we are mainly interested on the first case, and especially on the filter design directly on the discrete space. Many standard methods for this purpose use the Chebyshev approximation. As an example, Remez Exchange Algorithm is usually applied for the design of infinite precision linear phase (FIR) filters [9]. For (DSP) implementation, the most widely used approach to the problem is the rounding of the optimal infinite precision coefficients to its wordlength representation. However, the filters obtained are degraded and do not fulfill the spectral requirements for which they are expected.

Sequential and Progressive Search method (SPS) [13] is performed for the design of finite precision (FIR) filters for both minmax and least square approximation. Although it is possible to obtain optimal filter, the algorithm is unfeasible for long filter order.

The Tabu Search (TS) algorithm [3],[4],[5],[6] has proven high performance and low algorithmic complexity. Especially for the latter case, it seems that no algorithm could cope with (TS). The TS tool has also proven to be both versatile and easy to use, thus rapidly allowing its customisation to different optimisation applications. It exploits some of the most effective search techniques taken from the literature, as well as some new search strategies. In the case of digital FIR filter design, all (TS) algorithm applications have been used on the minmax sense. This paper is formulated with convenience that versatile methods performances are not affected by the approximation criterion. Least square approximation is the criterion of the choice of much DSP application where minimising error energy or error power is desired.

In this paper, we present digital filter design method based on least square approximation. It is called Iterative Least Square Direct method (ILSD) [1]. The method purpose is the filter design solution updating using several iterations. We have chosen the Tabu Search (TS) algorithm in order to compare the (ILSD) algorithm performance and cost function.

In section II, we present the problem statements and the characteristics of the error criterion chosen. In section III, an overview of the (ILSD) method is given. The algorithm modules are depicted in section IV. The results reported on section V deal with conventional Least Square optimisation of FIR digital filters and are compared to those of Tabu Search method.

II. PROBLEM STATEMENTS

Let us consider the design of N-1 order linear phase FIR digital filter with a frequency response H(f) usually written as

$$H(f) = \sum_{k=0}^{N-1} h_k e^{-j2\pi fk} \qquad (1)$$

In [8], It was shown that the frequency response amplitude of the four cases of linear phase (FIR) filters could be written in the form

$$P_n(f) = \sum_{k=0}^{n-1} a_k \cos 2\pi fk \qquad (2)$$

Where the number of terms, n, is:

$$n(= N/2 \text{ or } (N-1)/2 \text{ or } (N+1)/2)$$

And a_k is the resulting shifted sequence depending on the considered case. The function $P_n(f)$ is compared with a desired frequency response amplitude D(f) using the least square criterion. The approximation error e_n is given by

$$e_n = \frac{1}{Nf} \sum_{i=0}^{Nf} \left| D(f_i) - P_n(f_i) \right|^2 \qquad (3)$$

- i = 1,2,…, Nf-1
- Nf : Number on Sample frequency selected.
- D(f) : the desired frequency response amplitude. For an ideal low pass filter we have

$D(f_i) = 1$ if $f_i \in$ bandpass.
$D(f_i) = 0$ if $f_i \in$ stopband .

Hence, Eq. (3) will be

$$e_n = \frac{1}{Nf} \left\{ \sum_{i=0}^{k-1} \left\| 1 - P_n(f_i) \right\|^2 + \sum_{i=k}^{Nf-1} \left| P_n(f_i) \right|^2 \right\} \qquad (4)$$

The filter coefficients are restricted to the discrete values allowed by (bwl) bit binary word length. In this

paper we have chosen to use the fixed-point and power of two representations. The power of two representation yields a set of admissible values (D) which is defined as follow :

$$D = \left\{ \begin{array}{l} \alpha : \alpha = \sum_{k=1}^{2} c_k \cdot 2^{-g_k}, \quad c_k \in \{-1,0,1\}, \\ g_k \in \{1,2,...,b\} \end{array} \right\} \quad (2)$$

where b is the maximum number of shifts.

III. OPTIMIZATION METHOD

'ITERATIVE LEAST SQUARE DIRECT METHOD' (ILSD)

The (ILSD) method [1] copes with two main problems in the discrete filter design:

- The Least Square method (LSD) [2], [14] problem, which consists on the choice of the starting point and the order in which the other coefficients are considered.
- The problem of Depth First Search [14], which is the prohibitive computing time in the case of long filter order.

Therefore, this method has been performed in much iteration in order to improve the (LSD) performances.
To describe the (ILSD) method, we choose the following example:
Let us consider a filter defined by two coefficients {h(0), h(1)}. The processor wordlengh provided **is (bwl) and 'av'** admissible values could be represented.

The (ILSD) method runs as follows:
First, (LSD) is used to calculate the coefficients $h_{lsd}(0)$ and $h_{lsd}(1)$. Let us denote the filter design error by E_r.
$h_{lsd}(0)$ is fixed, then $h_{lsd}(1)$ will vary in d-diameter discrete values range centred around $h_{lsd}(1)$ (d<av).
Indeed, at each discrete combining, the least square design error is calculated. Let us denote it by E_{ilsd}.
If $E_{ilsd} < E_r$, then $E_r = E_{ilsd}$ and will be the reference error.
At the same manner, $h_{lsd}(1)$ is now fixed and $h_{lsd}(0)$ will vary in d discrete values range. E_{ilsd} is calculated again and compared to E_r.
This procedure is iterated many times till the error variation will vanish. Let ΔEr is the error variation:

$$\Delta Er = Er(\text{actual iteration}) - Er(\text{previous iteration}) \quad (5)$$

Which is provided by the final filter coefficients $(h_{ilsd}(0), h_{ilsd}(1))$ after (It) iterations.

.

IV. THE ALGORITHM

The (ILSD) algorithm is subdivided in two parts as depicted in Figure 1:

- The first part represents the (LSD) algorithm outlined in [2], [14], in which the filter coefficients are sequentially computed.
- The second part represents the iterative method which emphases are the (LSD) performance improvement. This is due by a local search of the new solution providing a low error design in the neighbourhood of the previous. Regarding the wordlength, the parameter (d) is fixed.

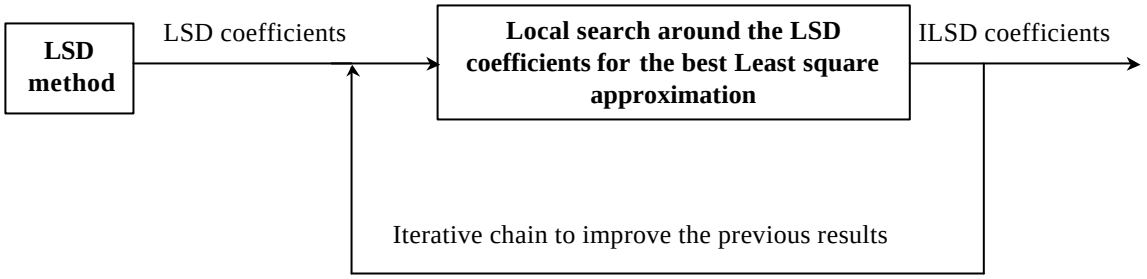


Figure1 The procedure of the Iterative Least Square Direct method

Filter length/Wordlength	Infinite precision	Rounded	TS/ Time (sec)	ILSD/ Time (sec)
8/7	0.0374	0.0358	0.0358/ 2.02	0.0356/ 1.9
8/15	0.0374	0.0374	0.0372/ 30.22	0.0314/ 11
8/19	0.0374	0.0374	0.0373/ 14.25	0.0314/ 302
16/15	0.0706	0.0706	0.0596/ 73.47	0.0389/ 120
16/15	0.0829	0.0829	0.0818/ 78.24	0.0472/ 163
56/13	0.0049	0.0049	0.0049/ 465	0.0030/ 153

Table 1. Results of ILSD Filter Design compared to TS in fixed-point representation.

Filter length/Wordlength	Infinite precision	Rounded	TS/ Time (sec)	ILSD/ Time (sec)
8/7	0.0374	0.1270	0.1270/ 10.84	0.1270/ 0.88
8/15	0.0374	0.0624	0.0556/ 10.06	0.0512/ 55
8/19	0.0374	0.0624	0.0477/ 10.37	0.0461/ 110
16/15	0.0706	0.0748	0.0748/ 47.35	0.0399/ 80
16/15	0.0829	0.0847	0.0704/ 71.57	0.0524/ 62
56/13	0.0049	0.0236	0.0236/ 415	0.0173/ 85

Table 2. Results of ILSD Filter Design compared to TS in power of two representation.

V. RESULTS

The (ILSD) algorithm has been tested and compared to Tabu Search algorithm using cases reported in literature. The software algorithm was developed in MATLAB and tested on a 300 MHz Pentium machine. A filter with length 8 with 7 bits in quantization, excluded the sign bit is denoted by '8/7'. The starting points for (TS) algorithm are chosen depending on the filter case. They could be Parks-Mc Clellan coefficients, random values or zero-values. Table 1 and Table two represent the application of (ILSD) using the fixed point and power of two representations respectively. The three first filters in the Table 1 and 2 have the passband edges (0,0.159) and the stopband edges (0.259,0.5). The fourth filter has the passband edges (0,0.318) and the stopband edges (0.371,0.5), the fifth filter has (0,0.05) passband edges and (0.104,0.5) stopband edges and the last filter has the passband edges (0,0.31) and the stopband edges (0.35,0.5). All the filters have equal weights in passbands and stopbands. In all examples the (ILSD) performances are better than those obtained with Tabu Search. The computing time varies from a example to another. The ILSD algorithm presents low algorithmic complexity than (TS) algorithm when long filter order is used. In the case of large processor wordlength, the (ILSD) computing time is relatively longer than that of (TS) but still remain reasonable (hundred seconds). In general, using both factors performance and computing time, it is clear that (ILSD) takes the advantage.

VI. CONCLUSION

In this paper, an evaluation of the Iterative Least Square Direct algorithm (ILSD) for digital filter design on several examples is presented. The concept is mainly based on an iterative filter approximation in the coefficients discrete space. The comparison with Tabu Search (TS) algorithm demonstrates the good (ILSD) algorithm performance and large applicability to a high range of filters. Furthermore, (ILSD) presents lowest algorithmic complexity. The next forthcoming step consists on how to well define the local discrete diameter in order to guarantee the optimality. A deeper mathematical study is on going.

REFERENCES

[1] A. N. Belbachir, B. Boulerial, M. F. Belbachir, "Une Approche Itérative pour la Conception de Filtres Numériques RIF à Phase Linéaire dans l'Espace Discret des Coefficients," Maghrebine Conference in Electrical Engineering, CMGE'99, Constantine, Algeria, December 1999.

[2] B. Boulerial, M. F. Bel Bachir, ‘’Une Méthode Directe au Sens des Moindres Carrés D.M.C pour la Conception de Filtres R.I.F à Phase Linéaire dans l’Espace Discret des coefficients’’, International Meeting on Components and Electronic Systems IMCES’99, Sidi Bel-Abbes, Algeria, 17-18 mai 1999.

[3] S. Bibbo, A. Fanni, A. Giua, A. Matta, ‘’ A General Purpose Tabu Search Code: an Application to Digital Filters Design’’ IEEE Int. Conf. On Sys., Man and Cybernitcs, San Diego (CA), Oct. 11-14 1998.

[4] F. Glover and M. Laguna, “Tabu search,” Modern Heuristic Techniques for Combinatorial Problems, Blackwell Scientific Publications, Oxford, pp. 70-150, 1993.

[5] F. Glover, “Tabu search fundamentals and uses,” unpublished technical report, University of Colorado, Boulder, 1994.

[6] F. Glover and M. Laguna, Tabu Search, Kluwer A. P., 1997.N. Benvenuto, M. Marchesi, ‘’Digital Filters Design by Simulated Annealing,’’ IEEE Trans. CAS, vol. 36, pp. 459-460, March 1989.

[7] Lawrence R. Rabiner, Bernard Gold, ‘’ Theory and Application of Digital Signal Processing,’’ PRENTICE-HALL, INC. 1975.

[8] J. H. Mc Clellan, T. W. Parks, and L. R. Rabiner, ‘’ A Computer Program for Designing Optimum FIR Linear Phase Digital Filters,’’ IEEE Trans. Audio Electroacoust., vol. AU-21, pp. 506-526, Dec. 1973.

[9] M. Minoux, ‘’Programmation mathématique, théorie et algorithmes,’’ tomes 1-2, Dunod, 1983.

[10] A. N. Belbachir, M. F. Belbachir, A. Fanni, S. Bibbò and B. Boulerial, “A New Approach toDigital Filter Design Using the Tabu Search,” IEEE NORdic SIGnal Processing, NORSIG'00, Sweden, June 2000.

[11] Y. C. Lim, S. R. Parker, ‘’ Discrete Coefficient FIR Digital Filter Design Based Upon an LMS Criteria,‘’ IEEE Trans. CAS, vol. CAS-30, pp. 723-739, Oct. 1983.

[12] A. N. Belbachir, ‘’Conception des Filtres Numériques RIF à Phase Linéaire dans l’Espace Discret des Coefficients,’’ thesis, University of Oran, Algeria, 2000.

[13] A. N. Belbachir, B. Boulerial, M. F. Belabchir,
“A New Approach to Finite Wordlength Coefficient FIR Digital Filter Design Using the Branch and Bound Technique,” European Signal Processing Conference, EUSIPCO'00, Tampere - Finland, September 2000.