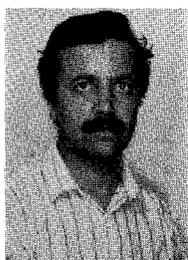


Conception de filtres récursifs

à phase linéaire

Design of recursive digital filters with linear phase



Mohamed Faouzi BEL BACHIR

Laboratoire de la Communication Parlée, ICP, Unité associée au CNRS, INPG-ENSERG, 46, avenue Félix-Viallet, 38031 GRENOBLE CEDEX.

Ingénieur en électronique (1976) et Magister (1984) de l'Université des Sciences et de la Technologie d'Oran (USTO), Algérie. Chargé de Cours à l'USTO, actuellement en détachement à l'ICP de Grenoble. Le centre d'intérêt est porté sur le filtrage et l'analyse temporelle de la parole.



Jean CALEN

Laboratoire de la Communication Parlée, ICP, Unité associée au CNRS, INPG-ENSERG, 46, avenue Félix-Viallet, 38031 GRENOBLE CEDEX.

Docteur en Informatique (1974) et ès Sciences de l'Université de Toulouse (1979). Actuellement chargé de recherches à l'ICP de Grenoble. Dirige dans ce laboratoire l'équipe de Décodage, Reconnaissance et Compréhension de la parole.

RÉSUMÉ

Cet article présente une méthode de conception de filtres numériques RII (Réponse Impulsionnelle Infinie) à phase linéaire. Le problème consiste à approcher au mieux les caractéristiques d'amplitude et de phase définies dans le domaine fréquentiel, préalablement données par l'utilisateur. La méthode proposée utilise la programmation linéaire (PL) qui exige que les contraintes soient mises sous forme linéaire.

Pour concevoir des filtres stables, la plupart des méthodes utilisant la PL insèrent dans le problème d'approximation des contraintes de stabilité linéaires suffisantes mais non nécessaires ce qui restreint le champ de recherche des filtres. De plus ces contraintes, supplémentaires vis-à-vis du problème d'approximation lui-même, alourdissent le schéma de résolution.

Une nouvelle démarche est utilisée dans cet article pour résoudre le problème de la stabilité des filtres. Celui-ci est traité en deux temps : (a) on suppose que la stabilité est assurée *a priori*, ce qui permet d'alléger la formulation du problème et (b) on la vérifie *a posteriori* à l'aide d'un critère. Des exemples de filtres sont donnés illustrant les améliorations apportées comparativement à ceux obtenus par des méthodes concurrentes.

MOTS CLÉS

Filtres à réponse impulsionnelle infinie et à phase linéaire, approximation, stabilité, programmation linéaire.

SUMMARY

A recursive digital filter method is presented where given magnitude and linear phase characteristics can be approximated in the frequency domain. The method uses linear programming (LP) which imposes the constraints to be linear in form.

To design stable filters, most of the techniques using LP include linear stability constraints that are sufficient but not necessary conditions for stability. Because of this, the filters designed would fall under the classification of a subclass of stable filters. These linear stability constraints complicate the problem.

A new approach is used for solving the problem of stability. The filters are assumed to be stable a priori—this allows to simplify the problem formulation—and to verify the stability a posteriori using a criterion. A set of filters are presented to illustrate the improvement achieved with comparison to those obtained by competitive methods.

KEY WORDS

Infinite impulse response digital filters with linear phase, approximation, stability, linear programming.

1. Introduction

De nombreux domaines d'application (le traitement de la parole par exemple) utilisent des méthodes de traitement de signal fondées sur des filtres à phase linéaire. Bien que les filtres RIF (Réponse Impulsionnelle Finie) répondent aux spécifications de phase exactement linéaire demandées dans ces cas-là, ils présentent quelques inconvénients, en particulier de mise en œuvre. En effet ils sont généralement d'un ordre plus élevé ce qui conduit à des temps de réponse prohibitifs pour des applications temps réel et n'ont pas une sélectivité en fréquence (pour un ordre donné). Une question est donc de savoir s'il est possible d'obtenir des filtres RII (Réponse Impulsionnelle Infinie) ayant des performances comparables en ce qui concerne la phase — on sait que ces filtres engendrent, en général, moins de calcul pour le filtrage en ligne que les filtres RIF — et des performances supérieures concernant l'amplitude.

Ces dernières années ont donc vu un regain d'intérêt pour les filtres numériques RII à phase linéaire par rapport aux filtres RIF et de nombreuses méthodes de conception de filtres RII à caractéristique de phase linéaire ont été développées. Ces méthodes partent d'un gabarit en fréquence imposé par l'utilisateur et tentent de l'approcher au mieux tant en ce qui concerne l'amplitude qu'en ce qui concerne la phase. Les méthodes d'approximation peuvent être rangées en deux groupes :

- (a) celles qui optimisent séparément l'amplitude et la phase [9];
- (b) celles qui les optimisent conjointement [1 à 5].

On constate que les résultats obtenus par cette dernière approche sont généralement meilleurs. En particulier Chottera et Julien [1] proposent une méthode d'optimisation utilisant la programmation linéaire. A son tour et parmi les méthodes de même catégorie, celle-ci a de nombreux avantages par rapport à celles qui utilisent une programmation non linéaire [3, 4, 5] : simplicité de mise en œuvre et certitude d'atteindre un des sommets du domaine convexe (optimum global) — en effet, les méthodes du gradient sont confrontées au problème du choix du point initial qui détermine la rapidité de convergence et peut conduire à un simple optimum local.

Cependant, le but étant d'obtenir des filtres stables, il faut introduire des contraintes de stabilité à un moment donné dans la procédure d'optimisation. Chottera et Julien introduisent ces contraintes directement dans le système général des équations de la PL (Programmation Linéaire) ce qui les oblige à utiliser des contraintes linéaires et donc à introduire deux biais dans la résolution :

- ces contraintes linéaires sont suffisantes et non nécessaires : ceci a pour effet de restreindre l'ensemble des filtres admissibles. La recherche du filtre se fait alors dans un sous-ensemble inclus dans l'ensemble des filtres stables et le filtre solution n'est pas obligatoirement optimal;
- le type des contraintes choisies imposent un échantillonnage équidistant de la réponse fréquentielle, ce qui empêche de privilégier certaines bandes fréquentielles et entraîne un volume de données et de calculs plus important si l'on veut un degré de finesse suffisant dans certaines bandes.

Partant des travaux de ces auteurs [1], notre propos dans cet article, est de décrire une méthode qui permet de lever ces deux biais tout en améliorant l'implantation machine.

2. Problème d'approximation

Le problème est de calculer les coefficients a_i et b_i (réels) de la fonction de transfert $H(z)$ du filtre RII définie de la manière suivante :

$$(1) \quad H(z) = \frac{P(z)}{Q(z)} = \frac{\sum_{n=0}^N a_n \cdot z^{-n}}{\sum_{i=0}^N b_i \cdot z^{-i}} \quad \text{avec } b_0 = 1$$

tels qu'elle approche au mieux la réponse fréquentielle donnée $D(w)$ suivante :

$$(2) \quad D(w) = G(w) \cdot e^{j\Phi(w)}$$

où l'amplitude $G(w)$ est donnée par l'utilisateur sur un ensemble de points $w_i, i \in [1, N_e], N_e$ nombre de

points d'échantillonnage, et la phase par la relation de linéarité :

$$(3) \quad \Phi(w_i) = -\tau_d \cdot w_i, \quad i = 1, 2, \dots, N_e$$

avec τ_d (retard de groupe) constante quelconque.

En appelant respectivement $Y_R(w_i)$ et $Y_L(w_i)$ la partie réelle et la partie imaginaire de $D(w_i)$, il vient :

$$(4) \quad Y_R(w_i) = G(w_i) \cdot \cos(-\tau_d \cdot w_i)$$

$$(5) \quad Y_L(w_i) = G(w_i) \cdot \sin(-\tau_d \cdot w_i)$$

Soit $r(w_i)$ l'erreur entre la réponse en fréquence du filtre cherché et celle du filtre désiré $D(w)$. On peut écrire :

$$(6) \quad r(w_i) = Y_R(w_i) + j Y_L(w_i) - P(w_i)/Q(w_i)$$

En multipliant les deux membres de (6) par $Q(w_i)$ on obtient l'erreur complexe pondérée $e(w_i)$:

$$(7) \quad e(w_i) = r(w_i) \cdot Q(w_i) \\ = [Y_R(w_i) + j Y_L(w_i)] \cdot Q(w_i) - P(w_i)$$

En posant maintenant :

$$(8) \quad e(w_i) = e_R(w_i) + j e_L(w_i)$$

Le problème d'approximation est posé comme suit : minimiser e_R et e_L en valeur absolue ou en d'autres termes minimiser la variable positive ε telle que :

$$(9) \quad |e_R(w_i)| < \varepsilon \quad \text{et} \quad |e_L(w_i)| < \varepsilon.$$

Après avoir explicité les expressions de e_R et e_L , la formulation générale du problème d'approximation devient :

$$(10) \quad \text{minimiser } \mathbf{g} = [b_1, b_2, \dots, b_N, \\ a_0, a_1, \dots, a_N, \varepsilon] \cdot [0, 0, 0, \dots, 0, 1]^T$$

sous les contraintes :

$$(11) \quad \sum_{n=0}^N b_n \cdot [G(w_i) \cdot \cos(w_i(n + \tau_d))] \\ - \sum_{n=0}^N a_n \cdot \cos(nw_i) - \varepsilon \leq 0$$

$$(12) \quad - \sum_{n=0}^N b_n \cdot [G(w_i) \cdot \sin(w_i(n + \tau_d))] \\ + \sum_{n=0}^N a_n \cdot \sin(nw_i) - \varepsilon \leq 0$$

$$(13) \quad - \sum_{n=0}^N b_n \cdot [G(w_i) \cdot \cos(w_i(n + \tau_d))] \\ + \sum_{n=0}^N a_n \cdot \cos(nw_i) - \varepsilon \leq 0$$

$$(14) \quad \sum_{n=0}^N b_n \cdot [G(w_i) \cdot \sin(w_i(n + \tau_d))] \\ - \sum_{n=0}^N a_n \cdot \sin(nw_i) - \varepsilon \leq 0$$

pour $i = 1, 2, \dots, N_e$.

Dans les inéquations ci-dessus τ_d peut être considéré comme un paramètre et les coefficients a_i et b_i les inconnues à déterminer.

3. Problème de stabilité

3.1. PROPOSITION DU PROBLÈME

Le programme linéaire (10)-(14) permet de résoudre le problème d'approximation mais n'assure pas à lui seul la stabilité des filtres.

Une condition nécessaire et suffisante pour qu'un filtre RII soit stable est qu'il ait tous ses pôles à l'intérieur du cercle unité. La formulation de cette condition en fonction des coefficients du filtre (les inconnues du problème) n'est pas linéaire et ne peut être utilisée telle quelle dans le schéma de résolution. On doit donc choisir (a) soit d'autres contraintes linéaires de stabilité si l'on veut les introduire dans le schéma de la PL, (b) soit résoudre le problème différemment.

(a) La condition linéaire de stabilité la plus utilisée est celle de Robinson [10]. Elle est donnée par :

$$(15) \quad \sum_{i=0}^N b_i \cdot \cos(i \cdot w) > 0, \quad 0 \leq w \leq \pi$$

La mise en œuvre de cette condition de stabilité suscite plusieurs remarques :

— Il faut théoriquement vérifier la condition (15) pour tout $w \in [0, \pi]$. Expérimentalement ceci est impossible, on est conduit à réduire le test sur un ensemble fini de valeurs échantillonnées w_i ($i = 1$ à N_e). Brophy et Salazar [8], à la suite de travaux expérimentaux, montrent que $N_e \geq 4N$ points équidistants représentent un échantillonnage suffisant (condition utilisée aussi par les auteurs [1]).

— La condition $N_e \geq 4N$ entraîne un temps de calcul et un volume de données très importants. En effet, la conception d'un filtre d'ordre N implique la résolution d'un programme linéaire avec une matrice de contraintes de $5N_e$ lignes et $2N+1$ colonnes or $N_e \geq 4N$, ce qui donne une matrice d'au moins $20N$ lignes et $2N+1$ colonnes.

(b) Pour éviter cette série d'inconvénients, nous proposons une autre méthode qui consiste à traiter le problème d'approximation en dehors de celui de la stabilité et de tester cette dernière de manière itérative à l'aide d'un critère nécessaire et suffisant.

3.2. CRITÈRE DE STABILITÉ

Notre choix s'est porté sur le critère de Bistritz [6] qui présente l'avantage de nécessiter une taille mémoire et un nombre d'opérations réduits (le nombre requis de multiplications est deux fois plus petit comparativement au test de Jury-Marden [6]).

Le principe de ce critère est de calculer une matrice de valeurs en fonctions des coefficients du filtre et de

vérifier que ces valeurs remplissent deux conditions [6]. Nous avons attaché une importance particulière à l'implantation de ce test en exploitant au maximum les propriétés de symétrie et d'antisymétrie locales de la matrice. Dans ce paragraphe nous rappelons brièvement le critère ainsi que sa formulation matricielle.

Le polynôme $Q(z)$ [dénominateur de $H(z)$ dans (1)] de degré N peut être formulé comme suit :

$$(16) \quad Q(z) = z^{-N} (d_1 \cdot z^N + d_2 \cdot z^{N-1} + \dots + d_{N+1})$$

$$(17) \quad d_{N+1-i} = b_{N-i} \quad i=0, 1, 2, \dots, N$$

Pour établir le critère de Bistritz, il s'agit de calculer les éléments suivants de la matrice $C=(c_{ij})$:

$$(18) \quad c_{1,j} = d_j + (-1)^N d_{N-j+2}, \quad j=1, 2, \dots, N+1$$

$$(19) \quad C_{2,1} = d_1 + (-1)^{N+1} \cdot d_{N+1}$$

$$(20) \quad \begin{cases} C_{2,j} = d_j + (-1)^{N+1} \cdot d_{N-j+2} - a_{2,j-1}, \\ j=2, 3, \dots, N \end{cases}$$

$$(21) \quad c_{i,j} = - \begin{vmatrix} c_{i-2,1} & c_{i-2,j+1} \\ c_{i-1,1} & c_{i-1,j+1} + c_{i-1,j} \end{vmatrix} \cdot \frac{1}{c_{i-1,1}}$$

avec $j=1, 2, \dots, N+2-i$ et $i=3, 4, \dots, N+1$.

Pour que $Q(z)$ ait tous ses pôles à l'intérieur du cercle unité [6] il faut et il suffit que :

– le premier élément de toutes les lignes de la matrice soit positif :

$$(22) \quad c_{i,1} > 0, \quad i=1, 2, \dots, N+1$$

– toutes les sommes suivantes calculées pour les lignes impaires soient positives :

$$(23) \quad \sigma_{2k-1} = \sum_{j=1}^M (-1)^{j+1} \cdot c_{2k-1,j} + (-1)^{M+2} \frac{c_{2k-1,M+1}}{2} \cdot P > 0$$

avec

$$K=1, 2, \dots, E[(N+2)/2]; \quad M=E[(N+3-2K)/2]$$

et avec $P=0$ si l'ordre N est impair, et $P=1$ si N est pair, $E[.]$: partie entière.

Enfin les conditions (22), (23) sont vérifiées au fur et à mesure du calcul de C . Dès détection d'instabilité (l'une des deux conditions non remplie), on arrête la procédure. Tous les éléments de la matrice sont finalement calculés seulement si $Q(z)$ possède tous ses pôles à l'intérieur du cercle unité.

4. Algorithme proposé

Le programme linéaire (10)-(14) est fonction du paramètre τ_d . Pour chaque valeur de τ_d correspond un nouvel ensemble de contraintes et donc une fonction g différente. Les filtres les plus intéressants sont ceux

qui possèdent un retard de groupe entier. La valeur de τ_d n'étant pas fixée au départ, il s'agit de calculer la valeur de τ_d^* correspondant à la plus petite erreur d'approximation g^* . On montre [1], que $\tau_d \in [N-1, 1]$ est un intervalle convenable pour la recherche de l'optimum g^* .

Nous avons utilisé la méthode du simplexe pour résoudre le programme linéaire (10)-(14). Nous avons opté pour sa forme duale celle-ci étant plus intéressante quand le nombre de contraintes est supérieur au nombre de variables [12, 13, 14].

Nous donnons ci-après les deux plus importantes phases de l'algorithme utilisé :

(1) Spécifier la caractéristique d'amplitude désirée sur N_e points du domaine fréquentiel (équidistants ou non) et l'ordre N du filtre.

(2) Rechercher itérativement la valeur τ_d^* correspondant à la plus petite erreur d'approximation g^* d'un filtre stable de la façon suivante :

* à chaque valeur entière i de τ_d avec $i \in [N-1, 1]$, résoudre le programme linéaire (10)-(14) par la méthode du simplexe en procédant comme suit :

– à chaque itération du simplexe vérifier la stabilité du filtre à l'aide du critère de Bistritz. Si le filtre est stable alors sauvegarder ses coefficients, son retard de groupe ainsi que l'erreur d'approximation g_i correspondante, sinon ne pas modifier les valeurs du dernier filtre stable trouvé;

* le filtre correspondant à $g^* = \min \{g_i\}$ pour $i \in [N-1, 1]$ est retenu.

5. Résultats

La méthode que nous proposons permet l'utilisation d'un échantillonnage à pas variable. Nous avons utilisé un échantillonnage équidistant avec deux pas différents, pour N_e échantillons :

- $N_e/2$ points équidistants sur $[F_c - \Delta F_1, F_c + \Delta F_2]$;
- $N_e/2$ points équidistants ailleurs,

avec F_c : fréquence de coupure du filtre.

Les valeurs ΔF_1 et ΔF_2 permettent de varier la distribution des échantillons sur l'intervalle normalisé des fréquences $[0, 0.5]$.

Une comparaison des résultats peut se faire ainsi par rapport aux filtres publiés par Chottera et Julien (CJ) qui paraissent les meilleurs au vue de la bibliographie. Pour des spécifications identiques, nous obtenons les résultats suivants visualisés sur les figures 1 à 5 :

- en pointillé les caractéristiques des filtres de CJ;
- en trait continu les caractéristiques des filtres que nous avons obtenus.

Nous avons préféré raisonner en termes de constance du retard de groupe plutôt qu'en terme de linéarité de la phase pour mettre mieux en relief les oscillations de la phase. Nous avons donc représenté les caractéristiques de retard de groupe au lieu de celles de la phase.

De plus pour mesurer la qualité des filtres obtenus, nous utilisons les mesures d'erreurs suivantes :

$$(24) \quad \left\{ \begin{array}{l} \delta_1 = \max \| |G(f)| - |H(f)| \| \\ f \in \text{bande passante} \end{array} \right.$$

$$(25) \quad \left\{ \begin{array}{l} \delta_2 = \max | \tau_d - \tau(f) | \\ f \in \text{bande passante} \end{array} \right.$$

5.1. COMPARAISON DU FILTRE CJ D'ORDRE 18 ($\tau=15$) AVEC CELUI D'ORDRE 16 ($\tau=11$) PROPOSÉ

Notre filtre d'ordre 16 a été conçu avec $N_e=30$, $\Delta F_1=\Delta F_2=11$, après 133 itérations du simplexe.

Remarquons (fig. 1a) que l'erreur δ_1 définie dans la bande passante est meilleure (rapport de 6), la pente de coupure est plus raide dans la bande de transition (fig. 1b). L'atténuation de 40 dB est préservée. L'erreur δ_2 (fig. 1c) est réduite par un facteur de 4.7 (nous avons négligé la zone très proche de la région de transition). Après analyse de toutes ces caractéristiques, notre filtre d'ordre 16 est plus performant que CJ d'ordre 18. Il est également plus rapide ($\tau=11$ contre $\tau=15$).

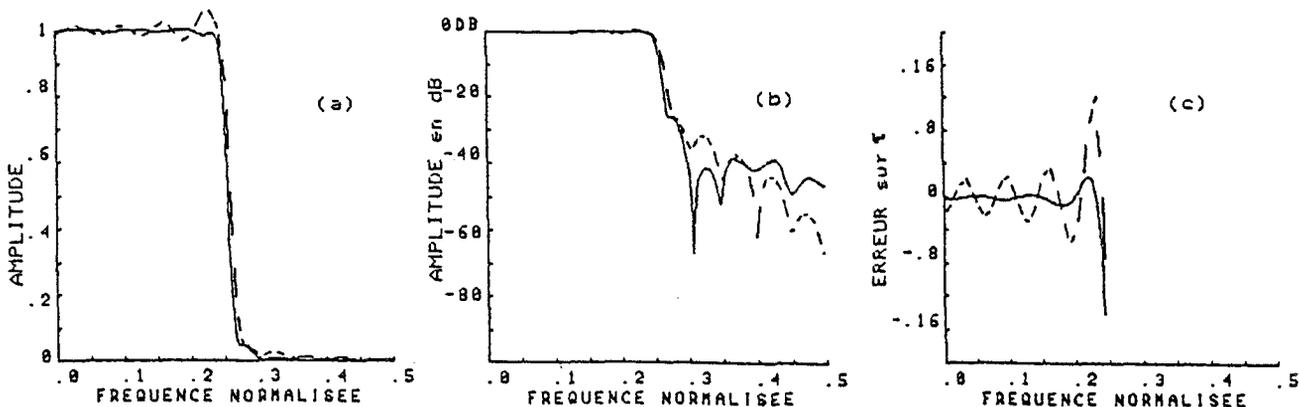


Fig. 1. — Comparaison du filtre passe-bas d'ordre 18 ($\tau=15$) [1] en pointillé avec celui proposé d'ordre 16 ($\tau=11$). Notons l'amélioration des performances dans les figures 1a, b, c représentant respectivement les caractéristiques d'amplitude, d'amplitude en décibels et l'erreur sur le retard de groupe.

Fig. 1. — Comparison with the 18th order filter characteristics ($\tau=15$) (dotted line) with the proposed 16th order filter ($\tau=11$). Let us note the behavior of the performances in Figures 1a, b, c respectively representing the characteristics of the magnitude, the magnitude in decibels and the error in group delay.

5.2. COMPARAISON DU FILTRE CJ D'ORDRE 18 ($\tau=15$) AVEC CELUI D'ORDRE 12 PROPOSÉ

Notre filtre d'ordre 12 a été obtenu après 71 itérations du simplexe et avec $N_e=30$, $\Delta F_1=\Delta F_2=11$.

Dans la bande passante (fig. 2a) les courbes d'amplitude présentent la même erreur δ_1 . Cependant le filtre d'ordre 12 présente moins de distorsions pour les basses fréquences. Les bandes d'atténuation (fig. 2b) sont équivalentes. L'erreur δ_2 du retard de groupe (fig. 2c) a été divisée par un facteur d'ordre 10 (nous n'avons pas représenté la région très proche de la région de transition). Il est également plus rapide $\tau=7$ contre $\tau=15$.

5.3. COMPARAISON DU FILTRE CJ D'ORDRE 12 ($\tau=9$) AVEC CELUI D'ORDRE 11 ($\tau=7$) PROPOSÉ

Notre filtre d'ordre 11 a été obtenu après 65 itérations du simplexe avec $N_e=22$, $\Delta F_1=.05$, $\Delta F_2=.25$.

Les caractéristiques visualisées sur les figures 3a, b, c donnent une appréciation directe de l'amélioration. Les erreurs δ_1 (fig. 3a) et δ_2 (fig. 3b) sont divisées respectivement par 6,7 et 2,5.

5.4. COMPARAISON DU FILTRE CJ D'ORDRE 12 ($\tau=9$) AVEC CELUI D'ORDRE 9 ($\tau=5$) PROPOSÉ

Notre filtre a été obtenu avec $N_e=22$, $\Delta F_1=.05$, $\Delta F_2=.25$, après 68 itérations. Les erreurs δ_1 (fig. 4a) et δ_2 (fig. 4b) ont été divisées respectivement par 1.8 et 3.8. L'atténuation de 32 dB a été préservée. Remarque : A la suite de leurs travaux [1], Chottera et Julien montrent que leur filtre d'ordre 12 est équivalent sinon meilleur à celui de T. Inukai [5]. Celui-ci a utilisé une approche par la programmation non linéaire en prenant comme point initial un filtre de Deczky [7].

5.5. COMPARAISON DU DIFFÉRENCIATEUR CJ D'ORDRE 17 AVEC CELUI D'ORDRE 15 PROPOSÉ

Le différentiateur d'ordre 15 que nous montrons (fig. 5) a été obtenu après 107 itérations et sur 16 échantillons équidistants sur l'axe des fréquences. La figure 5a représente la caractéristique d'amplitude en comparaison avec le différentiateur de Chottera et Julien.

- l'erreur sur l'amplitude est améliorée aux basses fréquences. Elle est équivalente ailleurs;
- l'erreur sur le retard de groupe (fig. 5c) a été divisée par un facteur supérieur à 50.

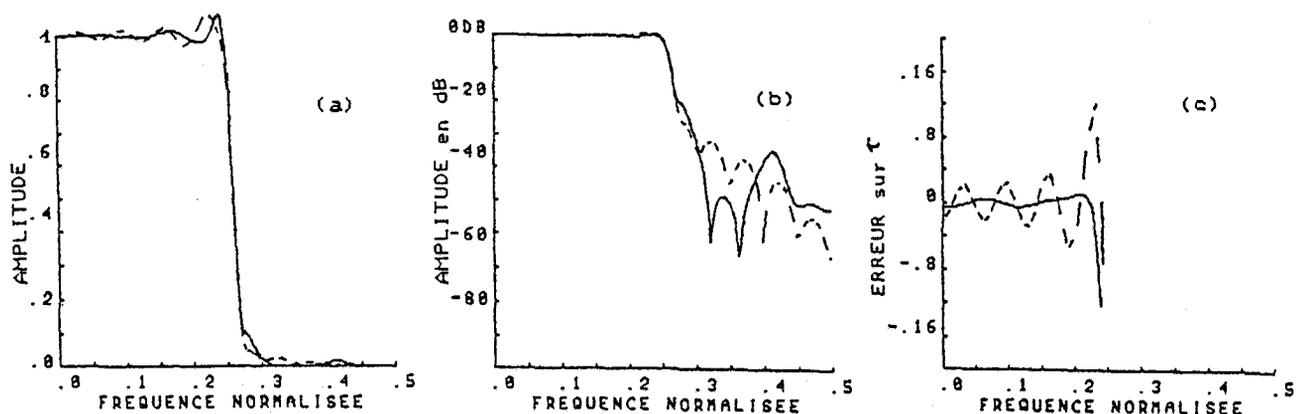


Fig. 2. — Comparaison du filtre d'ordre 18 ($\tau=15$) en pointillé avec le filtre proposé d'ordre 12 ($\tau=9$). Les figures 2a, b, c montrent respectivement les caractéristiques d'amplitude, d'amplitude en decibels et l'erreur sur le retard de groupe. Les performances sont équivalentes dans la bande passante (fig. 2a) et dans la bande d'atténuation (fig. 2b). L'erreur sur retard de groupe est plus petite pour le filtre proposé.

Fig. 2. — Comparison with the 18th order filter ($\tau=15$) [1] (dotted line) with the proposed 12th order filter ($\tau=9$). Figures 2a, b, c respectively represent the characteristics of the magnitude, the magnitude in decibels and the error in group delay. The error in group delay is smaller (Fig. 2c). The performances are equivalent in the passband (Fig. 2a) and in stopband (Fig. 2b).

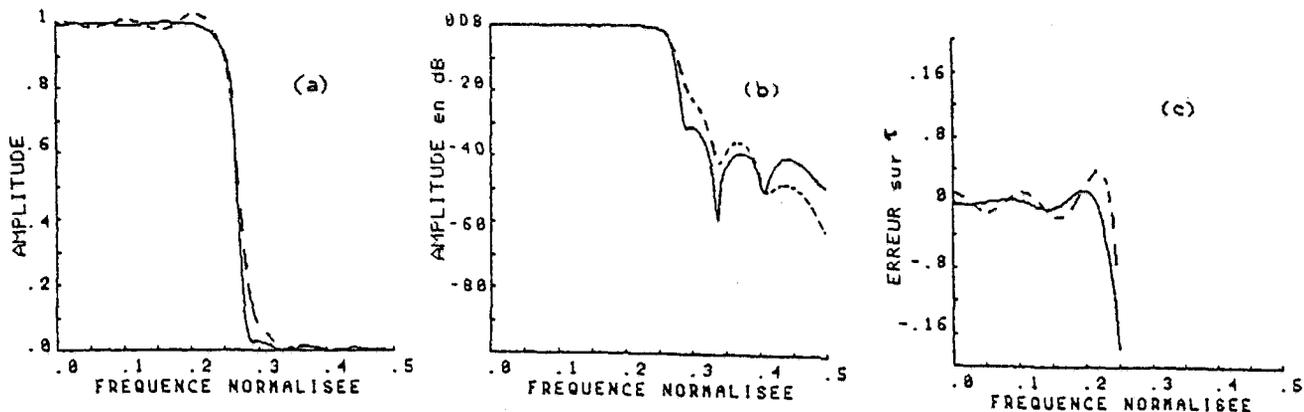


Fig. 3. — Comparaison du filtre passe-bas d'ordre 12 ($\tau=9$) [1] représenté en pointillé avec le filtre d'ordre 11 ($\tau=7$) proposé. Les figures 3a, b, c représentant respectivement les caractéristiques d'amplitude, d'amplitude en decibels et l'erreur sur le retard de groupe, montrent l'amélioration obtenue.

Fig. 3. — Comparison with the 12th order filter characteristics ($\tau=9$) [1] (dotted line) with the proposed 11th order filter ($\tau=7$). Let us note the behavior of the performances in Figures 3a, b, c respectively representing the characteristics of the magnitude, the magnitude in decibels and the error in group delay.

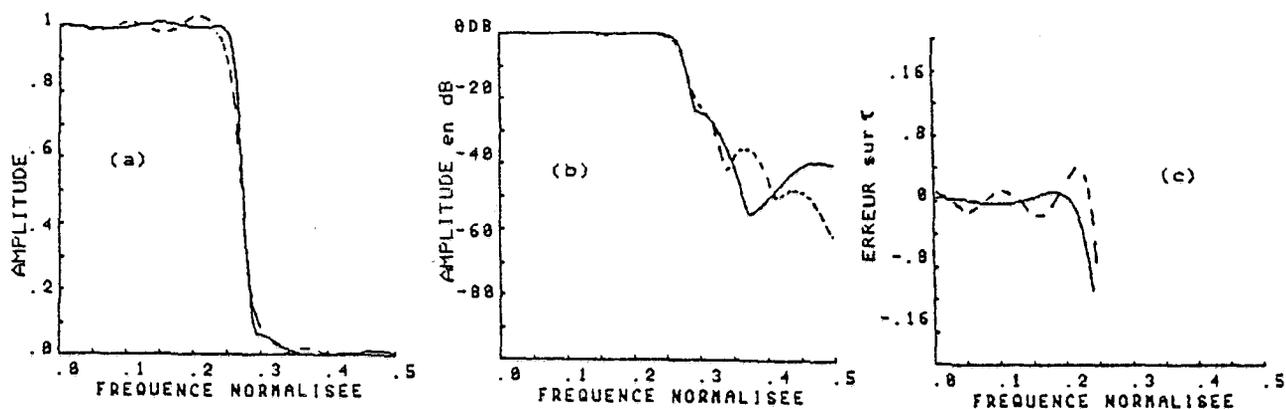


Fig. 4. — Comparaison des caractéristiques du filtre d'ordre 12 ($\tau=9$) [1] en pointillé avec le filtre proposé d'ordre 9 ($\tau=5$). Les figures 4a, b, c représentent respectivement les caractéristiques d'amplitude, d'amplitude en decibels et l'erreur sur le retard de groupe. Bien que d'ordre moins élevé, les performances du filtre proposé sont meilleurs sauf dans la bande d'atténuation où il y a équivalence.

Fig. 4. — Comparison with the 12th order filter ($\tau=9$) [1] (dotted line), with the proposed 9th order filter ($\tau=5$). Figures 4a, b, c respectively represent the characteristics of the magnitude, the magnitude in decibels and the error in group delay. The proposed lower ordered filter has better performances except in the stopband (Fig. 4b) where there is equivalence.

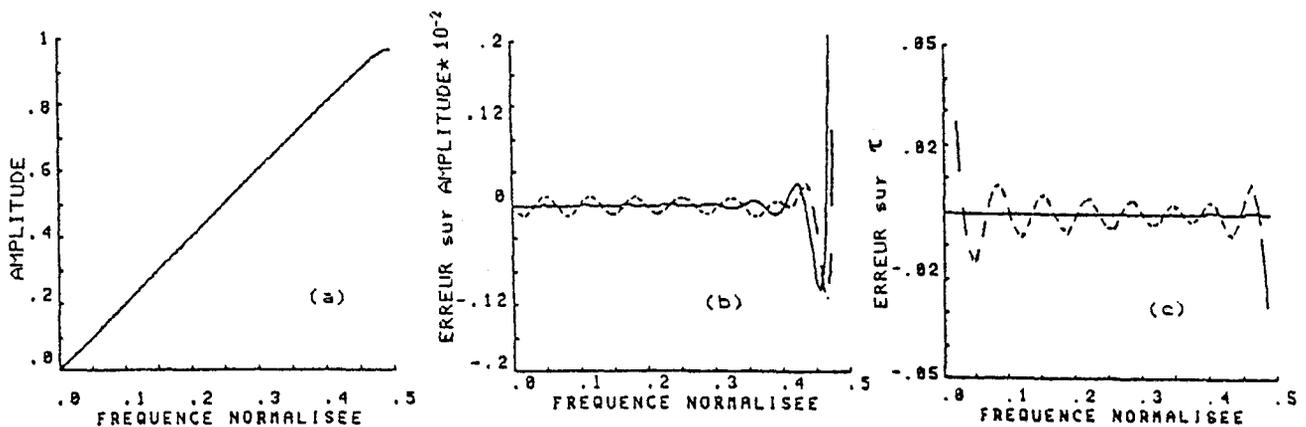


Fig. 5. — Comparaison du différentiateur d'ordre 17 [1] représenté en pointillé avec celui d'ordre 15 proposé. Les figures 5a, b, c représentent respectivement la caractéristique d'amplitude, l'erreur sur l'amplitude et l'erreur sur le retard de groupe. L'erreur sur l'amplitude (Fig. 5b) est plus petite sauf au voisinage de la fréquence $f = .5$ où il y a équivalence. L'erreur sur le retard de groupe (fig. 5c) est divisée par un rapport 50.

Fig. 5. — Comparison with the 17th differentiator [1] (dotted line) and the proposed 15th one. Figures 5a, b, c respectively represent the characteristic of the magnitude, the error in magnitude and in groupe delay. The error in magnitude (Fig. 5b) is smaller except near $f = .5$, where there is equivalence. The error in group (Fig. 5c) is 50 times smaller.

6. Discussion

Comparativement aux filtres proposés par Chottera et Julien et corrélativement à ceux d'autres auteurs, les filtres que nous présentons sont plus performants pour un ordre moins élevé. L'amélioration obtenue est due :

- d'une part à l'extension du domaine de recherche des filtres : en effet, appelons D le domaine convexe défini par les contraintes (11)-(14) du programme linéaire et par D_s le domaine des filtres stables définis dans D . L'utilisation de contraintes linéaires suffisantes et non nécessaires a pour effet de restreindre le domaine D_s des filtres stables admissibles. Dans ce cas l'espace de recherche D_s , défini par les conditions (11)-(14) du programme linéaire et les contraintes de stabilité est tel que :

$$D_s \subset D_s \subset D$$

Chottera et Julien effectuent leur recherche dans D_s tandis que nous effectuons la notre dans D en vérifiant la stabilité *a posteriori* dans D_s (ceci ne permet pas cependant d'affirmer que l'optimum absolu est toujours atteint : ceci nécessiterait des développements théoriques complexes qui ne sont pas le but de cet article);

- d'autre part à la diminution de l'erreur de troncature consécutive à la réduction de la quantité de calcul nécessaire pour l'obtention des coefficients des filtres.

Les autres conséquences dues à la nouvelle démarche, pour résoudre le problème de stabilité qui consiste à éliminer les contraintes linéaires (15) sont les suivantes :

- réduction de la matrice de contraintes du problème linéaire (de $5N_e$ colonnes, $2N+1$ lignes elle passe à $4N_e$ colonnes, $2N+1$ lignes);
- possibilité de faire un échantillonnage à pas variable (les contraintes de stabilité linéaires imposent un

échantillonnage équidistant). Ceci a permis encore de compresser les données. Rappelons que les auteurs [1] utilisent pour la conception de leurs filtres $N_e = 81$ points d'échantillonnages équidistants contre un N_e allant de 16 à 30 pour ceux que nous avons présentés. En général, le volume de données à calculer et à mettre en mémoire, au départ, a été divisé par un facteur compris entre 4 et 7;

- réduction du nombre de sommets du polyèdre convexe des contraintes. Le nombre moyen d'itérations à effectuer par le simplexe est plus petit, d'où convergence plus rapide de l'algorithme proposé.

7. Conclusion

La méthode que nous proposons utilise le simplexe et traite le problème de la stabilité en dehors du problème d'optimisation. Nos résultats comparés à ceux de la littérature sur le sujet — que ce soit des méthodes de programmation linéaire de Chottera et Julien ou de programmation non linéaire de Inukai — montrent que notre approche donne de meilleurs résultats expérimentaux. Les avantages se situent à plusieurs niveaux :

- Complexité numérique : gain d'espace et de temps-calcul, erreur de troncature réduite, vitesse de convergence de l'algorithme plus grande.

- Performance : les filtres obtenus présentent, comparativement à ceux de [1] des distorsions plus petites pour des ordres moins élevés et sont plus rapides.

- Souplesse : Possibilité d'utiliser un échantillonnage à pas variable.

Dans le cadre d'une future recherche, l'inclusion d'une fonction poids permettra de faire une meilleure approximation dans la région fréquentielle désirée, ce qui

donnera encore plus de souplesse à la méthode, du côté de son utilisation.

Manuscrit reçu le 15 mai 1987.

BIBLIOGRAPHIE

- [1] A. T. CHOTTERA et G. A. JULIEN, A Linear Approach to recursive digital filter design with linear phase, *IEEE Transaction on Circuits and Systems*, CAS 29, n° 3, March 1982, p. 139-149.
- [2] A. T. CHOTTERA et G. A. JULIEN, Design of two dimensional Recursive Digital Filters Using Linear Programming, *IEEE Transactions on Circuits and Systems*, CAS 29, n° 12, December 1982, p. 817-826.
- [3] G. CORTELLAZZO et M. R. LIGHTNER, The Use of Multiple Criterion Optimisation for simultaneous phase and magnitude design of IIR digital filters, *IEEE Transactions on Circuits and Systems*, CAS, n° 28, March 1981, p. 169-179.
- [4] G. CORTELLAZZO et M. R. LIGHTNER, Simultaneous design in both magnitude and group delay of IIR and FIR Filters Based on Multiple Criterion Optimisation, *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP 32, n° 5, October 1984, p. 107-122.
- [5] T. INUKAI, A Unified Approach to Optimal Recursive Digital Filter Design, *IEEE Transactions on Circuits and Systems*, 27, n° 7, July 1980, p. 646-649.
- [6] Y. BISTRITZ, A Stability New Test for linear Discret Systems in a Table Form, *IEEE Transactions on Circuits and Systems*, CAS. 30, n° 12, December 1983, p. 917-919.
- [7] A. G. DECZKY, Synthesis of Recursive Digital Filters Using the Minimum *p*-Error Criterion, *IEEE Transactions on Audio and Electroacoustic*, AU, n° 4, October 1972.
- [8] F. J. BROPHY et A. C. SALAZAR, Two Techniques for Digital Phase Networks, *The Bell Systems Technical Journal*, 54, n° 4, April 1975, p. 767-781.
- [9] L. R. RABINER et B. GOLD, *Theory and application of digital signal processing*, Englewood Cliffs, NJ, Prentice Hall.
- [10] E. A. ROBINSON, *Statistical Communication and Detection*, New York/Hafner, 1967.
- [11] A. G. DECZKY, Equiripple and Minimum Chebyshev Approximations for Recursive Digital Filters, *IEEE Trans. Acoustics, Speech, Signal processing*, ASSP-22, Apr. 1974, p. 98-111.
- [12] P. CHRETIENNE, Y. PRESQUIEUX et J. C. GRANDJEAN, *Agorithmes et pratiques de la programmation linéaire*, Collection Langages et Algorithmes de l'Informatique, Technip, 1980.
- [13] G. DANTZIG, *Linear Programming and Extensions*, Princeton University Press, Princeton, NJ, 1963.
- [14] R. FAURE, *Éléments de la recherche opérationnelle*, Gauthier-Villars, 1971.