De nouveaux résultats sur la synthèse de filtres RIF

Nicolas BRISEBARRE, Silviu-Ioan FILIP, Guillaume HANROT

ÉNS Lyon, LIP, E.P.I. AriC
46, Allée d'Italie F-69364 Lyon Cedex 07
nicolas.brisebarre@ens-lyon.fr, silviuioan.filip@ens-lyon.fr,
guillaume.hanrot@ens-lyon.fr

Résumé – Nous présentons un travail en cours sur la synthèse de filtres à réponse impulsionnelle finie (RIF). Après avoir mentionné notre nouvelle implantation de l'algorithme de Parks-McClellan, nous introduisons des outils de théorie algorithmique des nombres, les réseaux euclidiens, afin de faire progresser la question de la détermination (quasi-)optimale des coefficients en virgule fixe du filtre, à l'instar de ce que proposait [3] pour l'approximation polynomiale.

Abstract – We give a brief overview of the Parks-McClellan filter design algorithm and introduce an approach for designing quasioptimal fixed-point coefficient FIR filters using euclidean lattices, which has been previously used to design (quasi-)optimal polynomial approximations for mathematical functions [3].

1 Introduction

Le problème de *quantification* (quantization en anglais) des coefficients d'un filtre numérique a été étudié depuis 1973, au moins, quand Rabiner et Chan [4] ont examiné les propriétes statistiques de l'erreur de la réponse fréquentielle du filtre lorsque ses coefficients sont arrondis. Leur travail, pour ce qui concerne la quantification en arithmétique virgule fixe, a été affiné ultérieurement dans [11] et constitue depuis la manière standard de synthétiser un filtre à réponse impulsionnelle finie (RIF ou FIR en anglais) dans Matlab [1]. D'autres approches, s'appuyant sur la programmation linéaire entière (PLE) ont été introduites par Kodek [8, 9, 7]. Bien que ces travaux fournissent, en théorie, des moyens d'obtenir une quantification optimale, leur complexité exponentielle en limite la portée à des filtres avec un assez petit nombre de coefficients (< 60, disons) de taille en bits modérées (< 15).

Nous présentons ici brièvement un travail en cours qui traite de ce problème de quantification pour les filtres RIF. La section 2 a pour objet la synthèse de filtres RIF à coefficients réels en utilisant un algorithme célèbre dû à James McClellan et Thomas Parks [12]. L'information fournie par cette routine est utilisée comme entrée pour le nouveau procédé de quantification, introduit en section 3. Nous présentons quelques exemples et comparaisons en section 4 avant de conclure et de mentionner des travaux à venir dans la dernière section.

2 Synthèse de filtre RIF minimax

De multiples questions de synthèse de filtres RIF à phase linéaire peuvent être traitées via le calcul des coefficients réels *optimaux* d'une réponse fréquentielle de la forme $H_d(\omega) = \sum_{k=0}^{n} h_k \cos(k\omega)$, qui minimise une certaine mesure d'erreur. Le problème fondamental est le suivant :

Problème 1 (Équioscillation)**.** *Soit Ω un sous-ensemble fermé de $[0, \pi]$ et $D(\omega)$ une réponse fréquentielle idéale, continue sur Ω. Étant donné un **degré de filtre** $n \in \mathbb{N}$, déterminer $H_d(\omega) = \sum_{k=0}^{n} h_k \cos(\omega k)$ telle que la fonction d'erreur pondérée $E(\omega) = W(\omega)\,(D(\omega) - H_d(\omega))$ ait une norme uniforme **minimale***

$$\|E(\omega)\|_{\infty,\Omega} = \sup_{\omega\in\Omega} |E(\omega)|\,,$$

où W désigne la fonction de poids continue et strictement positive sur Ω.

La solution est caractérisée par le résultat suivant :

Théoreme 1 (Théorème d'alternance)**.** *Dans le contexte du problème 1, une condition nécessaire et suffisante pour que $H_d(\omega)$ soit l'**unique** réponse fréquentielle de degré au plus n qui minimise l'erreur pondérée d'approximation $\delta = \|E(\omega)\|_{\infty,\Omega}$ est que $E(\omega)$ possède **au moins** $n+2$ fréquences extrêmales équioscillantes sur Ω ; i.e. il existe au moins $n+2$ valeurs ω_k appartenant à Ω telles que $\omega_0 < \omega_1 < \ldots < \omega_{n+1}$ et*

$$E(\omega_k) = -E(\omega_{k+1}) = \lambda(-1)^k\delta, \qquad k = 0, \ldots, n,$$

où $\lambda \in \{\pm 1\}$ est fixé.

Démonstration. Voir [5, Ch. 3.4]. □

Un exemple typique de spécification de filtre associé au problème 1 est donné dans la figure 1. Elle décrit un filtre passe-haut où $\Omega = [0, 0.4\pi] \cup [0.6\pi, \pi]$,

$$D(\omega) = \begin{cases} 1, & 0 \leqslant \omega \leqslant 0.4\pi, \\ 0, & 0.6\pi \leqslant \omega \leqslant \pi, \end{cases} \text{ et } W(\omega) = 1, \text{ avec } \omega \in \Omega.$$

1. http://fr.mathworks.com/help/dsp/examples/optimized-fixed-point-fir-filters.html

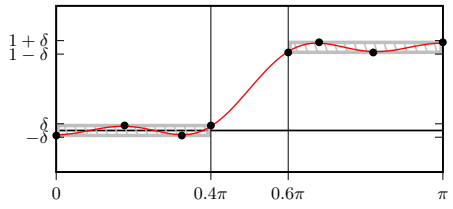

FIGURE 1 – Filtre RIF passe-haut de degré $n = 6$ satisfaisant les conditions du théorème 1.

Les $n + 2 = 8$ extrêma équioscillants, dont l'existence est annoncée par le théorème 1 sont clairement visibles.

L'algorithme d'échange de Parks-McClellan propose un procédé itératif qui converge vers cette solution optimale. Il fait partie de manière essentielle de *tout* module de traitement du signal de référence et s'appuie sur le travail de Remez [13].

Nous avons développé notre propre implantation de cet algorithme [6]. Une des raisons initiales de ce travail était que plusieurs informations produites lors de l'éxécution de cette routine sont utiles à notre méthode de quantification et elles n'étaient pas fournies par les implantations existantes. Notre implantation s'est en fait révélée plus robuste et d'une capacité de passage à l'échelle supérieure à celle de ses concurrentes.

3 Réseaux euclidiens et quantification

L'algorithme de Parks-McClellan produit en sortie un filtre minimax $H_d(\omega)$, où les coefficients h_k sont des nombres réels. Arrondir lesdites valeurs des h_k pour tenir compte des contraintes de précision imposées par une implémentation peut conduire à une perte de précision significative, tout particulièrement dans un format "virgule fixe", fréquent dans les applications en traitement du signal. Dans la suite, nous présentons une méthode heuristique, reposant sur des techniques de théorie algorithmique des nombres (les réseaux euclidiens), pour améliorer l'erreur commise lors du processus d'arrondi.

3.1 Réseaux euclidiens

Informellement, on peut voir un tel réseau comme une grille de points régulièrement disposés dans l'espace m-dimensionnel ; plus formellement, on a :

Définition 1 (Réseau euclidien)**.** *Soient $b_1, b_2, \ldots, b_n$ des vecteurs linéairement indépendants de $\mathbb{R}^m, m \geqslant n$. Le réseau engendré par $(b_1, \ldots, b_n)$ est défini par :*

$$\Gamma((b_i)_{1\leqslant i\leqslant n}) = \left\{ \sum_{i=1}^{n} x_i b_i, x_i \in \mathbb{Z} \right\}.$$

La famille de vecteurs $(b_1, \ldots, b_n)$ est une **base** *du réseau.*

De façon équivalente, si l'on définit B comme la matrice $m \times n$ dont les colonnes sont les $(b_i)_{1\leq i\leq n}$, le réseau engendré par B est $\Gamma(B) = \Gamma((b_i)_{1\leqslant i\leqslant n}) = \{Bx | x \in \mathbb{Z}^n\} = B\mathbb{Z}^n$.

Il faut noter qu'un réseau de dimension $n \geq 2$ a une infinité de bases ; de façon générale, si B est une base, alors pour tout $U \in \mathrm{GL}_n(\mathbb{Z})$, BU est aussi une base. Certaines de ces bases sont plus adaptées que d'autres pour décrire la géométrie (mais aussi pour résoudre des problèmes algorithmiques) du réseau.

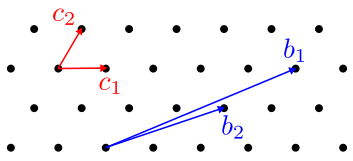

FIGURE 2 – Deux bases d'un même réseau.

Prenons par exemple la figure 2. Elle montre deux couples $(b_1, b_2)^T$ et $(c_1, c_2)^T$ qui engendrent le même réseau de dimension 2. Il est raisonnable de défendre le fait que la base (c_i) donne une meilleure vision que la base (b_i) pour explorer le voisinage d'un point du réseau donné. La notion de distance/longueur que nous utilisons ici est liée à la norme euclidienne (ou encore ℓ_2) définie par $\|x\|_2 = \sqrt{\sum x_i^2}$; nous la noterons plus simplement dans la suite $\|x\|$.

Trouver une base constituée de vecteurs courts est un problème très voisin du problème de trouver un vecteur non nul le plus court du réseau, problème appelé *problème du vecteur le plus court* (SVP). Disposer d'une base constituée de vecteurs courts simplifie également la résolution du *problème du vecteur le plus proche* (CVP), qui dans sa forme la plus simple demande de trouver un vecteur t du réseau qui minimise la distance à un vecteur $v \in \mathbb{R}^m$ donné en entrée, cf. figure 3.

SVP et CVP sont des problèmes calculatoirement difficiles [1]. Tous les algorithmes pour les résoudre ont une complexité exponentielle en la dimension, tout comme les algorithmes pour vérifier une solution candidate. En pratique, on est donc souvent amené à se contenter d'algorithmes approchés (mais de complexité polynomiale), tel l'algorithme LLL [10] pour le problème SVP, ou LLL complété par l'algorithme de Babai [2] pour le problème CVP.

3.2 L'algorithme de quantification

L'idée qui sous-tend notre algorithme est de chercher une approximation $H_d^\star(\omega)$ pour le filtre qui remplisse les conditions de précision et se comporte de façon proche du filtre optimal $H_d(\omega)$ à coefficients réels construit par l'algorithme de Parks-

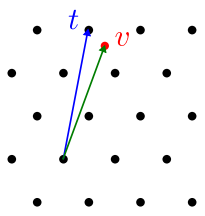

FIGURE 3 – Le problème du vecteur le plus proche (CVP) : trouver $t \in \Gamma(B)$ minimisant $\|v - t\|$.

McClellan. Bien que ce procédé reste heuristique, nos tests initiaux sur des filtres passe-bas sont prometteurs. En cas d'échec il reste possible d'utiliser le filtre obtenu par arrondi naïf.

L'algorithme se décompose en quatre étapes :

1. Utiliser l'algorithme de Parks-McClellan pour construire le filtre optimal de degré n à coefficients réels $H_d(\omega)$.
2. Réécrire le problème de quantification pour le transformer à un problème à coefficients entiers.

Dans la mesure où nous cherchons des coefficients dans un format virgule fixe, nous pouvons les chercher sous la forme $h_i = m_i/2^{e_i}$, où $m_i, e_i \in \mathbb{Z}$. Les exposants e_i sont des constantes fixées et connues liées au format et à la position de la virgule préalablement choisie pour chaque coefficient, il nous suffit donc de déterminer la valeur des mantisses $(m_k)_{0\leqslant k\leqslant n}$.

3. Choisir $n+1$ valeurs $(\omega_k^\star)_{0\leqslant k\leqslant n}$ dans Ω.

Nous construisons maintenant une instance du problème CVP qui nous permet d'encoder notre problème de quantification ; cette instance est de la forme suivante :

$$m_0 \underbrace{\begin{pmatrix} \frac{1}{2^{e_0}} \\ \vdots \\ \frac{1}{2^{e_0}} \end{pmatrix}}_{b_0} + \cdots + m_n \underbrace{\begin{pmatrix} \frac{\cos(n\omega_0^\star)}{2^{e_n}} \\ \vdots \\ \frac{\cos(n\omega_n^\star)}{2^{e_n}} \end{pmatrix}}_{b_n} \sim \underbrace{\begin{pmatrix} H_d(\omega_0^\star) \\ \vdots \\ H_d(\omega_n^\star) \end{pmatrix}}_{v}$$

La principale question concernant cette étape est le choix des $n+1$ fréquences $\omega_i^\star$. L'heuristique que nous choisissons est la suivante : si l'on regarde la réponse en fréquence de notre filtre, nous aimerions que ses zéros soient aussi proches que possible de ceux du filtre optimal $H_d(\omega)$; nous forçons ceci en choisissant comme points $\omega_i^\star$ les zéros de la fonction d'erreur associée au filtre réel optimal, qui sont calculés lors de la dernière itération de l'algorithme de Parks-McClellan. Cette idée rapproche notre problème d'une question d'interpolation : trouver une fonction à coefficients entiers prenant une valeur proche de 0 en les points $\omega_i^\star$ – que nous appelerons dorénavant de ce fait « points d'interpolation ».

D'autres choix seraient possibles pour ces $n+1$ points, comme 1) choisir les points d'interpolation équidistants dans les bandes de passage et de coupure (à l'instar de la première étape de l'algorithme de Parks-McClellan) ou 2) les choisir parmi les extrêma de la fonction d'erreur associée au filtre réel optimal.

4. Résoudre le problème de vecteur le plus proche correspondant dans le réseau euclidien $\Gamma((b_i)_{0\leqslant i\leqslant n})$.

Cela revient à trouver les coefficients $(m_0, m_1, \ldots m_n) \in \mathbb{Z}^{n+1}$ qui minimisent la quantité $\left\|\sum_{k=0}^{n} m_k b_k - v\right\|_2$, c'est-à-dire que l'on cherche un vecteur dans le réseau engendré par la base $B = (b_0, b_1, \ldots, b_n) \in \mathbb{Z}^{(n+1)\times(n+1)}$ qui, idéalement, est le plus proche de v. Nous traitons ce problème de manière approchée en utilisant l'algorithme de Babai [2].

Il est important de bien insister sur le fait que notre algorithme est fondé sur plusieurs heuristiques. Nous pointons en particulier trois d'entre elles :

a. nous transformons le problème de recherche de filtre optimal en un problème discret via le choix de $n+1$ points d'interpolation ;
b. nous remplaçons le problème initial d'approximation en norme infinie en un problème d'approximation en norme L^2. Bien que la solution que nous obtenons soit généralement de bonne qualité en norme infinie, il n'y a aucune raison qu'elle soit optimale ;
c. l'utilisation de l'algorithme LLL combiné avec l'algorithme de Babai produit un résultat en norme L^2 qui est proche de la vraie solution du problème CVP, mais n'est pas nécessairement la solution exacte.

En dépit de ces heuristiques agressives, nos tests sur des filtres passe-bas RIF sont encourageants.

TABLE 1 – Spécification des deux familles de filtres

Filter Type	A		B	
Bands	$\left[0, \frac{2\pi}{5}\right]$	$\left[\frac{\pi}{2}, \pi\right]$	$\left[0, \frac{2\pi}{5}\right]$	$\left[\frac{\pi}{2}, \pi\right]$
$D(\omega)$	1	0	1	0
$W(\omega)$	1	1	1	10

TABLE 2 – Comparaison des erreurs d'approximation

Filtre	E^*	Arrondi au plus proche $\overline{E}^*$	Notre méthode E_{LLL}	${E_{FP}}^*$ optimale
A17/8	0.01594584	0.03267182	0.03267182	0.02983816
A22/8	0.00712762	0.03705975	0.03188672	0.02962304
A62/22	0.00000797	0.00001717	0.00001302	0.00001077†
B17/9	0.05271937	0.15903877	0.11718750	0.07709547
B22/9	0.02104800	0.11718750	0.07786062	0.05679037
B62/22	0.00002489	0.00005245	0.00003814	0.00002959†

4 Résultats

Nous avons implanté les algorithmes décrits ci-dessus pour le calcul de filtres RIF optimaux et l'algorithme de quantification en C/C++, en utilisant pour les filtres de grande longueur les bibliothèques multiprécision entière `GMP` [2] et flottante `MPFR` [3]. Nous avons également utilisé l'implémentation de l'algorithme LLL de la bibliothèque `fplll` [4]. Le programme correspondant a été testé sur des environnements Linux et Windows avec une version récente (4.8.2) du compilateur `g++`.

Dans ce qui suit, les familles de filtres décrites dans la table 1 sont utilisées pour évaluer nos algorithmes. Ainsi, B est un filtre passe-bas avec un poids unitaire dans la bande de passage et un poids de 10 dans la bande de coupure.

L'intérêt de notre algorithme heuristique de quantification réside dans sa capacité à calculer des filtres RIF de bonne qualité en un temps de calcul polynomial en le degré (en raison des propriétés de l'algorithme LLL), tandis que les solveurs optimaux [7] restent à ce jour exponentiels.

Dans le tableau 2, nous avons noté E^* l'erreur d'approximation (définie dans la partie 2) quand aucune restriction sur la précision des coefficients n'est prise en compte, $\overline{E}^*$ l'erreur obtenue pour le filtre obtenu en arrondissant le filtre optimal au plus proche, E_{LLL} l'erreur obtenue en utilisant notre méthode

2. téléchargeable sur http://gmplib.org/
3. téléchargeable sur http://www.mpfr.org/
4. téléchargeable sur https://github.com/dstehle/fplll

TABLE 3 – Temps de calcul (secondes)

Filtre	Notre approche	Méthode optimale ([7])
A22/9	0.13	0.23
A27/10	0.21	8.91
B22/10	0.14	1.67
B27/10	0.22	5.40

et $E_{FP}{}^*$ la meilleure erreur de quantification possible (calculée à l'aide de PLE). Le symbole † signifie que pour ces problèmes-là, la routine de PLE n'a pu terminer en un temps raisonnable et les résultats alors fournis correspondent à la meilleure valeur obtenue au moment de l'interruption du programme. La notation retenue pour le filtre (par exemple A17/8) se comprend comme : la famille de filtres choisie, le degré n et le nombre de bits par coefficient (en virgule fixe).

La table 2 montre que notre méthode égale ou améliore l'arrondi naïf dans tous les cas. Même si notre méthode n'offre à ce jour aucune garantie d'optimalité, on constate néanmoins qu'on s'approche de l'optimum, avec l'avantage supplémentaire d'une méthode nettement plus efficace par rapport aux solveurs utilisant de la PLE (ce serait encore plus frappant en augmentant la dimension, mais les solveurs PLE ne passent alors pas à l'échelle) – une comparaison des temps de calculs est donnée dans la table 3 ; cette comparaison s'est effectuée sur des machines de vitesse d'horloge identique.

5 Conclusion

Dans ce travail, nous avons présenté de premiers résultats sur un nouvel algorithme permettant de déterminer des filtres RIF mieux quantifiés, reposant sur des techniques de théorie algorithmique des nombres. Ces résultats nous semblent très encourageants : un algorithme qui améliore la méthode naïve tout en étant plus rapide que la recherche de l'optimum. Nous pensons que ces premiers résultats soulèvent aussi des questions que nous discutons brièvement dans les paragraphes qui suivent.

L'aspect restant le plus mal compris de notre algorithme de quantification est le choix des points d'interpolation. Aucun des trois choix que nous avons suggéré pour ces points ne semble être le meilleur dans tous les cas pour les exemples que nous avons testé, même si le choix mis en avant (les zéros de la fonction d'erreur optimale) est souvent le meilleur. Étudier une formalisation basée sur d'autres techniques d'interpolation (telle l'interpolation de Hermite, une interpolation des valeurs et des dérivées en $(n+1)/2$ points) serait intéressant. Des expériences ont également montré qu'ajouter des poids aux lignes de la matrice réduite par LLL (ce qui revient à donner plus d'importance à certains points d'interpolation qu'à d'autres) améliore dans de nombreux cas la qualité du filtre. On peut également penser à améliorer la qualité de la troisième heuristique discutée plus haut en remplaçant LLL par un algorithme approchant mieux CVP (tel BKZ), la contrepartie étant un temps de calcul plus important. Il faudrait enfin étudier l'apport de méthodes combinant réseaux et programmation linéaire.

La recherche de filtres RII est également très importante en traitement du signal numérique, et nous souhaiterions étendre l'approche fondée sur des outils d'arithmétique des ordinateurs et théorie algorithmique des nombres présentée ici à ce contexte. Cette adaptation présente de vraies difficultés supplémentaires ; ainsi pour un tel filtre, la fonction de transfert est une fonction rationnelle et la quantification doit également prendre grand soin de perturber le moins possible la localisation des zéros et des pôles (complexes) de la fonction de transfert optimale.

Enfin, nous souhaiterions développer un outil logiciel exploitant le travail présenté ici pour générer du code VHDL synthétisant la fonction de transfert du filtre quantifié sur une carte FPGA. Il peut être pertinent également de développer des outils et des méthodes pour certifier (au sens des outils de preuve formelle) la qualité des filtres produits par notre algorithme.

Références

[1] M. Ajtai. The shortest vector problem in L_2 is NP-hard for randomized reductions (extended abstract). In *STOC*, 10–19, 1998.

[2] L. Babai. On Lovász' lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1) :1–13, 1986.

[3] N. Brisebarre and S. Chevillard. Efficient polynomial L^∞ approximations. In *Proceedings of the 18th IEEE Symposium on Computer Arithmetic*, pages 169–176, 2007.

[4] D. Chan and L. Rabiner. Analysis of quantization errors in the direct form for finite impulse response digital filters. *Audio and Electroacoustics, IEEE Trans. on*, 21(4) :354–366, 1973.

[5] E.W. Cheney. *Introduction to Approximation Theory*. AMS Chelsea Publishing Series. AMS Chelsea Publishing, 1982.

[6] S. Filip. A robust and scalable implementation of the Parks-McClellan algorithm for designing FIR filters. submitted, available at https://hal.inria.fr/hal-01136005, 2015.

[7] D. Kodek. LLL Algorithm and the Optimal Finite Wordlength FIR Design. *Signal Proc., IEEE Trans. on*, 60(3) :1493–1498, 2012.

[8] D. Kodek and K. Steiglitz. Comparison of optimal and local search methods for designing finite wordlength FIR digital filters. *Circuits and Systems, IEEE Trans. on*, 28(1) :28–32, 1981.

[9] D. Kodek. Performance limit of finite wordlength FIR digital filters. *Signal Proc., IEEE Trans. on*, 53(7) :2462–2469, 2005.

[10] A.K. Lenstra, Jr. Lenstra, H.W., and L. Lovász. Factoring polynomials with rational coefficients. *Math. Ann.*, 261(4) :515–534, 1982.

[11] J.J. Nielsen. Design of linear-phase direct-form FIR digital filters with quantized coefficients using error spectrum shaping. *Acoustics, Speech and Signal Proc., IEEE Trans. on*, 37(7) :1020–1026, 1989.

[12] T. Parks and J. McClellan. Chebyshev Approximation for Nonrecursive Digital Filters with Linear Phase. *Circuit Theory, IEEE Trans. on*, 19(2) :189–194, 1972.

[13] E. Remes. Sur le calcul effectif des polynomes d'approximation de Tchebichef. *C.R.A.S.*. 199, 337–340, 1934.