الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

10/91

وزارة الجامعات
Ministère aux Universites

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT ELECTRONIQUE

PROJET DE FIN D'ETUDES

SUJET

SIMULATION DES FILTRES NUMERIQUES R. I. I. EN VIRGULE FIXE

Proposé par : B. DERRAS **Etudié par :** N. SADI AHMED **Dirigé par :** B. DERRAS

PROMOTION

Juin 1991

E.N.P. 10, Avenue Hacen Badi - El-Harrach - Alger

Ecole Nationale Polytechnique

DEPARTEMENT D'ELECTRONIQUE

PROJET DE FIN D'ETUDE

SIMULATION DES FILTRES NUMERIQUES RII EN VIRGULE FIXE

PROPOSEE ET DIRIGE PAR : M.B.DERRAS

ETUDIE PAR : SADI AHMED NAFISSA

PROMOTION JUIN 1991

Mes Remerciements . . .

les plus chaleureux sont adressés à Mr B.DERRAS , mon promoteur, pour avoir fait preuve de générosité et de compréhension à mon égard tout au long de cette année . Je tiens aussi à lui exprimer ma profonde gratitude pour le soutien et l'aide permanents dont il m'a fait part afin de mener à bien ce présent projet. De même, je saisie l'opportunité de ces quelques lignes pour lui témoigner mon immense respect.

Mes plus vifs remerciement s'adressent à M^{elle} Djamila, ingenieur au centre de calcul de l'Ecole pour son aide précieuse .

إهداء

إلى الحبيبة أمي ... إلى الغالي أبي ...

إلى جزيرة و هناء و صلاح الدين ...

إلى كل من أحبهم ...

إلى كل قارئ شغوف يقصد الإستفادة بما تحمله هذه الصفحات

و أتمنى أن يصيب مبتغاه و مراده ...

نفيسة

ملخص المشروع

الغاية الأساسية من هذه الدراسة هي التمكن من مضاهاة عمل المرشحات العددية ذات الإستجابة الدافعية اللا متناهية في الزمن ، بهدف إظهار مدى تأثر عملية الترشيح من جراء إستعمال سجلا ت محدودة الطول لحفظ المعطيات و إنجاز العمليات الحسابية ، و بخاصة في حالة التمثيل القيمي بالفاصلة الكسرية الثابتة. كما تمكن هذه الدراسة من التحقق من صلا حية الفرضيات المستعملة في نظرية البنيات الفضلى عن طريق النتائج العملية ، بالإ ضافة إلى دراسة بنيتين ناتجتين عن تحليل المرشح إلى خلا يا من المرتبة الثانية موصولة على التوازي ، و التي تشكل حلا وسطا بين البنيات الفضلى و المباشرة الإ جمالية ، من حيث قيمة الخطأ الحسابي و بساطة الأ جهزة الضرورية لتحقيقها . و قد يفسح هذا التصنيف مجالا أوسع لإ ختيار البنية المناسبة حسب المقاييس التقنية و الإ قتصادية المحددة .

PROJECT ABSTRACT

This work is devoted to the simulation of IIR digital filters in fixed point arithmetic in order to analyse the output error behavior with the respect to the finite wordlength effects.

It has been shown that a minimum output error variance can be reached by the use of the digital state-space descriptions. But, unfortunately these structures present a high hardware complexity which leads to look for other structures presenting a compromise between noise gain and complexity.

Two types of compromise structures based on second order parallel section filter decomposition are proposed in this project in the aim to permit a larger range choice of the convenient structure for specific applications.

RESUME DU PROJET

La finalité essentielle de ce projet est de simuler les filtres numériques RII en virgule fixe afin d'analyser leur comportement vis à vis des effets de la limitation de la précision de toutes les valeurs intervenant dans l'opération du filtrage et de vérifier la validité des hypothéses faites dans la théorie des structures à gain minimal.

Deux types de structures basées sur la theorie de la décomposition du filtre en cellules du second ordre disposées en paralléle, sont etudiées et simulées. L'avantage de ces deux structures par rapport aux structures optimale et canonique est le bon compromis qu'elles offrent entre la qualité du filtrage et la complèxité des circuits.

SOMMAIRE

CHAPITRE I

INTRODUCTION GENERALE

CHAPITRE I

INTRODUCTION GENERALE

1-1 Préliminaire

Il y a quelques années, la numérisation des filtres analogiques était seulement un moyen pour simuler leurs comportements sur ordinateur, dans le but de tester leurs performances et de corriger leurs imperfections avant leurs réalisations analogiques définitives. Ce moyen est très efficace et économique, car il peut en quelques sortes assurer le bon fonctionnement des filtres qui constitue un facteur capital dans la plupart des applications.

De nos jours, le grand développement de la technologie des circuits intégrés numériques offre avec la complicité de l'arsenal mathématique du traitement numérique des signaux, la possibilité de réaliser et d'utiliser des filtres proprement numériques. Ces derniers se distinguent par leur grande fiabilité, par leurs vitesses de traitement et par leurs coûts réduits.

La simulation de ces filtres avant leur implantation permet de résoudre les problèmes que peut poser leur réalisation matérielle; en plus des problèmes classiques de l'échantillonnage, le facteur de qualité, la binarisation et la représentation arithmétique,... il y a le problème de la précision avec laquelle doit s'opérer le filtrage du fait de l'utilisation de registres de tailles réelles pour représenter les valeurs des paramétres des filtres et des signaux qui les parcourent ainsi que les valeurs résultant des diverses opérations mathématiques.

L'impact de la limitation de la longueur des mots sur la qualité du filtrage, et les effets néfastes qu'elle cause, méritent une sérieuse considération et une plus grande part d'attention.

Les nombreuses investigations qui ont été engagées dans ce sens ont permis de prévaloir l'efficacité de la représentation des filtres numériques dans l'espace d'état par l'opportunité de trouver des structures ayant des caractéristiques optimales vis-à-vis des effets de la longueur finie des mots.

Cependant, ces structures optimales présentent une certaine complexité dans leurs réalisations vu le nombre d'opérations à effectuer donc du coût de la réalisation et du temps de calcul.

Ce qui a amené à chercher des structures de compromis qui ont pour objectif de réduire, simultanément la variance du bruit de calcul par rapport à celle de la structure canonique et la complexité materielle par rapport à la structure optimale .

Ce présent travail s'interésse à la simulation des filtres numériques RII en virgule fixe pour mettre en relief les conséquences de l'utilisation de registres de longueur finie sur la qualité du filtrage et pour vérifier l'applicabilité de la théorie de minimisation du gain de bruit de calcul. Les structures optimales sont approchées par deux méthodes: **MULLIS-ROBERTS** et **HWANG**, en diminuant la probabilité de dépassement de la gamme des valeurs représentables par une normalisation convenable .

Deux structures de compromis sont proposées, elles sont basées sur la décomposition du filtre en cellules du second ordre mises en parallèles dans l'une, les cellules sont optimales séparément et dans l'autre, elles sont canoniques. Les performances de telles structures du point de vue gain de bruit de calcul, nombre d'opérations, la sensibilité de la fonction de transfert ainsi que l'invariance à une transformation fréquentielle sont illustrés par la simulation d'un filtre RII prototype.

1-2 Organisation du projet

Cette étude débute par une brève présentation des outils mathématiques nécessaires pour la caractérisation et la synthèse des filtres numériques ainsi que leurs propriétés (chapitre II).

Le chapitre III est consacré à l'analyse des effets dûs à la limitation de la longueur des mots et à mettre en évidence le compromis qui existe entre le bruit de calcul et les erreurs de dépassement dans les filtres numériques RII, tout en présentant les différents types de quantification d'arithmétique et de caractéristique de dépassement.

Les chapitres V et IV sont consacrés aux méthodes et procédures d'optimisation des erreurs de calcul en sortie du filtre par des structures optimales et des structures de compromis qui sont présentées par des cellules du second ordre optimales ou canoniques connectés en parallèles.

Le chapitre VI contient les résultats de la simulation des différentes structures pour un filtre donné, ainsi que l'illustration de certains résultats théoriques.

Et enfin, vient la conclusion générale au chapitre VIII.

CHAPITRE II

GENERALITES SUR LA SYNTHESE DES FILTRES NUMERIQUES.

CHAPITRE II

GENERALITES SUR LA SYNTHESE DES FILTRES NUMERIQUES

2.1 Systèmes et signaux discrets

2.1.1 Definitions

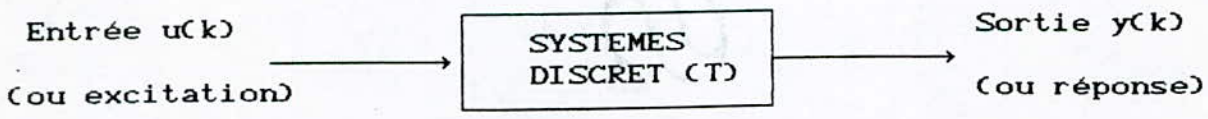
Un signal discret est une suite de valeurs numériques réelles ou complexes. Il est dit numérique si son amplitude est quantifiée. Les valeurs du signal sont, en général, régulièrement réparties dans le temps. L'intervalle de temps entre deux valeurs du signal discret est appelé "période d'echantillonnage".

Comme dans le cas continu, les signaux discrets sont classés selon deux catégories:

- Les signaux déterministes dont l'évolution en fonction de la variable temps peut être prédite parfaitement par une représentation mathématique appropriée.
- Les signaux aléatoires, par contre, ont un comportement imprévisible; ils se caractérisent par leurs propriétés statistiques et fréquentielles.

Un signal périodique dont le comportement est imprévisible sur une période et qui est generé à partir d'une relation de recurrence parfaitement deterministe, est dit "pseudoaleatoire".

Un système discret agit sur un signal discret à son entrée pour produire un autre signal discret à sa sortie.



Fig(2-1) - Schematisation d'un système discret.

Un signal discret est causal s'il ne produit de réponse à un instant donné k, que pour les excitations antérieures; celà est dû au fait que, pour les systèmes physiques, l'effet ne peut précéder la cause.

Un système discret est linéaire, s'il réalise le principe de superposition; si T est l'opérateur caractérisant l'effet du système alors:

$$T[a\,u(k) + b\,u'(k)] = a\,T[u(k)] + b\,T[u'(k)] \qquad (2\text{-}1)$$

où a et b sont des constantes et u(k) et u'(k) sont deux entrées différentes à l'instant k (entier).

Un système discret est invariant dans le temps si :

$$T[u(k)] = y(k) \Rightarrow T[u(k-ko)] = y(k-ko) \qquad (2\text{-}2)$$

où u(k) et y(k) sont respectivement l'entrée et la sortie du système T et ko un entier [1].

2.1.2 Transformée de FOURIER

La transformée de FOURIER (TF) d'un signal discret u(k) est donnée par :

$$U(f) = \sum_{k=-\infty}^{+\infty} u(k)\, e^{-j2\pi f k} \qquad (2\text{-}3)$$

où f est la fréquence.

Une condition suffisante pour que U(f) existe est :

$$\sum_{k=-\infty}^{+\infty} |u(k)| < \infty \qquad (2\text{-}4)$$

U(f) est une fonction complexe ou réelle de f et est périodique de période unité. En général, on considère U(f) sur l'intervalle principal [-1/2 ,1/2]; La transformée inverse de U(f) est alors:

$$u(k) = \int_{-1/2}^{1/2} U(f)\, e^{j2\pi f k}\, df \qquad (2\text{-}5)$$

Pour un signal u(k) réel, le spectre d'amplitude $|U(f)|$ est paire et le spectre de phase $\arg[U(f)] = \mathrm{arctg}[\mathrm{Im}(U(f))/\mathrm{Re}(U(f))]$ est impaire.

Si en plus u(k) est périodique de période N, on définit la transformée de FOURIER discrète (TFD):

$$U(n) = \sum_{k=0}^{N-1} u(k)\, e^{-2j\pi nk/N} \qquad (2\text{-}6)$$

et donc

$$u(k) = \frac{1}{N} \sum_{n=0}^{N-1} U(n)\ e^{2j\pi nk/N} \qquad (2\text{-}7)$$

Où U(n) représente le domaine fréquentiel et u(k) représente le domaine temporel [1],[2].

La TFD est très efficace dans le traitement numérique des signaux car c'est la seule transformation qui peut être programmées sur ordinateur; des algorithmes appelés transformée de FOURIER rapide (TFR) ont été développés pour accélérer et simplifier le calcul de la TFD [3].

2.1.3 Echantillonnage et recouvrement

L'echantillonnage consiste en l'observation d'un signal continu à certains instants qui sont généralement séparés par une durée constante Te appelée période d'echantillonnage. Cette discrétisation dans le temps entraîne une répétition périodique du spectre du signal continu.

Pour que cette periodicité dans le domaine frequentiel ne dèforme pas le spectre repeté (phenomène du recouvrement spectral) et permet la reconstitution du signal à partir de ses echantillons, il faut et il suffit que la fréquence d'echantillonnage (où de répitition) Fe=1/Te soit egale ou superieure au double de la fréquence maximale contenue dans le signal [4].

2.1.4 Transformée en Z

La transformation en Z est un outil plus puissant que la transformation de FOURIER qui ne représente qu'un cas particulier de celle-ci. Pour un signal u(k), on definit sa transformée en Z bilatérale comme etant:

$$Z\,[u(k)] = U(Z) = \sum_{k=-\infty}^{+\infty} u(k)\ z^{-k} \qquad (2\text{-}8)$$

Pour les systèmes et les signaux causals, on utilise la forme unilatérale:

$$U(z) = \sum_{k=0}^{+\infty} u(k)\ z^{-k} \qquad (2\text{-}9)$$

Cette transformée existe pour un ensemble de points du plan complexe Z appelè "region de convergence", et dans laquelle:

$$\sum_k |u(k)\ z^{-k}| < \infty \qquad (2\text{-}10)$$

La transformée en Z inverse est donnée par:

$$u(k) = \frac{1}{2\pi j} \oint_\Gamma U(z)\ z^{k-1} dz \qquad (2\text{-}11)$$

où Γ est un contour fermé entourant l'origine dans le sens trigonométrique positif à l'intérieur de la région de convergence.

2.1.5 Réponse impulsionnelle

Un système linéaire est caractérisé complètement dans le domaine temporel par sa réponse impulsionnelle, qui est la réponse du système à une impulsion de Kronecker donnée par:

$$\delta(k) = \begin{cases} 1 & \text{si } k=0 \\ 0 & \text{sinon} \end{cases}$$

Pour une entrée u(k), la sortie y(k) d'un système discret s'exprime par:

$$y(k) = \sum_{l=-\infty}^{+\infty} u(l)\ h(k-l) \qquad (2\text{-}12)$$

$$= (u * h)(k) = (h * u)(k)$$

où h(k) est la réponse impulsionnelle du système est * est l'opérateur de la convolution.

Pour un système causal $h(k)=0$ pour $k<0$ (2-13)

* Stabilitè des systèmes discrets

Un système linèaire discret est stable au sens de BIBO, si pour une entrée bornée, le système dèlivre une sèquence bornée à sa sortie, autrement dit la norme de h(k) est finie [1]:

$$\|h\|_1 = \sum_k |h(k)| < \infty \qquad (2\text{-}14)$$

La transformée en Z de l'équation (2-12) est:

$$Y(Z) = H(Z)\ U(Z) \qquad (2\text{-}15)$$

où $$H(Z) = \frac{Y(z)}{U(Z)} \qquad (2\text{-}16)$$

est appelèe fonction du transfert du système.

On rappelle que l'expression (2-15) est obtenue en utilisant la propriété de convolution de la transformée en Z [1],[2].

2.1.6 Equation aux différences finies

L'excitation u(k) et la réponse y(k) d'un système numérique linéaire invariant dans le temps (LIT), satisfont une equation linéaire aux différences finies à coefficients constants de la forme:

$$\sum_{n=0}^{N} a_n \, y(k-n) = \sum_{m=0}^{M} b_m \, u(k-m) \qquad (2\text{-}17)$$

ou encore

$$y(k) = \sum_{m=0}^{M} \frac{b_m}{a_0} \, u(k-m) - \sum_{n=1}^{N} \frac{a_n}{a_0} \, y(k-n) \qquad (2\text{-}18)$$

où M, N, n et m sont des entiers. N est appelè l'ordre du système.

2.1.7 Réponse fréquentielle

En passant à la transformée en Z, on peut écrire à partir de l'equation (2-17) :

$$\sum_{n=0}^{N} a_n \, Y(z) \, z^{-n} = \sum_{m=0}^{M} b_m \, U(z) \, z^{-m} \qquad (2\text{-}19)$$

La fonction de transfert du système s'exprime alors par:

$$H(z) = \frac{Y(z)}{U(z)} = \frac{\sum_{m=0}^{M} b_m \, z^{-m}}{\sum_{n=0}^{N} a_n \, z^{-n}} \qquad (2\text{-}20)$$

La réponse frequentielle du système est un cas particulier de la fonction de transfert H(Z) pour laquelle $|z|=1$ (ou $z=e^{2j\pi f}$) , elle est donnée par:

$$H(z) = \frac{\sum_{m=0}^{M} b_m \, e^{-2j\pi fm}}{\sum_{n=0}^{N} a_n \, e^{-2j\pi fn}} = G \, \frac{\prod_{i=1}^{M} (e^{2j\pi f} - z_i)}{\prod_{i=1}^{N} (e^{2j\pi f} - p_i)} \qquad (2\text{-}21)$$

où G est une constante.

H(f) caractérise complètement le système dans le domaine frequentiel:

$$y(k) = h(k) * u(k) \xrightarrow{TF} Y(f) = H(f)\,U(f) \qquad (2\text{-}22)$$

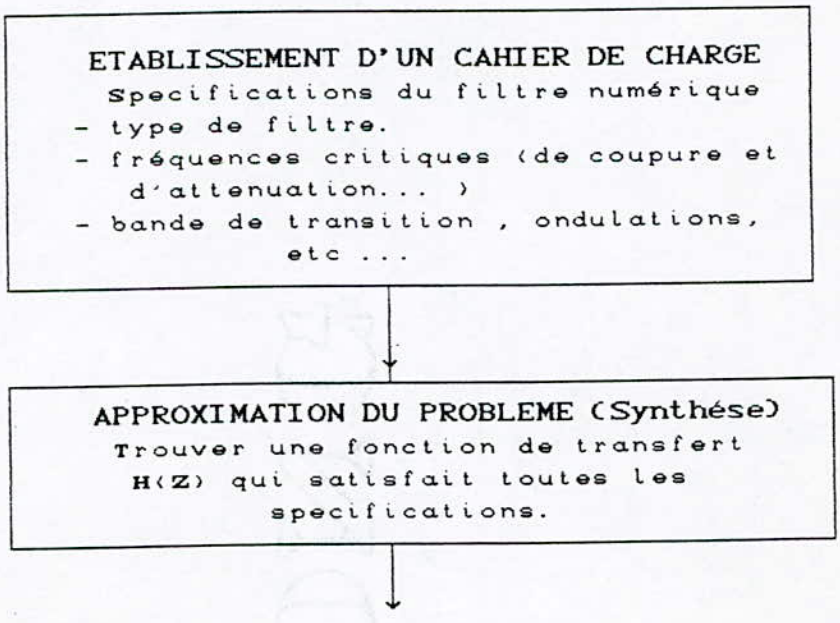
Les racines z_i du polynôme du numerateur de H(f) sont appelées les zeros du système et les racines p_i du dénominateur de H(f) sont appelées les pôles du système.

2.2 Les filtres numériques

Les filtres numériques sont des systèmes physiques donc causals qui ont pour fonction de modifier la distribution des composantes fréquentielles d'un signal selon des spécifications bien determinées, en utilisant des opérations arithmetiques de précision limitée. On s'intéressera surtout aux systèmes LIT.

Ces filtres ont été conçus pour simuler les filtres analogiques sur ordinateur. Ce qui a permis d'etudier leurs performances et d'optimiser leurs paramétres avant leurs eventuelles realisations. Mais avec le grand essor qu'a connu la technologie des circuits integrés, l'interêt des filtres numériques s'est accru et des mèthodes de synthèse propres à ce type de filtres ont étè largement developpées.

La conception des filtres numériques se fait généralement suivant le schéma ci-dessous:



MODELISATION DU PROBLEME

Simulation et mise au point
Obtenir une realisation qui definit la structure interne du filtre H(Z). Cette realisation doit être optimale vis-à-vis de la performance et des moyens actuels de calcul (ou traitement)

↓

CONCRETISATION DU PROBLEME

Determination du materiel necèssaire pour réaliser le filtre avec une complexité réduite, une performance satisfaisante et un débit de données optimal.

On distingue deux grandes classes de filtres numériques :

A- Filtres à reponse impulsionnelle de durée finie (RIF)

Où h(k) est non nulle sur un intervalle de temps N fini. Ils sont realisés de maniére non recursive suivant l'expression suivante:

$$y(k) = \sum_{n=0}^{N-1} h(n)\ u(k-n) \qquad (2\text{-}17)$$

Leur schéma fonctionnel est de la forme [2]:

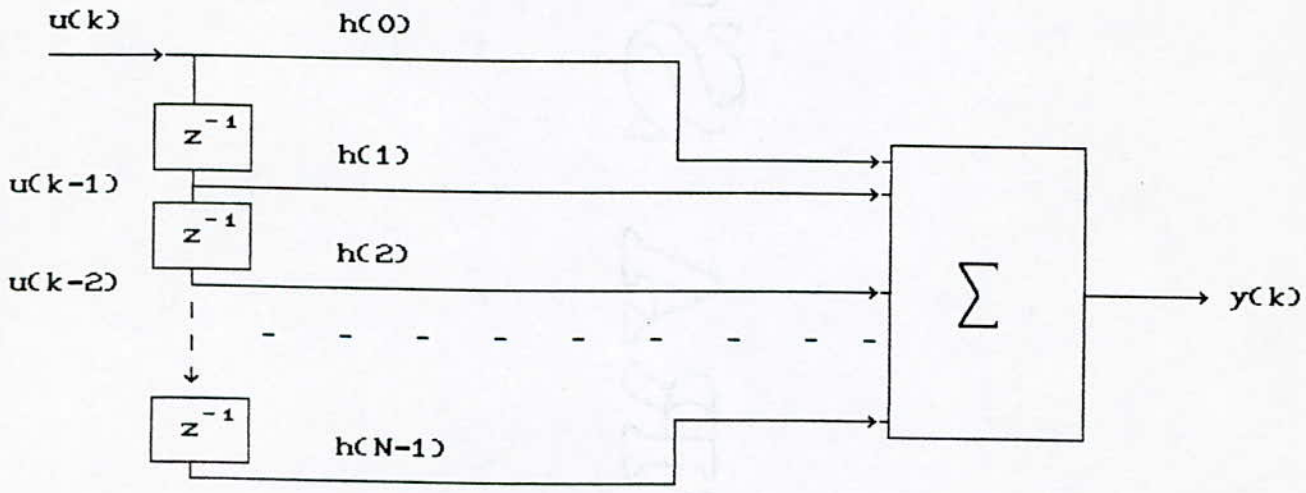


Fig (2-2) - Schéma fonctionnel d'un filtre RIF

On peut synthetiser ce type de filtres par plusieurs méthodes, parmi lesquelles :

1- Methodes du fenêtrage temporel

Soit un filtre de réponse impulsionnelle

$$h(k) = F^{-1}[H(\omega)]$$

non causale et infinie. On choisit une fenêtre w(k) qui limite la durée de h(k) dans $[k_1, k_2]$ et construit:

$$\hat{h}(k) = \begin{cases} h(k-k_2)\ w(k-k_2) & 0 \le k \le k_2 - k_1 = N-1 \\ 0 & \text{ailleurs} \end{cases} \qquad (2\text{-}24)$$

qui est ainsi causale; on determine alors le filtre RIf

$$\hat{H}(\omega) = \sum_{k=0}^{N-1} \hat{h}(k)\ e^{-j\omega k}$$

$$= e^{-j\omega(N-1)/2} \sum_{m=k_1}^{k_2} h(m)\ w(m)\ e^{-jm\omega} \qquad (2\text{-}25)$$

La fenêtre est choisie de telle sorte que:

- la linéarité de la phase soit conservée
- l'erreur

$$\varepsilon^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left| H(\omega) - \hat{H}(\omega) \right|^2 d\omega \qquad (2\text{-}26)$$

soit minimale

- Le rapport (Amplitude du lobe principale)/(Amplitude du 1er lobe secondaire) et la largeur du lobe principal (résolution de la fenêtre) soient acceptables pour conserver le maximum d'energie de la reponse frequentielle H(ω) à approximer [1].

2- Méthode d'échantillonnage en fréquence

Soit H(ω) la réponse fréquentielle spécifiée qu'on échantillonne en N points du cercle unité, puis on détermine la réponse impulsionnelle par la TFD inverse:

$$H(n) = \sum_{k=0}^{N-1} h(k)\ e^{-j2\pi nk/N} \qquad (2\text{-}27)$$

$$h(k) = \frac{1}{N} \sum_{n=0}^{N-1} H(n)\ e^{j2\pi nk/N}$$

Ayant les N échantillons de h(k), on calcule sa transformée en Z

$$H(z) = \sum_{k=0}^{N-1} h(k)\ z^{-k}$$

$$= \frac{1}{N} \sum_{n=0}^{N-1} H(n)\ \frac{1 - z^{-N}}{1 - z^{-1} e^{j2\pi n/N}} \qquad (2\text{-}28)$$

La réponse fréquentielle approximée $\hat{H}(\omega)$ est donc

$$\hat{H}(\omega) = H(z)\Big|_{z=e^{j\omega}} = \frac{1}{N}\sum_{n=0}^{N-1} H(n)\ I_N(n,\omega) \qquad (2\text{-}29)$$

où $$I_N(n,\omega) = \frac{1 - e^{-j\omega N}}{1 - e^{-j\omega}\ e^{j2\pi n/N}}$$

est appelée fonction d'interpolation [5].

Le nombre N des échantillons et les valeurs de $H(\omega)$ dans les bandes de transition seront choisis de façon à respecter les contraintes sur le niveau des ondulations et sur la largeur de la bande de transition [1].

3- Méthode du minimax

Cette méthode consiste à trouver une approximation $\hat{H}(e^{j\omega})$ de $H(\omega)$ de façon à minimiser la norme[1]:

$$\|\hat{H} - H\|_\infty = \max_\omega \ |\hat{H}(\omega) - H(e^{j\omega})| \qquad (2\text{-}30)$$

La méthode la plus classique est de trouver $H(e^{j\omega})$ sous forme d'un polynôme de CHEBYSHEV de variable:

$$x = \cos\omega \qquad (-1 \le x \le 1) \qquad (2\text{-}31)$$

et de degré au plus égal au nombre d'échantillons de h(k) tel que

si $H(\omega) = p(x=\cos\omega)$, $H(\omega) = d(x)$ avec

$$\varepsilon = \|p - d\|_\infty = \max_{-1\le x\le 1} |p(x) - d(x)|$$

alors

$$\begin{cases} p(x_i) - d(x_i) = (-1)^i \varepsilon \\ (x_i^2 - 1)\,(p'(x_i) - d'(x_i)) = 0 \end{cases} \qquad (2\text{-}32)$$

pour $-1 \le x_i < x_{i+1} \le 1$ $\qquad i = 0, 1, \ldots$

Cette méthode garantit des ondulations d'amplitudes égales (théorème de CHEBYSHEV) dans la bande passante et dans la bande d'affaiblissement selon l'algorithme utilisé [1],[6].

B- Filtres à réponses impulsionnelles de durée infinie (RII)

Leurs réalisations sont récursives, car la sortie de ce type de filtres depend non seulement des entrées précédentes et présentes, mais aussi des sorties antérieures. Leurs fonctions de transfert sont de la forme:

$$H(z) = \sum_{k=0}^{+\infty} h(k)\, z^{-k} = \frac{\sum_{m=0}^{M} b_m\, z^{-m}}{\sum_{n=0}^{N} a_n\, z^{-n}} \qquad (2\text{-}33)$$

avec $M \leq N$ (causalité).

Les constantes ai et bi (i=0,1,..) sont appelées coefficients du filtre.

Le schéma fonctionnel pour de tels filtres est le suivant:[2]

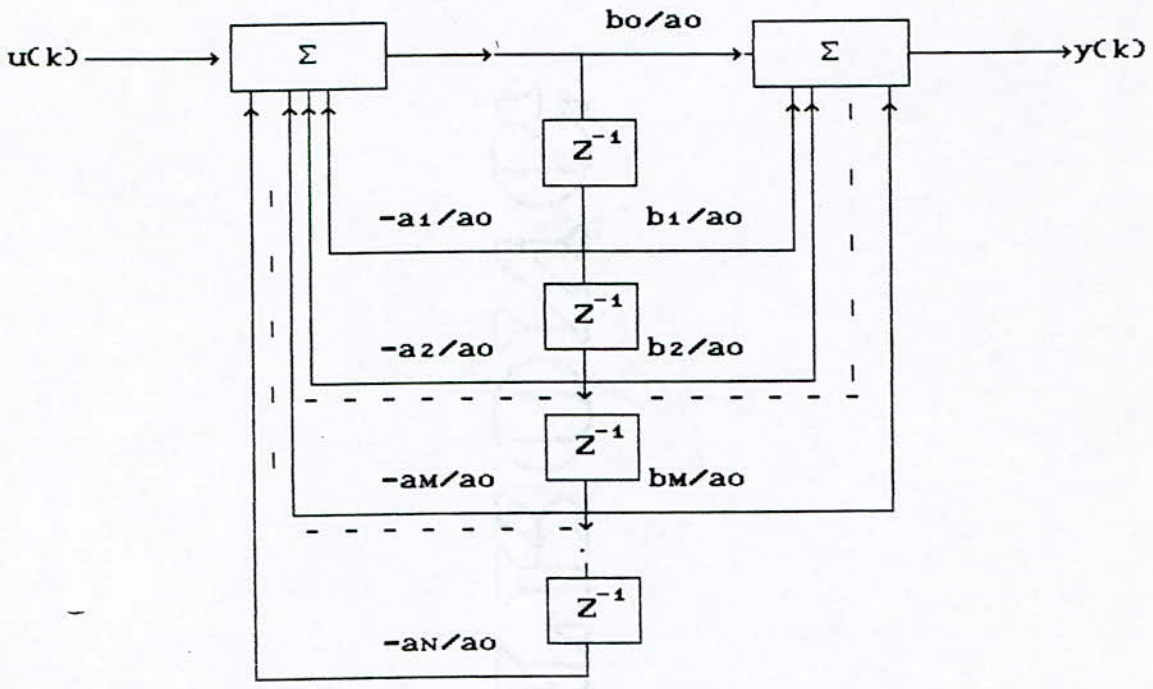


Fig (2-3)- Schéma fonctionnel d'un filtre RII

Les méthodes de synthèse de ce type de filtres se présentent en deux categories:

- Les méthodes classiques qui ètablissent une correspondance entre les domaines analogique et numérique [2].
- Les méthodes algorithmiques d'optimisation assistées par ordinateur [6].

1- Méthodes classiques

a- Méthode de l'invariance de la réponse impulsionnelle

Le principe de cette méthode est de trouver une réponse impulsionnelle h(k) d'un filtre numérique, qui soit la forme échantillonnée de la réponse approximée $\hat{h}(t)$ du filtre analogique desiré [2].

$$h(k) = T_e\, \hat{h}(k\, T_e)$$

où T_e est la période d'échantillonnage,

avec

$$\hat{H}(s) = \mathcal{L}(\hat{h}(t)) = \frac{\alpha_0+\alpha_1 s +.....+\alpha_M s^M}{1 +\beta_1 s +.....+\beta_N s^N} \qquad (2\text{-}35)$$

où $\mathcal{L}$ est l'opérateur de la transformée de LAPLACE.

$$\text{donc} \quad \hat{h}(t) = \begin{cases} c_1 e^{p_1 t} + c_2 e^{p_2 t} + ... + c_N e^{p_N t} & t \geq 0 \\ 0 & t < 0 \end{cases} \qquad (2\text{-}36)$$

(on suppose ici que les pôles p_i de $\hat{H}(s)$ sont simples).

$$\text{En posant } \hat{h}_i(t) = \begin{cases} c_i e^{p_i t} & t \geq 0 \\ 0 & t < 0 \end{cases} \qquad (2\text{-}37)$$

la forme échantillonnée est alors :

$$h_i(k) = \begin{cases} T_e\, c_i\, e^{p_i T_e k} & k \geq 0 \\ 0 & k < 0 \end{cases} \qquad (2\text{-}38)$$

dont la transformée en z est

$$H_i(z) = \frac{T_e\ c_i}{1 - e^{p_i T_e} z^{-1}} \qquad (2\text{-}39)$$

$$\text{donc} \quad H(z) = \sum_{i=1}^{N} H_i(z) = \sum_{i=1}^{N} \frac{T_e\ c_i}{1 - e^{p_i T_e} z^{-1}} \qquad (2\text{-}40)$$

$$\text{et} \quad H(e^{j\omega T_e}) = \sum_{i=-\infty}^{+\infty} \hat{H}\left[j\omega + jn\frac{2\pi}{T_e}\right] \qquad (2\text{-}41)$$

où $H(e^{j\omega T_e})$ est la somme des versions de $\hat{H}(jw)$ répétées à des périodes de $2\pi/T_e$ (effet de l'échantillonnage de $\hat{h}(t)$). Pour éviter le phénomène du recouvrement de ces versions, cette méthodes de synthèse se restreint seulement aux filtres à bandes étroites et à transitions raides[1].

b- Méthode de la transformation bilinéaire

Cette méthode consiste à construire pour chaque filtre analogique $\hat{H}(s)$, un filtre numérique H(z) tel que:

$$H(z = e^{j\omega T_e}) = \hat{H}(s = \Omega(\omega)) \qquad (2\text{-}42)$$

où $\Omega(\omega)$ est appelée la distorsion de fréquence.

Si ωT_e prend ses valeurs dans l'intervalle $[-\pi,\pi]$ (théorème

d'échantillonnage), alors $\Omega(\omega)$ varie de $]-\infty,+\infty[$. Cette transformation qui applique le plan complexe S dans le plan complexe Z, est appelée transformation bilinéaire; elle est définie par :

$$z = \frac{1+s}{1-s} \quad \Leftrightarrow \quad s = \frac{z-1}{z+1} \qquad (2\text{-}43)$$

d'où $$j\Omega(\omega) = \frac{e^{j\omega T_e}-1}{e^{j\omega T_e}+1}$$

c'est à dire $$\omega_a = \Omega(\omega) = tg\ (\omega T_e/2) \qquad (2\text{-}44)$$

où ω_a est la fréquence analogique correspondant à la fréquence numérique ω.

Cette methode évite le recouvrement et conserve la stabilité, cependant elle introduit une distorsion dans l'axe des fréquences qui est généralement remédiable par le procedè dit de préwarping [2].

2- Méthodes d'optimisation

a-Approximation de PADE

Pour synthetiser un filtre numérique dont la réponse fréquentielle est $H(\omega)$ avec $-\frac{\pi}{T_e} < \omega < \frac{\pi}{T_e}$, on l'approxime par un filtre RII causal :

$$\hat{H}(z) = \frac{\sum_{i=0}^{M} b(i)\, z^{-i}}{1+\sum_{i=1}^{N} a(i)\, z^{-i}} = \frac{B_M(z)}{A_N(z)} \qquad N \geq M \qquad (2\text{-}45)$$

avec $a(0) = 1$

ou encore $$\hat{H}(z) = \sum_{k=0}^{+\infty} \hat{h}(k)\, z^{-k}$$

La méthode consiste à égaliser $h(k)$ à $\hat{h}(k)$ pour $T = M + N + 1$ échantillons, c'est à dire pour $k= 0,1,\ldots T$.

Donc avec $\hat{H}(z)\ .A_N(z) = B_M(z)$

$$\text{on a} \quad \sum_{i=0}^{M} a(i)\, h(k-i) = b(k) \qquad \text{pour } k=0,1,\ldots\ldots,M \qquad (2\text{-}46)$$

$$\sum_{i=0}^{N} a(i)\, h(k-i) = 0 \qquad \text{pour } k=M+1,\ldots\ldots,N$$

Ainsi la résolution de ce système linéaire (2-46) permet de trouver les coefficients a(i) et b(i).

Cette méthode est fiable pour les filtres d'ordres élevés, mais en revanche, elle nécéssite une capacité mémoire importante et la stabilité du filtre obtenu n'est pas toujours garantie [7].

b-Méthode des moindres carrés [6]

Cette méthodes consiste à généraliser l'approximation précédente pour le reste des échantillons de la réponse impulsionnelle h (pour k= T+1 à ∞). On cherche les coefficients a(i) et b(i) avec une contrainte sur l'erreur d'approximation:

$$\begin{aligned} \varepsilon(z) &= (H(z) - \hat{H}(z))\ U(z) \\ &= Y(z) - \hat{Y}(z) \end{aligned} \qquad (2\text{-}47)$$

qui doit avoir une énergie minimale en supposant que l'entrée u(k) soit un bruit blanc. Soit à minimiser :

$$W = \frac{1}{2\pi} \int_{-\pi}^{\pi} | \varepsilon(e^{j\theta}) |^2 \, d\theta = E\,[e^2(k)] \qquad (2\text{-}48)$$

avec $E\,[e^2(k)] = \sum_{k=0}^{+\infty} (h(k) - \hat{h}(k))^2$

Comme la minimisation de cette expression conduit à des équations non linéaires, on modifie la formulation de l'erreur de façon à rester dans le cas linéaire; c'est la méthode des moindres carrés modifiès.

Sachant que : $$E(z) = \left[H(z) - \frac{B_M(z)}{A_N(z)}\right] U(z) \qquad (2\text{-}49)$$

on pose $$\begin{aligned} \hat{E}(z) &= E(z)\ A_N(z) \\ &= (H(z)\ A_N(z) - B_M(z))\ U(z) \end{aligned} \qquad (2\text{-}50)$$

donc il suffit de trouver les coefficients a(i) et b(i) qui minimisent l'energie de l'erreur modifiée:

$$\begin{aligned} W &= \frac{1}{2\pi} \int_{-\pi}^{\pi} | H(e^{j\theta})\ A_N(e^{j\theta}) - B_M(e^{j\theta}) |^2 d\theta \\ &= \sum_{k=0}^{+\infty} (h(k) * a(k) - b(k))^2 \end{aligned} \qquad (2\text{-}51)$$

La minimisation de W par rapport à a(k) et b(k) conduit au

système d'equation suivant:

$$\begin{matrix} \leftarrow & N+1 & \rightarrow\leftarrow & M+1 & \rightarrow \end{matrix}$$

$$\begin{bmatrix} R & | & -H_o^T \\ \hline -H_o & | & I \end{bmatrix} \begin{bmatrix} A \\ \hline B \end{bmatrix} = \begin{bmatrix} \alpha_n \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad (2\text{-}52)$$

avec $[R]_{ij} = \sum_l h(l)\, h(l+|i-j|) \quad i,j = 0,\ldots,N$

$[Ho]_{ij} = h(i-j) \qquad i \le j,\ i = 0,\ldots,N$

$[A]_i = a(i-1) \qquad i = 1,\ldots,N+1$

$[B]_i = b(i-1) \qquad i = 1,\ldots,M+1$

et α_n l'énergie minimale de l'erreur.

La résolution de ce système linéaire peut être faite par n'importe quel algorithme de résolution des systèmes linéaires. Mais du point de vue temps de calcul et espace mémoire, l'algorithme de C.T.MULLIS et R.A.BOBERTS se révèle le plus efficace [7].

2-3 Transformation fréquentielle

C'est un outil mathématique trés efficace dans la synthése des différents types de filtres : passe-bas, passe-haut, passe bande, etc..., à partir d'un filtre prototype H(z) généralement passe bas.

Le filtre construit est

$$G(z) = H(F(z)) \qquad (2\text{-}53)$$

où F(z) est appelée transformation fréquentielle et est donnée par

$$F(z) = \pm \prod_{i=1}^{m} \left[\frac{z - \alpha_i^*}{1 - \alpha_i z} \right] \qquad (2\text{-}54)$$

avec $|\alpha_i| < 1 \qquad \forall\, i$

Si H(z) est d'ordre n, G(z) sera d'ordre m n . F(z) se comporte de la même manière que la variable z dans H(z). Par conséquent la stabilitè de H(z) implique celle de G(z).

A partir d'un filtre analogique prototype $\hat{H}(s)$, on peut synthétiser un filtre numérique G(z) avec des caractèristiques bien specifiées, par deux procedès: [1]

1- Soit appliquer une transformation fréquentielle analogique pour trouver $\hat{G}(s)$, ensuite utiliser la transformation bilinéaire pour obtenir G(z).

2- Soit utiliser, d'abord, la transformation bilinéaire pour

trouver le filtre numérique prototype correspondant H(z), ensuite, appliquer une transformation fréquentielle numérique pour obtenir G(z).

Les transformations fréquentielles peuvent aussi être utiliser pour linéariser la réponse de phase d'un filtre numérique [2];[7]. Notons aussi que le spectre d'amplitude de telles transformations est constant sur toute la gamme fréquentielle, la raison pour laquelle F(z) est appelée filtre passe tout.

2.4 Tableau comparatif entre les filtres RIF et RII

Filtres R I F	Filtres R I I
* Ces filtres sont toujours stables, souvent utilsés dans les systémes adaptatifs et multicadences [8].	* La stabilité depend de la position des poles par rapport au cercle unité[8].
* leur formulation est simple et leur structure est non recursive.	* Plus dèlicats au stade de leur concrétisation, sauf pour les filtres élliptiques[8].
* Phase linéaire facilement obtenue.	* La linéarité de la phase peut etre seulement approchée à l'aide de systémes d'égalisation du temps de propagation de groupe[8].
* Les erreurs de calculs dûes aux effets des registres finis ne sont pas cumulatives et sont simples à analyser.	* Les conséquences de la limitation de la longueur des registres sont graves car les erreurs de calcul sont cumulatives.
* Temps de calcul long surtout si l'ordre est grand (bande de transition étroite).	* Calculs simples et moins longs même pour des transitions trés raides.
	* Possibilité d'avoir des réponses fréquentielles très séle ctives en jouant sur les pôles.
* Nécéssitent un espace mémoire important si l'ordre est grand.	* Nécéssitent un faible espace mémoire.

CHAPITRE III

EFFETS DE LA LONGUEUR LIMITEE DES MOTS SUR LE FILTRAGE NUMERIQUE

CHAPITRE III

EFFETS DE LA LONGUEUR LIMITEE DES MOTS SUR LE FILTRAGE NUMERIQUE

3-1-Introduction

Dans les realisations aussi bien materielles que logicielles (simulation) des filtres numériques, on est amené à stocker les nombres dans des registres de longueurs finies. Par conséquent, les valeurs du signal et des paramétres caractérisant le filtre doivent être quantifiées avant d'être stockées. L'utilisation de registres limités oblige à accorder une grande attention aux effets qu'elle occasionne au filtrage; on peut en citer:

1-Le bruit de la conversion anlogique-numerique.

2-Le bruit de calcul incorrélé.

3-L'inéxactitude de la réponse du filtre dûe à la quantification des coefficients.

4-Le bruit de calcul corrélé ou les cycles limites.

Ces effets qui affectent la performance du filtre dépendent essentiellement:

-du type d'arithmétique utilisée dans l'algorithme du filtre

-du type de la quantification utilisée pour réduire les mots à la longueur desirée (par l'arrondi ou la troncature).

-de la structure choisie du filtre utilisé.

Indépendamment de l'encodage des signaux (c'est à dire, de l'arithmétique), les multiplications, et dans certains cas, les additions requièrent généralement une augmentation de la longueur du mot pour contenir le résultat de l'opération.

Comme le nombre d'opérations effectuées sur le signal reste fini, on peut accommoder l'accroissement de la longueur des mots par l'utilisation de registres plus longs que ceux du signal original afin de pouvoir stocker les résultats des opérations arithmétiques. Cependant, dans ces cas,de très longs registres peuvent être necessaires et pour cette raison il est d'ordinaire en pratique, de réduire la taille des mots.

Pour les filtres non récursifs, une telle réduction de la taille des mots cause un signal d'erreur additif à la sortie similaire à celui dû à la quantification dans le convertisseur analogique-numérique.

Par contre, pour les filtres récursifs, une réduction de la longueur des mots est nécessaire à chaque boucle de retour , pour éviter une éventuelle augmentation de la longueur des mots. Cette opération non linéaire introduit des erreurs causant des effets graves [9].

D'autre part, les filtres utilisés le plus souvent en pratique sont ceux ayant de hauts facteurs de qualité donc des gains importants;en conséquence les signaux peuvent devenir plus grands que la valeur maximum représentable par le nombre de digits disponibles, en considérant la gamme dynamique desirée. Si un dépassement a lieu dans un filtre, le digit (généralement, binaire) le plus significatif doit être, à son tour, stocké dans un registre.

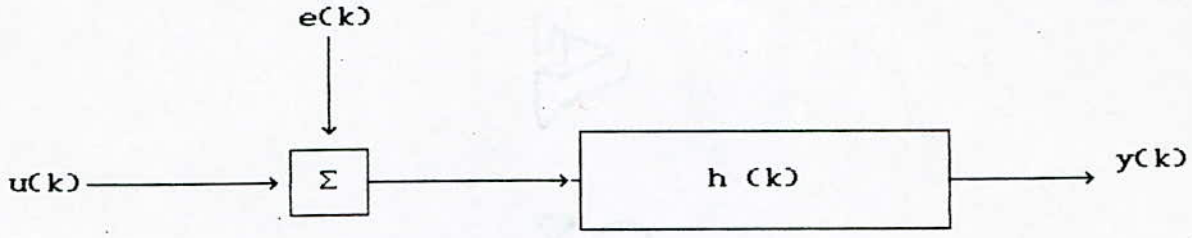
3-2 Conversion analogique numérique

Dans la plupart des applications de filtrage numérique, le signal original à filtrer est de nature analogique; c'est pourquoi un convertisseur analogique numérique (CAN) est toujours une partie intégrante du filtre numérique. Le CAN assure l'échantillonnage du signal continu avec une cadence au moins égale ou supérieure au double de la fréquence limite du signal et le codage des échantillons en une suite de nombres binaires (en général) de longueurs finies.

L'erreur entre un nombre réel u(k) et sa représentation binaire finie; $[u(k)]_Q$, est appelée bruit de quantification. Elle est supposée aléatoire et est définie par:

$$e(k) = u(k) - [u(k)]_Q \qquad (3\text{-}1)$$

On schématise l'effet de la quantification à l'entrée d'un filtre numérique de réponse impulsionnelle h(k) par (Fig 3-1)



Fig(3-1)-Introduction de l'erreur de quantification

La sortie du filtre s'ecrit:

$$y(k) = \underbrace{u(k) * h(k)}_{\text{signal de sortie utile}} + \underbrace{e(k) * h(k)}_{\text{effet du bruit}} \qquad (3\text{-}2)$$

3-3 La quantification

La quantification est l'approximation de chaque valeur du signal par un multiple entier d'une quantité elementaire q appelée pas de quantification. Cette opération consiste à faire passer le signal dans un organe ayant une caractéristique en marche d'escalier Fig(3-2).

Le centrage de cette caractéristique dépend de la maniére avec laquelle l'approximation est faite:

-l'approximation par l'arrondi consiste à arrondir toute valeur comprise entre $(n-\frac{1}{2})q$ et $(n+\frac{1}{2})q$ à nq. Cette quantification rend la moyenne du signal d'erreur nulle (Fig(3-2-(a)).

-l'approximation par la troncature consiste à approcher par nq toute valeur comprise entre nq et (n+1)q (Fig 3-2-(b)).

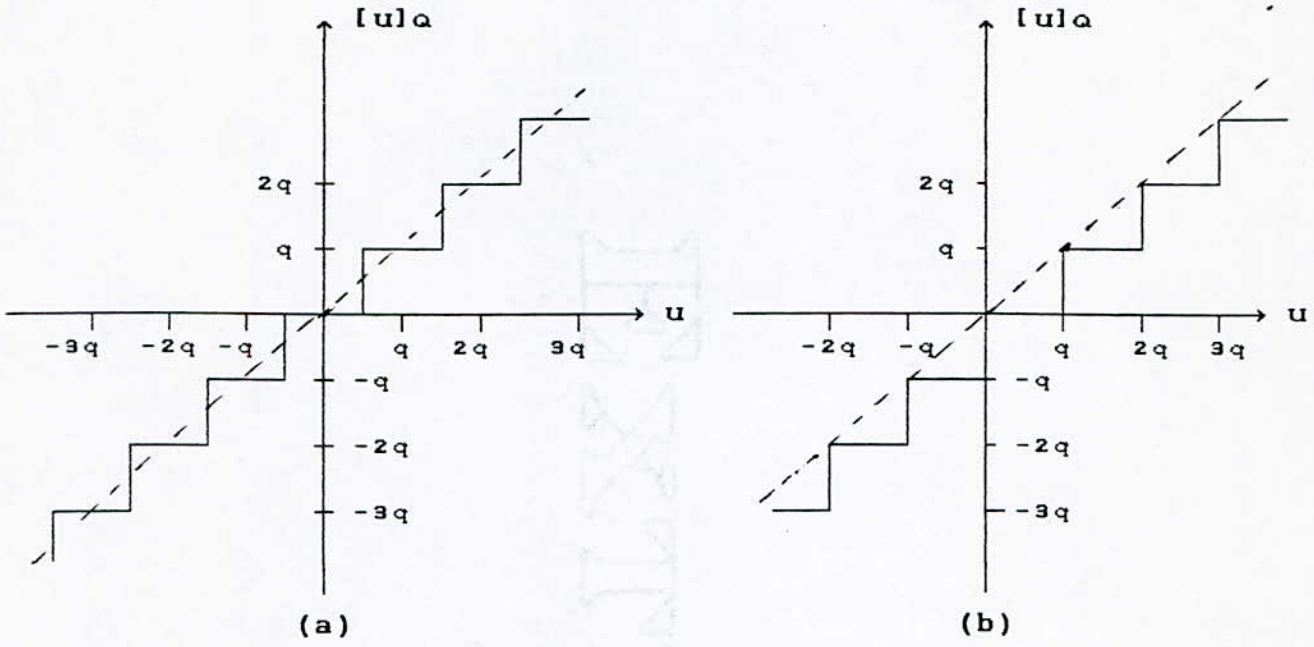


Fig (3-2) - Caractèristique du quantificateur
(a) par l'arrondi, (b) par la troncature

Si la gamme des valeurs du signal couvre le domaine [-Δ ,Δ] et si l'on code (en binaire) chaque valeur par des mots de B bits,

$$q = \Delta\, 2^{-B+1} \qquad (3\text{-}3)$$

dans ce cas la quantification est dite uniforme, (q constant)[8].

La distribution du bruit est uniforme, et sa puissance (variance) est estimée à:

$$P = \frac{q^2}{12} \qquad (3\text{-}4)$$

3-4-Types de représentation arithmétique binaire

Il existe diverses façons d'établir la correspondance entre l'ensemble des amplitudes quantifiées et l'ensemble des nombres binaires qui doivent les représenter. Les signaux à coder ayant des valeurs, en géneral, positives et négatives, les représentations préferées sont celles qui conservent l'information du signe. Les plus courantes sont:

3-4-1 Représentation en virgule fixe

Généralement un nombre V s'exprime dans la base r par :

$$V = \sum_{i=-m}^{n} b_i \, r^i \qquad \text{avec} \quad 0 \leq b_i \leq r-1, \; b_i \text{ entier} \qquad (3\text{-}5)$$

on le note dans cette base:

$$V = (b_n \, b_{n-1} \ldots b_0 . b_{-1} \ldots b_{-m}) \qquad (3\text{-}6)$$

La représentation binaire correspond à r = 2 avec b_i= 0 ou 1. Le point séparant V en deux parties s'appelle virgule binaire.

Dans la représentation, en virgule fixe, la virgule binaire se situe entre le premier et le second bits du registre les plus significatifs, où la premiére position représente le bit de signe.

Dans cette représentation un nombre V fractionnaire et signé peut s'écrire sous trois formes :[8],[10],[11]

1-Signe et valeur absolue:

$$V_{sva} = \begin{cases} 0.b_{-1}b_{-2}\ldots b_{-m} & \text{si } V \geq 0 \\ 1.b_{-1}b_{-2}\ldots b_{-m} & \text{si } V < 0 \end{cases} \qquad (3\text{-}7)$$

2-Complément à 1 :

$$V_{c1} = \begin{cases} V & \text{si } V \geq 0 \\ 2 - 2^{-B} - |V| & \text{si } V < 0 \end{cases} \qquad (3\text{-}8)$$

où B est la longueur du mot (nombre de bits)

3-Complément à 2 :

$$V_{c2} = \begin{cases} V & \text{si } V \geq 0 \\ 2 - |V| & \text{si } V < 0 \end{cases} \qquad (3\text{-}9)$$

Le choix entre ces trois représentations arithmétiques est généralement dicté par les considérations des moyens matériels disponibles et de la programmation.

L'opération d'addition (ou soustraction) est directe dans le cas du complément à 2. L'arithmétique du signe et valeur absolue convient mieux à la multiplication car elle se fait simplement par la multiplication bit par bit des valeurs absolues et par l'ajustement du bit de signe du résultat.

Dans la plupart des réalisations de filtres numériques en virgule fixe, la position de la virgule binaire est supposée être à la droite du premier bit le plus significatif.

Ainsi la gamme des nombres représentables va de -1.0 à $1.0 - 2^{-(B-1)}$ où B est le nombre de bits du mot. C'est pourquoi souvent, les signaux sont normalisés afin de respecter la gamme choisie [10].

3-4-2 Représentation en virgule flottante

La représentation en virgule flottante d'un nombre consiste en deux parties; la mantisse et l'exposant, telle que un nombre V, dans cette représentation est noté :

$$V = m \times 2^{e} \qquad (3\text{-}10)$$

où e est l'exposant (signé),

m est la mantisse signée et normalisée avec $\frac{1}{2} \leq m < 1$

Généralement le nombre de bits assigné à la mantisse est égal au trois-quart du nombre de bits total du mot [8].

Ce type de représentation permet une extension de la dynamique du fait de l'effet multiplicatif introduit par l'exposant, cependant il entraine une complication des opérations arithmétiques et des circuits [11].

C'est pour cette raison qu'on se limite, dans la suite de ce travail, à l'étude du cas de la représentation en virgule fixe uniquement.

3-5 Erreurs de calcul et de dépassement

En plus de l'erreur causée par la conversion analogique -numérique à l'entrée du filtre qui est indépendante de la réalisation de celui-ci, des erreurs intrinséques au filtre dûes à la limitation de la longueur des mots (représentés en virgule fixe) introduisent des effets indésirables dans le filtrage . Ces effets causent trois types d'erreurs :

1-Erreurs de calcul dûes à la quantification des produits

A chaque instant, dans un filtre numérique,un signal representé par b_1 bits est multiplié par un coefficient representé par b_2 bits en donnant, à la sortie du multiplieur, un résultat sur b_1+b_2

bits. Comme la longueur des registres est fixe pour tout le système, chaque sortie des multiplieurs doit être quantifiée avant la prochaine opération [10].

2-Erreurs de quantification des coéfficients

Durant l'étape de synthése, les coefficients de la fonction de transfert du filtre sont normalement évalués avec une grande précision. S'ils sont quantifiés, la réponse fréquentielle va diffèrer considérablement de la réponse désirée et si, en plus,le pas de quantification est mal choisi, le filtre peut ne pas satisfaire les spécifications désirées.

Cet effet est appelé "sensibilité" des coefficients à la quantification c'est un effet déterministe qui peut être dominé par le bruit de calcul [1].

En effet la limitation du nombre de bits des coefficients se traduit par le fait qu'ils ne peuvent prendre qu'un nombre limité de valeurs, il s'en suit que les pôles ont un nombre limité de positions possibles à l'interieur du cercle unité, il en est de même pour les zéros.

Ainsi la quantification à B bits de la valeur absolue des coefficients limité à 2^{2B}, le nombre de positions que peuvent prendre les pôles dans un quart du cercle unité et à 2^{B} le nombre de fréquences de coupure [8],[10].

3- Cycles limites dûs aux effets non linéaires de la quantification :

Dans l'étude du bruit de calcul, on suppose toujours que les valeurs des échantillons du signal d'entrée du filtre numérique sont de même ordre de grandeurs que les différents multiples du pas de quantification q. Ce qui nous permet d'admettre que les échantillons du bruit de calcul sont statistiquement incorrélès aussi bien entre eux qu'avec la suite du signal d'entréé.

Cependant, le signal à l'entrée peut atteindre des valeurs faibles durant certains temps, par conséquent, les erreurs de quantification tendent à devenir fortement corrélés et peuvent causer l'instabilité du filtre par l'apparition d'une auto-oscillation appelée cycles limites [8],[10].

En effet, bien que les conditions de stabilité soient remplies et que le signal à l'entrée soit absent,un signal périodique peut

apparaitre, généralement avec de faibles amplitudes. Ces oscillations tiennent au fait qu'en réalité le signal d'entrée n'est jamais nul en l'absence de données à l'entrée; le signal d'erreur dû à la quantification des nombres avant la mise en mémoire est appliqué au filtre.

Une borne peut être obtenue pour ces oscillations sachant que le signal d'erreur e(k) posséde en lui même, dans le cas d'arrondi (avec pas q), une borne donnée par :

$$|e(k)| \leq \frac{q}{2} \qquad (3\text{-}11)$$

Si la réponse impulsionnelle du filtre est h, les auto-oscillations sont limitées par:

$$|y(k)| \leq \frac{q}{2} \sum_{i} |h(i)| \qquad (3\text{-}12)$$

Cette borne est en fait trés large; une estimation plus réaliste de l'amplitude des auto-oscillations est donnée par :

$$Am = \frac{q}{2} \text{ Max } |H(\omega)| \qquad (3\text{-}12)$$

où $H(\omega)$ est la réponse impulsionnelle du filtre.

Pour que les cycles limites ne soient pas gênants, le nombre de bits des mémoires doit être suffisamment grand et le pas q suffisamment petit [1]. Il faut noter que ces cycles limites peuvent être éliminés par l'utilisation de la troncature au lieu de l'arrondi [10].

3-5-1 Oscillations de dépassement

Si l'amplitude d'un signal interne d'une réalisation en virgule fixe excède la gamme dynamique, un dépassement se produit et le signal de sortie va être distordu.

L'erreur causée par les dépassements est déterminée par la méthode utilisée pour représenter un nombre dépassant la gamme. Ce dernier peut être transformé par plusieurs façons, selon la caractéristique de dépassement choisie. Parmi lesquelles, on en cite:

a-La caractéristique de complément à 2 :utilisée souvent dans la représentation en complément à deux. Elle consiste à attribuer à chaque nombre dépassant la gamme, le complement à 2 du plus grand

(ou plus petit) nombre représentable.

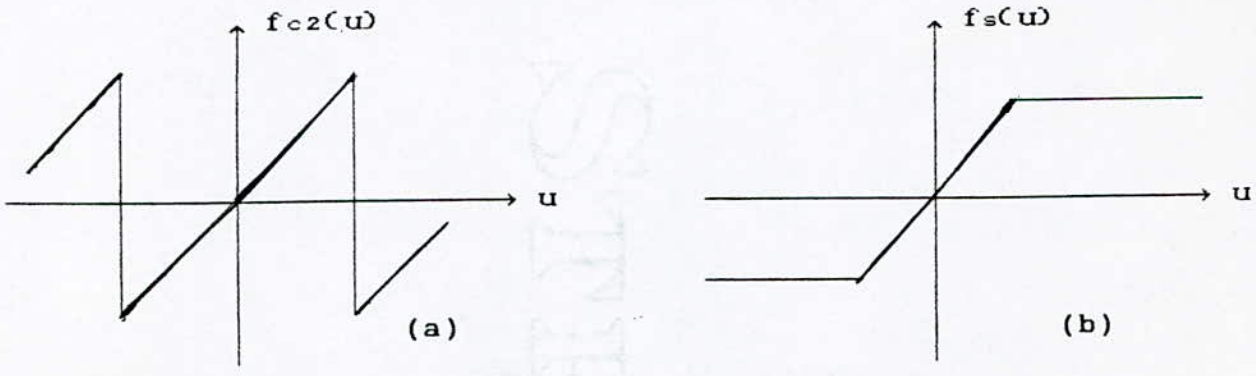
Cette caractéristique est périodique (voir Fig(3-3-(a)) et a pour propriété d'annuler les dépassements intermédiaires dans un accumulateur et d'obtenir des résultats de somme qui soient dans la gamme [1],[9].

b-La caractéristique de saturation qui consiste à remplacer le nombre dépassant la gamme par le plus grand (ou le plus petit) nombre représentable (Fig(3-3-(b)). L'erreur de dépassement de cette caractéristique est inférieure à celle du complément à 2, mais elle est difficile à réaliser matériellement [1],[9].

Lorsqu'un dépassement se produit à l'entrée, la propagation de l'erreur peut causer d'autres dépassements. Aussi, aprés un dépassement interne la sortie du filtre peut, en dependant des pôles de celui-ci,devenir indépendante du signal d'entrée; c'est l'auto-oscillation de depassement. Les oscillations de dépassement sont causées par la non linéarité de la caractéristique de dépassement utilisée [1],[9],[10].

Les dépassements peuvent être minimisés en élargissant la gamme des représentations possibles; et ceci en augmentant la valeur de Δ de (3-3).

Cependant, si le nombre de niveaux de quantification reste le même la valeur du pas q doit être augmentée aussi; en conséquence, une diminution de la possibilté de dépassement n'est obtenue qu'au prix d'une augmentation de l'erreur de quantification.



Fig(3-3) Caractéristiques de dépassement :
(a) complement à 2, (b) saturation

CHAPITRE IV

REPRESENTATION DES FILTRES NUMERIQUES PAR LES VARIABLES D'ETAT

CHAPITRE IV

REPRESENTATION DES FILTRES NUMERIQUES PAR LES VARIABLES D'ETAT

4-1- Introduction

Jusqu'à présent, le filtre numérique a étè representé par sa caracteristique d'entrée-sortie, c'est à dire sa fonction de transfert ou sa reponse impulsionnelle qui caracterisent complétement les propriétés et les specifications du filtre à realiser. Mais étant donné que tout système numérique n'effectue que trois types d'operatins: les multiplications, les additions et les decalages; la realisation du filtre numerique exige en plus de la determination de la fonction de transfert, la connaissance de l'ordre des operations elementaires à exécuter. Donc il devient imperatif de concevoir une representation plus detaillée du filtre afin de décrire exactement l'opération du filtrage.

La connaissance de la manière dont s'effectuent les opérations à l'interieur du filtre permet de mieux analyser les erreurs dûes à l'utilisation de registres de longueurs finies et par conséquent de trouver répondant aux specifications du filtre et présentant un meilleur rapport signal/bruit.

Pour ce fait, on utilise la représentation par les variables d'etat pour décrire la realisation du filtre numerique, qui constitue en plus d'un outil mathématique puissant, un moyen d'analyse souple et efficace que nous allons découvrir par la suite.

4-2 Representation d'etat des filtres numeriques

Tout système linéaire d'ordre N, à l'instant k, auquel est appliquée une suite u(k) à l'entrée et qui fournit une suite y(k) en sortie, est defini par le couple d'equation dites d'etat suivant:

$$\begin{aligned} x(k+1) &= A\ x(k) + B\ u(k) \\ y(k) &= C\ x(k) + D\ u(k) \end{aligned} \qquad (4-1)$$

où x(k) est un vecteur Nx1 dont chaque élément est une variable d'état du système (ou filtre).

A est une matrice N x N appelée "matrice du système" (filtre).

B est un vecteur N x 1, c'est le vecteur de "commande".

C est un vecteur 1 x N appelée vecteur "d'observation".

D est le coéfficient de la transition directe de l'entrée à la sortie.

L'état du filtre, à l'instant k, en fonction de l'état initial x(0) est

$$x(k) = A^k \; x(0) + \sum_{i=1}^{k} A^{i-1} \; B \; u(k-i) \qquad (4-2)$$

La fonction de transfert du système est obtenue en prenant la transformée en Z des equations (4-1):

$$(z\,I - A)^{-1} \; X(z) = B \; U(z) \qquad (4-3)$$

$$Y(z) = C \; X(z) + D \; U(z)$$

En éliminant X(z), on obtient:

$$H(z) = C \; (z\,I - A)^{-1} \; B + D \qquad (4-4)$$

Les pôles de la fonction de transfert ainsi obtenue, ne sont en fait que les racines du polynômes caracteristique de la matrice A tel que:

$$a(z) = \det(\, z\,I - A\,) \qquad (4-5)$$

Ceux sont donc les valeurs propres de la matrice A.

Pour assurer la stabilité du système, il suffit que les modules des valeurs propres de A soient inférieurs à l'unité. Ainsi, en régime libre (entrée nulle):

$$x(k) = A^k \; x(0) \qquad (4-6)$$

tendra vers zéro quand k tend vers l'infini. En régime établi, l'équation (4-2) devient pour les systèmes stables:

$$x(k) = \sum_{i=1}^{+\infty} A^{i-1} \; B \; u(k - i) \qquad (4-7)$$

(quand k tend vers l'infini)

La réponse impulsionnelle h(k) du filtre peut être exprimée

en fonction des paramétres d'etats; sachant que la sortie du filtre s'ecrit:

$$y(k) = \sum_{l=0}^{+\infty} h(l)\ u(k-l) \qquad (4\text{-}8)$$

et par ailleurs ,

$$y(k) = C\ x(k) + D\ u(k)$$

$$= C \sum_{i=1}^{+\infty} (A^{i-1} B\ u(k-i)) + D\ u(k)$$

$$= \sum_{i=1}^{+\infty} C\ A^{i-1} B\ u(k-i) + D\ u(k) \qquad (4\text{-}9)$$

Enfin, par identification des equations (4-8) et (4-9), on obtient

$$h(k) = \begin{cases} 0 & ,k<0 \\ D & ,k=0 \\ C\ A^{k-1}\ B & ,k>0 \end{cases} \qquad (4\text{-}10)$$

* Changement de coordonnées

La fonction de transfert (ou la réponse impulsionnelle) specifie la relation entrée-sortie du filtre sans pour autant apporter une information sur sa structure interne.

Comme le vecteur d'etat x(k) represénte N variables d'etat internes, on peut changer le système de coordonées de ce vecteur dans l'espace d'etat sans changer la relation d'entrée-sortie (ou la reponse impulsionnelle du filtre). Dans ce sens, il existe une certaine latitude dans le choix des paramétres d'etat pour lesquels les valeurs propres de la matrice A imposées par le filtre restent inchangées.

En appliquant une transformation T, une matrice non singuliére N x N, le nouveau vecteur d'etat est donnée par:

$$x'(k) = T^{-1}\ x(k) \qquad (4\text{-}11)$$

Les paramétres d'etat se transforment alors comme suit

$$(A,\ B\ ,\ C,\ D) \Longrightarrow (A' = T^{-1} A\ T, B' = T^{-1} B, C' = CT,\ D) \qquad (4\text{-}12)$$

On peut verifier que ce type de transformation conserve la fonction de transfert et donc la reponse impulsionnelle [1].

Une structure canonique ou directe du filtre est definie par des paramétres de la forme:

$$A = \begin{bmatrix} 0 & & \\ \vdots & I & \\ \cdot & & \\ -\frac{a_N}{a_o} & -\frac{a_{N-1}}{a_o} \cdots & -\frac{a_1}{a_o} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}, \quad C^T = \begin{bmatrix} b_N - \frac{bo}{ao} a_N \\ \vdots \\ b_1 - \frac{bo}{ao} a_1 \end{bmatrix}, \quad D = \frac{b_o}{a_o} \qquad (4\text{-}13)$$

où a_i et b_i $(i=0,\ldots N)$ sont les coefficients du dénominateur de la fonction de transfert du filtre.

4-3 Matrice K et la stabilité de LYAPUNOV

La représentation des filtres numeriques linéarires par les variables d'etat est plus efficace quand il s'agit de calculer les quantités qui dependent de la structure interne du filtre et de connaitre la maniére dont ces grandeurs changent quand la structure est modifiée.

Ces quantités à calculer sont généralement statistiques et interviennent quand les entrées appliquées aux filtres sont stationnaire au sens large [1].

Parmi ces quantités , on s'intéresse surtout à la matrice de covariance de l'etat du filtre décrit par les paramétres (A,B,C,D) et qui est donnée par :

$$K = E\,[x(k)\; x^T(k)] \qquad (4\text{-}14)$$

En supposant que la stabilité du systéme fait que

$$x(k+1) = x(k) \quad \text{pour} \quad k \longrightarrow +\infty \qquad (4\text{-}15)$$

donc les elements de K sont :

$$K_{ij} = E\,[\,x_i(k)\; x_j(k)\,] \qquad (4\text{-}16)$$

En remplaçant l'equation (4-7) dans (4-14), on a:

$$K = \sum_{l=0}^{\infty} \sum_{m=0}^{\infty} (A^l B)\, r_{uu}(l-m)\, (A^m B)^T \qquad (4\text{-}17)$$

avec $r_{uu}(\tau) = E\,[\,u(k)\; u(k+\tau)\,]$

C'est la fonction d'autocorrélation de l'entrée.

Pour une entrée bruit blanc centré et normalisé, on a

$$r_{uu}(\tau) = \delta(\tau) \qquad (4\text{-}18)$$

La double somme de l'équation (4-17) devient alors:

$$K = \sum_{i=0}^{+\infty} (A^i B)(A^i B)^T \qquad (4\text{-}19)$$

On peut eviter les sommes infinies dans le calcul de la matrice K en utilisant les équations d'état:

$$K = E[x(k+1)\, x^T(k+1)]$$

$$= E[(A\,x(k) + B\,u(k))(A\,x(k) + B\,u(k))^T] \qquad (4\text{-}20)$$

Et sachant que x(k) et u(k) sont incorrélés pour des entrées bruit blanc, alors (4-20) devient:

$$K = E[A\,x(k)\,x^T(k)\,A^T + B\,u(k)\,u^T(k)\,B^T]$$

d'où

$$K = A\,K\,A^T + B\,B^T \qquad (4\text{-}21)$$

Cette derniére expression est appelée equation de LYAPUNOV.

Un cas particulier de la théorie de la stabilité de LYAPUNOV est:

Si le couple (A,B) est contrôlable, c'est à dire que matrice de contrôlabilité

$$M = \left[B,\ AB,\ A^2 B,\ \ldots,\ A^{N-1}B \right] \qquad (4\text{-}22)$$

est de rang N, et si la matrice K est definie positive alors la matrice A doit être stable [1].

4-4- Structures éliminant les oscillations de dépassement

Les oscillations de dépassement se produisent si des vecteurs d'état augmentent en amplitude aprés une multiplication par la matrice A du filtre. Comme la caractéristique de dépassement diminue l'amplitude du vecteur d'état si celle-ci depasse la gamme, des oscillations répetées ne peuvent se produire à moins que le processus de la multiplication par A augmente la grandeur du vecteur d'état.

Une condition nécéssaire pour la production des oscillations de

dépassement est que la grandeur de A.x soit superieure à celle de x, pour un x donné de l'espace d'etat.
on peut trouver une autre structure d'etat caracterisée par (A,B,C,D) pour laquelle la grandeur de x n'augmente pas du fait de la multiplication par A. On definit la norme d'un vecteur comme étant

$$\|x\| = (x^T d \; x)^{1/2} \qquad (4\text{-}23)$$

où d est une matrice diagonale positive.

La norme de la matrice A est definie par:

$$\| A \| = \max_{x \neq 0} \frac{\|A\ x\|}{\|x\|}$$

$$= \max_{x \neq 0} \left(\frac{x^T A^T d\ A\ x}{x^T\ x} \right)^{1/2} \qquad (4\text{-}24)$$

Pour éviter les oscillications de dépassement pour une entrée nulle (ou constante), il suffit que:

$$\| A\ x \| \leq r \| x \| \quad \text{avec} \quad 0 < r \leq 1 \qquad (4\text{-}25)$$

Autrement dit, il suffit de trouver la matrice d telle que [1]:

$$Q = r^2 d - A^T d\ A \qquad (4\text{-}26)$$

soit une matrice definie positive.

Parmi les structures qui satisfont la condition : Q définie positive; les structures normales ($A\ A^T = A^T A$), de gain de bruit de calcul minimal, et en treillis [13]. Les formes directes (canoniques) sont susceptibles de produire des oscillations de dépassement qui dependent de la position des pôles.

Dans le cas des filtres du second ordre, si la matrice A (2 x 2) dont les valeurs propres λ verifient: $|\lambda| < 1$, alors il existe une matrice diagonale definie positive d pour laquelle Q de (4-26) est definie positive, si et seulement si, les conditions suivantes sont satisfaites:

$$a_{12} a_{21} \geq 0 \text{ , ou}$$
$$\text{si } a_{12} a_{21} < 0 \quad \text{alors} \quad |a_{11} - a_{22}| + \det(A) < 1 \qquad (4\text{-}27)$$

tion de dépassement [14], [15].

4-5- Normalisation des filtres numériques en virgule fixe

Pour éviter les dépassements dans les registres internes qui causent des erreurs importantes, on normalise convenablement la réalisation du filtre.

Normaliser revient à mettre toutes les valeurs numériques des variables internes dans une gamme appropriée à la réalisation matérielle. La gamme d'une variable interne est necéssairement limitée par le fait de l'utilisation des registres de longueur finie.

4-5-1- Régles de normalisation

Dans la représentation en virgule fixe, une variable interne v(k) telle que

$$v(k) = (f * u)(k) \qquad (4\text{-}28)$$

avec f(k) la réponse impulsionnelle entre l'entrée u(k) et la la variable d'état v(k);
est bornée par le fait que l'entrée u(k) est limitée par:

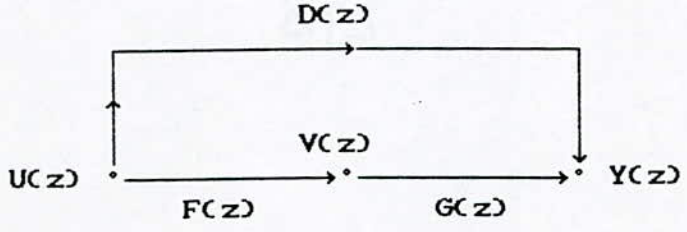
$$|u(k)| \leq \Delta \qquad (4\text{-}29)$$

où Δ dépend du pas de quantification (d'aprés l'equation (3-3)) et qui est généralement normalisé à 1 (dans la représentation en en virgule fixe).

La gamme des valeurs de v dépend ainsi de la nature de l'entrée u(k) et de la suite f(k). En effet, si un filtre caracterisé par sa fonction de transfert

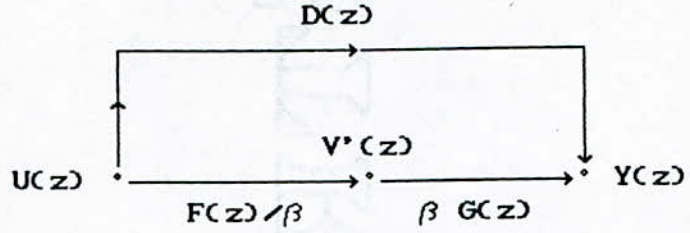
$$H(z) = D(z) + F(z)\ G(z) \qquad (4\text{-}30)$$

et son graphe de fluence (Fig(4-1)):



Fig(4-1) Filtre non normalisé

Il est alors normalisé (c'est à dire que v'(k) devient dans la gamme), de la façon suivante (Fig(4-2)



Fig(4-2) Filtre normalisé

De cette manière, H(z) reste invariant et

$$|v'(k)| \leq \frac{1}{\beta} \sum_{m} |f(m)|\,|u(k-m)| \qquad (4\text{-}31)$$

et comme $|u(k)| \leq 1$ alors $|v'(k)| \leq \frac{1}{\beta} \sum_{m} |f(m)|$ (4-32)

Donc β est choisi de façon à vérifier la relation (4-32) on peut choisir la norme 1 telle que:

$$\beta = \|f\|_1 = \sum_{l=0}^{\infty} |f(l)| \qquad (4\text{-}33)$$

Mais étant donnée que cette norme constitue une borne largement conservatrice de la gamme et que la norme 2:

$$\|f\|_2 = \left[\sum_{l=0}^{\infty} f^2(l)\right]^{1/2} \leq \|f\|_1 \qquad (4\text{-}34)$$

On peut donc choisir

$$\beta = \delta \|f\|_2 \qquad (4\text{-}35)$$

Avec δ un paramètre choisi subjectivement afin d'obtenir une bonne représentation des valeurs de la variable v'(k) dans la gamme, d'éviter ainsi les oscillations dûes aux amplitudes faibles (ou cycles limites), et de réduire les erreurs d'arrondi.

En conclusion, un filtre est normalisé s'il vérifie la contrainte de normalisation définie par une des régles de normalisation dont on cite:[1]

1- Normalisation par la norme 1:

$$\|f\|_1 = 1 \qquad (4\text{-}36)$$

2- Normalisation par la norme 2:

$$\delta \|f\|_2 = 1 \qquad (4\text{-}37)$$

Dans notre cas on opte pour la norme 2 qui permet à la fois de

conserver la gamme et de réduire la probabilité des cycles limites et les erreurs d'arrondi par le choix d'une valeur adèquate de δ.

D'aprés les résultats de simulation, $\delta = 4$ constitue un compromis optimal entre les erreurs d'arrondi et les erreurs de dépassement.

Mais généralement on prend $\delta = 1$.

4-5-2 Normalisation des paramétres d'état

La normalisation d'un filtre par la norme 2, consiste à normaliser le vecteur d'etat qui s'exprime par:

$$x(k)= \sum_{l=0}^{\infty} A^{l} B\, u(k-1-l) \qquad (4\text{-}38)$$

La réponse impulsionnelle de l'entrée à x(k) est alors:

$$f(k) = \begin{cases} 0 & ,k\leq 0 \\ A^{k-1} B & ,k>0 \end{cases} \qquad (4\text{-}39)$$

et donc $F(z)= (zI - A)^{-1}B$

f(k) est un vecteur N x 1 à l'aide duquel on peut exprimer la matrice K de (4-20) pour une entrée bruit blanc normalisé comme suit:

$$K = \sum_{k=0}^{\infty} f(k)\, f^{T}(k) \qquad (4\text{-}40)$$

avec $$K_{ij} = (f_i , f_j) = \sum_{k=0}^{\infty} f_i(k)\, f_j(k) \qquad (4\text{-}41)$$

donc $$K_{ii} = \sum_{k=0}^{\infty} f_i^2(k) = \|f\|_2^2 \qquad (4\text{-}42)$$

Donc pour normaliser ce filtre, il faut lui appliquer une transformation T diagonale pour laquelle la contrainte de normalisation (norme 2) soit :

$$\delta \sqrt{K'_{ii}} = 1 \qquad (4\text{-}43)$$

dans la nouvelle représentation d'etat.

Donc il suffit de prendre :

$$T_{ii} = \delta \sqrt{K_{ii}} \qquad (4\text{-}44)$$

car d'aprés les équations (4-12) et (4-20), on a :

$$K \Longrightarrow K' = T^{-1} K\, T^{-T} \qquad (4\text{-}45)$$

et donc $$K'_{ij} = \frac{K_{ij}}{T_{ii}\, T_{jj}}$$

d'où $$K'_{ii} = 1/\delta^2 \qquad (4\text{-}46)$$

La transformation de normalisation T doit être appliquée aussi aux paramétres d'etats (A,B,C,D) suivant la relation (4-12).

En resumé, pour normaliser un filtre numérique:

i)- on resout l'équation de LYAPUNOV (4-20) pour trouver K.

ii)- on utilise la relation (4-44) pour construire T.

iii)- on applique T aux paramétres (A,B,C,D).

On remarque que la normalisation change la représentation d'etats du filtre, donc sa structure interne et par conséquent le bruit de calcul.

4-6 Analyse du bruit de calcul dans le filtre numerique RII

Le bruit de calcul est l'erreur dûe, essentiellement, à la quantification des résultats des multiplications au niveau des accumulateurs durant l'opération de filtrage.

On assimile le quantificateur (dans le cas d'arrondi) à une source de bruit blanc centré de variance $q^2/12$, (où q est le pas de quantification), et de distribution uniforme sur [-q/2, q/2]. Ce modèle de bruit remplit deux conditions:

a) Les sources des différents accumulateurs sont incorrelées.

b) Chaque source est incorrelée avec l'entrée.

On suppose dans cette analyse que la probabilité des dépassements est trés faible aprés une normalisation appropriée des registres internes du filtre.

Ainsi le graphe de fluence du filtre est donnée par Fig(4-3)

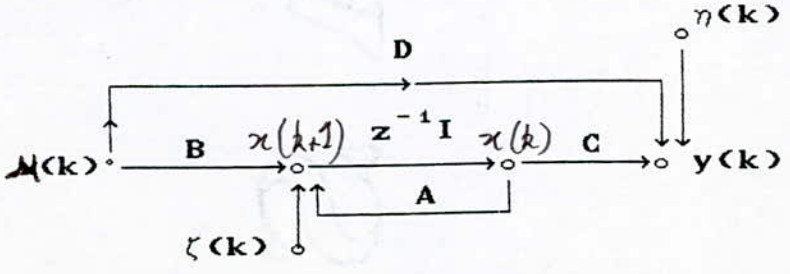


Fig (4-3) Graphe de fluence d'un filtre numérique RII

Où les sources d'erreurs de quantification sont associées aux

noeuds internes et au noeud de la sortie.

Les equations d'etat du filtre deviennent donc:

$$x(k+1) = A\ x(k) + B\ u(k) + \eta(k)$$
$$y(k) = C\ x(k) + D\ u(k) + \zeta(k) \qquad (4\text{-}47)$$

où $\zeta(k)$ est la source de bruit correspondant à la quantification des produits A x(k) et B u(k), et $\eta(k)$ est celle associée aux produits Cx(k) et Du(k). ($\eta(k)$ est un scalaire).

Comme la normalisation concerne la fonction de transfert entrée-variable d'etat F(z) qu'on schématise par:

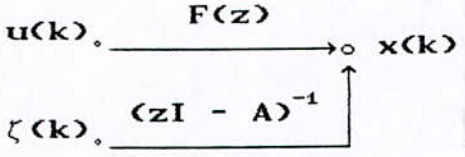


Fig (4-4)

avec $F(z) = (zI - A)^{-1}B$ (4-48)

$$\text{donc}\quad f(k) = \begin{cases} 0 & , k \leq 0 \\ A^{k-1}B & , k > 0 \end{cases}$$

Remarque

On considère que la contribution de $\zeta(k)$ est negligeable dans la gamme de x(k) où les erreurs de dépassement sont préponderantes pour un filtre non normalisé.

Le bruit de calcul concerne la fonction de transfert variable variable d'etat-sortie G(z). Le principe de superposition permet de représenter G(z) comme suit:

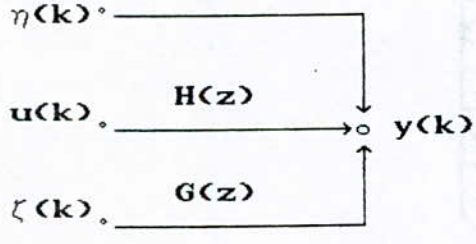


Fig (4-5)

où $G(z) = C(zI - A)^{-1}$ (4-49)

$$\text{donc}\quad g(k) = \begin{cases} 0 & , k \leq 0 \\ C\ A^{k-1} & , k > 0 \end{cases}$$

g(k) représente la réponse impulsionnelle du système variable d'etat-sortie. D'aprés les deux equations d'etat (4-1) et (4-47) l'erreur à la sortie est:

$$\Delta y(k) = \sum_{i=0}^{k-1} g(k-i)\ \zeta(i) + \eta(k)$$

$$= \sum_{i=0}^{k-1} C\, A^{k-i-1}\, \zeta(i) + \eta(k) \qquad (4\text{-}50)$$

Donc la variance de cette erreur s'exprime comme:

$$E\left[\, \Delta y^2(k) \,\right] = E\left[\left\{ \sum_{i=0}^{k-1} C\, A^{k-i-1}\, \zeta(i) + \eta(k) \right\} \left\{ \sum_{j=0}^{k-1} C\, A^{k-j-1}\, \zeta(i) + \eta(k) \right\}^T \right]$$

$$= \sum_{i=0}^{k-1} \sum_{j=0}^{k-1} \left[\, C\, A^{k-i-1} \,\right] E\left[\zeta(k)\, \zeta^T(k)\right] \left[\, C\, A^{k-j-1} \,\right]^T + E\left[\, \eta^2(k) \right] \qquad (4\text{-}51)$$

Comme les différentes composantes de $\zeta(k)$ sont incorrélées entre elles, on écrit:

$$E\left[\zeta(k)\, \zeta^T(k)\right] = \frac{q^2}{12}\, Q \qquad (4\text{-}52)$$

où $\quad Q = \mathrm{diag}\,(\nu_1\ \nu_2 \ldots \nu_N)$

ν_i est le nombre de sources de bruit (ou de multiplieurs) associées au noeud de la ieme variable d'état. Dans ce qui suit, on notera ν_s, le nombre de sources associées au noeud de la sortie.

La variance de l'erreur de sortie à la limite ($k \longrightarrow \infty$), s'écrit :

$$\sigma_y^2 = E\left[\Delta y^2(k)\right] = \frac{q^2}{12} \sum_{l=0}^{\infty} (C\, A^l)\, Q\, (C\, A^l)^T + \frac{q^2}{12}\, \nu_s \qquad (4\text{-}53)$$

En notant par Tr la trace d'une matrice, (4-53) s'écrira:

$$\sigma_y^2 = \frac{q^2}{12} \left[\sum_{l=0}^{\infty} \mathrm{Tr}\left[(C\, A^l)^T (C\, A^l)\, Q \right] + \nu_s \right]$$

$$= \frac{q^2}{12} \left[\, \mathrm{Tr}(W\, Q) + \nu_s \,\right] \qquad (4\text{-}54)$$

où $$W = \sum_{l=0}^{\infty} (C\, A^l)^T (C\, A^l) = \sum_{k=1}^{\infty} g^T(k)\, g(k) \qquad (4\text{-}55)$$

En procédant de la même maniére que pour la matrice K (voir section 4-3), on élimine la somme infinie dans (4-55) et on obtient:

$$W = A^T W\, A + C^T C \qquad (4\text{-}56)$$

L'equation (4-54) peut aussi s'ecrire:

$$\sigma_y^2 = \frac{q^2}{12}\left[\sum_{i=1}^{N} \nu_{ii} W_{ii} + \nu s\right]$$

$$= \frac{q^2}{12}\left[\sum_{j=1}^{N} \nu_{ii}\left[\sum_{k=1}^{\infty} g^2(k)\right] + \nu s\right]$$

$$= \frac{q^2}{12}\left[\sum_{j=1}^{N} \nu_{ii} \|g_j\|_2^2 + \nu s\right] \qquad (4\text{-}57)$$

Dans le cas le plus défavorable pour un filtre d'ordre N, le nombre des multiplieurs au niveau de chaque noeud est au maximum :

$$\nu s = \nu i = N+1 \qquad \text{pour tout } i$$

Dans ce cas, la variance est:

$$\sigma_y^2 = \frac{q^2}{12}(N+1)\left[Tr(W) + 1\right] \qquad (4\text{-}58)$$

Et pour le cas le plus favorable, on utilise des registres de longueur double pour accumuler les produits, donc une seule quantification est effectuée au niveau de chaque noeud, donc l'expression de la variance devient:

$$\sigma_y^2 = \frac{q^2}{12}\left[Tr(W) + 1\right] \qquad (4\text{-}59)$$

Dans tous les cas, on definit le gain de bruit comme etant le terme

$$G = Tr(W) = \sum_{i=1}^{N} W_{ii} \qquad (4\text{-}60)$$

En effectuant un changement de coordonnées dans l'espace d'etat par une transformation T, non singuliére, N x N; la matrice W se transforme d'aprés la relation (4-56) en:

$$W' = T^T W T \qquad (4\text{-}61)$$

Le gain de bruit devient alors : $G' = Tr(T^T W T)$ (4-62)

En particulier, si T est la transformation de normalisation definie précédemment par l'equation (4-44), W' aura pour composantes :

$$W'_{ij} = [T^T W T]_{ij} = \delta^2 \sqrt{K_{ii} K_{jj}}\, W_{ij} \qquad (4\text{-}63)$$

Par conséquant, la variance de la sortie devient:

$$\sigma_y^2 = \frac{q^2}{12}(N+1)\left[\delta^2 \sum_{i=1}^{N} K_{ii} W_{ii} + 1\right] \qquad (4\text{-}63)$$

Cette expression donne la variance de l'erreur de sortie en fonction des paramétres du filtre non normalisé.

L'objectif essentiel de tout ce qui va suivre (voir chapitre V) sera la minimisation de cette variance et plus précisément la quantité $\sum_{i=1}^{N} K_{ii} W_{ii}$, simplement par une transformation de coordonnées tout en conservant les contraintes de normalisation.

4-7 Remarque importante:

Si la valeur de δ est importante, la probabilité de dépassement dans la gamme des variables internes est faible, cependant, l'erreur de calcul à la sortie, d'aprés l'équation (4-64) sera importante à cause de la mauvaise correspondance entre la gamme dynamique et les nombres à représenter.

Si la valeur de δ est faible, l'erreur dûe au dépassement va prédominer dans la variance de l'erreur en sortie du filtre.

Il a étè montré [16], par simulation, que la valeur $\delta = 4$ constitue un compromis optimal entre les erreurs de dépassement et les erreurs de quantification des opérations arithmetiques.

CHAPITRE V

STRUCTURES OPTIMALES

CHAPITRE V

STRUCTURES OPTIMALES

5-1- Minimisation du gain de bruit dans le cas de registres de longueurs égales

La minimisation du gain de bruit est relative à la variance de l'erreur de calcul à la sortie du filtre. Elle n'aura de sens que si l'on suppose que le facteur de normalisation a étè convenablement choisi afin de rendre les dépassements peu probables et l'augmentation du bruit de calcul qui en découle négligeable.

Minimiser le gain de bruit d'un filtre numérique décrit par ses paramétres d'etats (A, B, C,D) et ses matrices K et W, revient à trouver une structure par une transformation T, qui change les paramétres du filtre en ($T^{-1}AT, T^{-1}B, CT, D$) et K et W en $T^{-1}K\,T^{-T}$ et $T^{T}W\,T$, respectivement, de sorte que le gain de bruit soit minimal avec la contrainte de normalisation.

Comme la quantité $\sum_{i=1}^{N} Kii\ Wii$ est invariante à une transformation diagonale, il est donc possible de minimiser d'abord le gain de bruit ensuite d'appliquer une transformation de normalisation à la structure trouvée.

Actuellement, il existe deux méthodes de base permettant de construire les structures à gain de bruit minimal, la méthode de **Mullis-Roberts** [16] et la méthode de **Hwang**[17]. Bien que ces deux méthodes sont différentes du point de vue du traitement mathématique du problème, elles aboutissent chacune à un même gain minimal.

A - Methode de Mullis-Roberts [16]

Cette méthode traite deux cas; le cas des registres de longueurs égales et celui des registres de longueurs optimales. On se limitera à l'étude du premier cas uniquement.

La méthode consiste à rechercher le gain de bruit optimale à partir des conditions de son existence qui permettent d'etablir la transformation réalisant la structure optimale.

Si le pas de quantification $q = 1.2^{-B+1}$ et B le nombre de bits des registres considerés, l'expression de la variance en sortie

(4-64) devient :

$$\sigma_y^2 = \frac{N\ (N+1)}{3}\left(\frac{\delta}{2^B}\right)^2\left[\frac{1}{N}\sum_{i=1}^{N} K_{ii}W_{ii}\right] + \frac{N+1}{3.2^{2B}} \qquad (5\text{-}1)$$

Si N,δ et B sont donnés alors il suffit de minimiser le gain de bruit donné par:

$$G = \sum_{i=1}^{N} K_{ii}W_{ii}$$

A cet effet ,on utilise la propriété des matrices K et W : dans le cas où celles-ci sont réelles, symétrique et définies positives, l'inégalité suivante est vérifiée :

$$\left[\frac{1}{N}\sum_{i=1}^{N} K_{ii}W_{ii}\right] \geq M\alpha^2 \qquad (5\text{-}2)$$

où

$$M\alpha = \frac{1}{N}\sum_{i=1}^{N} \mu_i \qquad (5\text{-}3)$$

et $\mu_1^2, \mu_2^2, \ldots, \mu_N^2$ sont les valeurs propres de la matrice produit KW, et dont les racines carrées sont appelées les modes de second ordre [1].

Pour que la borne inférieure soit atteinte, il est nécéssaire et suffisant de satisfaire les deux conditions suivantes [1],[16]:

1) $D\bar{o}^1\ K\ D\bar{o}^1 = Do\ W\ Do$ où Do est une matrice diagonale
autrement dit $K = D'W\ D'$ telle que $D' = Do^2$

2) $K_{ii}W_{ii} = K_{jj}W_{jj} \quad \forall\ i,j = 1,\ldots,N$

Si K est normalisée ($K_{ii} = 1/\delta^2$) alors $W_{ii} = W_{jj} \quad \forall\ i,j = 1,\ldots,N$

La condition 1) implique que si $K' = D\bar{o}^1\ W\ D\bar{o}^1 = W'$, alors [6]

$$Tr(K') = Tr(W') = \sum_{i=1}^{N} \mu_i = N\ M\alpha \qquad (5\text{-}4)$$

La condition 2) implique l'égalité suivante:

$$K'_{ii}W'_{ii} = K_{ii}W_{ii} = K_{jj}W_{jj} = \text{constante} = \alpha \quad \forall\ i,j=1,..,N \qquad (5\text{-}5)$$

donc $Tr(K') = N\ M\alpha = N\sqrt{\alpha}$ et $\alpha = M\alpha^2$ (5-6)

d'où $\sum_{i=1}^{N} K'_{ii}W'_{ii} = \alpha\ N = N\ M\alpha^2$

donc on a bien l'egalité $\frac{1}{N}\sum_{i=1}^{N} K'_{ii}W'_{ii} \equiv M\alpha^2$ (5-7)

Dans ces conditions pour tout ε positif infiniment petit, il

existe une transformation To pour laquelle on a :

$$0 \leq \frac{1}{N} \sum_{i=1}^{N} (\, To^{-1} K' \, To^{-T})_{ii} \, (T^T \, W' \, T)_{ii} - M\alpha^2 < \varepsilon \qquad (5\text{-}8)$$

Pour trouver To, il suffit de diagonaliser les matrices K et W, et d'appliquer une succession de rotations à la matrice diagonale W' afin de rendre ses élèments identiques. Le nombre de ces rotations depend du choix de ε .

Le gain de bruit minimal obtenu est :

$$G_{min} = \frac{1}{N} \left[\sum_{i=1}^{N} \mu_i \right]^2 \qquad (5\text{-}9)$$

B - Méthode de Hwang [17]

Dans cette methode il s'agit de construire la transformation T en maintenant la contrainte de normalisation.

Soit T la transformation qui minimise le gain de bruit et qu'on factorise en:

$$T = R \, S \qquad (5\text{-}10)$$

avec R : une matrice orthogonale ($R \, R^T = I$)

et S : une matrice definie positive

donc en diagonalisant S, T s'ecrit :

$$T = R \, Ro \, \Lambda \, Ro^T = R_1 \, \Lambda \, Ro^T \qquad (5\text{-}11)$$

où $\Lambda = \mathbf{diag}\,(\lambda_1 \ldots \lambda_N)$

et Ro : matrice des vecteurs propres de S, elle est othogonale

Le gain de bruit est alors:

$$\begin{aligned} G_1 = Tr(\, T^T \, Wo \, T \,) &= Tr(\, T^T \, T \, Wo \,) \\ &= Tr(\, \Lambda^2 \, R_1{}^T \, Wo \, R_1) \\ &= \sum_{i=1}^{N} \lambda_i^2 \, r_i^2 \end{aligned} \qquad (5\text{-}12)$$

où $r_i^2 = (R_1{}^T \, Wo \, R_1)_{ii}$

Et T doit assurer la normalisation donc:

$$T^{-1} \, Ko \, T^{-T} = \begin{bmatrix} 1_1 & X \\ X & \ddots{}_1 \end{bmatrix} \quad (\delta = 1) \qquad (5\text{-}13)$$

Sachant que [15]

$$\det Wo \leq \prod_{i=1}^{N} r_i^2 \qquad (5\text{-}14)$$

et

$$\det Ko \leq \prod_{i=1}^{N} \lambda_i^2$$

Et que la moyenne arithmétique est supérieure ou égale à la moyenne géometrique, ce qui permet d'écrire:

$$\frac{1}{N}\left[\sum_{i=1}^{N}\lambda_i^2\ r_i^2\right] \geq \left[\prod_{i=1}^{N}\lambda_i^2\ r_i^2\right]^{1/N} \qquad (5\text{-}15)$$

alors

$$G_1 = Tr(\ T^T\ Wo\ T\ ) \geq N\ \left(\prod_{i=1}^{N} r_i^2\right)\left(\prod_{i=1}^{N}\lambda_i^2\right) \geq N\ [\det\ KoWo]^{1/N} \qquad (5\text{-}16)$$

Comme les valeurs propres du produit KoWo sont invariantes à une transformation d'état T, cette borne inférieure de (5-16) est aussi invariante.

L'egalité dans (5-16) est realisée si et seulement si:

1- KoWo est symétrique.

2- Ko^{-1} et Wo sont equivalents à une constante prés.

Pour minimiser le gain :

1- On normalise d'abord le systéme par To;

$$K_1 = \bar{To}^{1}\ Ko\ \bar{To}^{T} = I \quad \text{donc} \quad To\ To^T = Ko \qquad (5\text{-}16)$$

$$W_1 = To^T\ W\ \ To$$

2- On applique la matrice T décrite précédemment, donc:

$$G_1 = Tr(\ T^T\ W_1\ T\ ) = Tr(\ \Lambda^2\ R_1{}^T\ W_1\ R_1) = \sum_{i=1}^{N}\lambda_i^2\ r_i^2 \qquad (5\text{-}17)$$

Avec la contrainte

$$Ro\ \Lambda^{-2}\ Ro^T = \begin{bmatrix} 1 & X \\ X & \ddots\ 1 \end{bmatrix} \qquad (5\text{-}18)$$

Dans ces conditions, on obtient:

$$\sum_{i=1}^{N} 1/\lambda_i^2 = N \qquad (5\text{-}19)$$

$$\prod_{i=1}^{N}\lambda_i^2 \geq 1 \qquad (5\text{-}20)$$

3- En utilisant la fonctions de Lagrange donnée par:

$$\mathcal{L}(\lambda) = \sum_{i=1}^{N}\lambda_i^2\ r_i^2 + \alpha\left[\sum_{i=1}^{N} 1/\lambda_i^2 - N\right] \qquad (5\text{-}21)$$

qu'on minimise par rapport à λ_i et α pour obtenir, en considèrant

la contrainte (5-19):

$$\lambda_i = \left[\frac{\sum_{i=1}^{N} r_i}{N \; r_i} \right]^{1/2} \qquad (5\text{-}22)$$

donc le gain minimal est:

$$G_{min} = Tr(\; T^T \; W_1 \; T \;) = \frac{1}{N} \left(\sum_{i=1}^{N} r_i \right)^2 \qquad (5\text{-}23)$$

Comme

$$\det(\; K_o W_o) = \det(\; W_1) = \prod_{i=1}^{N} \mu_i^2 \qquad (5\text{-}24)$$

où μ_i sont les modes du second ordre de KoWo et

$$\det(R_1^T \; W_1 \; R_1) = \det(W_1) \qquad (5\text{-}25)$$

Si $R_1^T W_1 R_1$ est une matrice symétrique définie positive, alors:

$$\sum_{i=1}^{N} r_i^2 \geq \sum_{i=1}^{N} \mu_i^2 \qquad (5\text{-}26)$$

avec r_i et μ_i positifs pour tout i.

L'egalité est atteinte dans (5-26) si la matrice $R_1^T W_1 R_1$ est diagonale, c'est à dire que R_1 est la matrice des vecteurs propres de W1.

Donc on a : $G_{min} = \frac{1}{N} \left(\sum_{i=1}^{N} \mu_i \right)^2 \geq N \; [\det \; K_o W_o]^{1/N}$ (5-27)

La borne inferieure est atteinte si et seulement si:

$$\mu_i^2 = \mu_j^2 \qquad \text{pour tous i, j} \qquad (5\text{-}28)$$

4- On detérmine Ro sachant la contrainte (5-18); étant donné que Ro est une matrice orthogonale, elle est décomposable en facteurs de matrice de rotation élèmentaires de la forme:

$$R_i = \begin{bmatrix} I & \vdots & 0 & \vdots & 0 \\ \cdots & & \cdots & & \cdots \\ 0 & \vdots & \cos\psi_i & \vdots & \sin\psi_i \\ \cdots & & \cdots & & \cdots \\ 0 & \vdots & -\sin\psi_i & \vdots & \cos\psi_i \\ \cdots & & \cdots & & \cdots \\ 0 & \vdots & 0 & \vdots & I \end{bmatrix} \qquad i = 1,2,\ldots,N \qquad (5\text{-}29)$$

(N-1) transformations sont nécessaires pour construire Ro, qui vérifie (5-18).

5- Finalement on construit la transformation de minimisation T :

$$T = T_o\, R_1 \Lambda R_o^T = T_o\, R_1 \Lambda (R_o \ldots R_3 R_2)^T \qquad (5\text{-}30)$$

qu'on applique aux paramétres du filtre (Ao,Bo,Co,Do).

Cette procédure est explicitée dans la section (6-2) pour le cas de filtre d'ordre 2.

5.2 Algorithmes de calcul des matrices K et W

Les matrices K et W peuvent être calculées à partir de leurs expressions:

$$K = \sum_{k=0}^{+\infty} (A^k B)(A^k B)^T = A K A^T + B B^T \qquad (5\text{-}31)$$

$$W = \sum_{k=0}^{+\infty} (C A^k)^T (C A^k) = A^T W A + C^T C \qquad (5\text{-}32)$$

La procédure qui calcule K calcule aussi W en remplaçant A par A^T et B par C^T. Soient les deux algorithmes suivants [1],[16]:

1er Algorithme

C'est un algorithme simple et efficace, où à la ieme itération, sont calculés 2^i termes de la somme infinie (5-31).

Cependant, la convergence de la procédure dépend de la position des pôles du filtre par rapport au cercle unité.

La procédure est la suivante:

(1)- Intialiser : $F \longleftarrow A$ et $K \longleftarrow B B^T$

(2)- Calculer : $K \longleftarrow F K F^T + K$

(3)- Elever F au carré : $F \longleftarrow F^2$

(4)- Refaire(2) et (3) jusqu'à F = 0

2eme Algorithme

Cet algorithme cherche d'abord une forme directe (canonique) intermédiaire, puis calcule la matrice de convariance R de cette structure qui est symétrique, et du type de Toeplitz dont la 1ere ligne est [ro,r1,....,rN-1]. Ensuite, il calcule la matrice T qui permettra de revenir à la structure initiale en donnant la matrice K.

Les étapes de cette procédure sont les suivantes:

(1) Pour (A,B) donnés, calculer les coefficients a_k du polynôme caracteristique : $\det(\lambda I - A) = 0$

(2) Initialiser X(0)=0 et calculer $X(k+1) = AX(k) + Ba_k$ pour $0 < k < N$

(3) Construire T = [X(N)...X(1)]

(4) Calculer r(0),..., r(N) satisfaisant l'equation suivante :

$$r(i) + \sum_{j=1}^{N} a_j\, r(|i-j|) = \begin{cases} 1 & , i=0 \\ 0 & , 0 \le i < N \end{cases}$$

(5) Construire la matrice R (NxN) ayant la forme de Toeplitz [5]:

$$R_{ij} = r(|i-j|)$$

(6) Calculer K à partir de T et R par $K = T\,R\,T^T$

Remarque:

Le deuxiéme algorithme n'est pas iteratif, donc il calcule K en un nombre d'étapes fini et connu à priori.

5.3 Propriétés des structures à gain de bruit minimal

5.3.1 Invariance par rapport à une transformation fréquentielle

Une propriété remarquable des structures optimales est la conservation des modes du second ordre par une transformation fréquentielle. En effet, comme les modes du second ordre d'un filtre H(z) définissent le gain en bruit minimal d'aprés l'equation (5-9), la famille de filtres G(z) telle que:

$$G(z) = H(F(z)) \qquad (5\text{-}33)$$

Où F(z) est une transformation fréquentielle d'ordre m de la forme:

$$F(z) = \pm \prod_{i=1}^{m} \left(\frac{z - \alpha_i^*}{1 - \alpha_i z} \right) \; ; \; |\alpha_i| < 1 \qquad (5\text{-}34)$$

et a pour modes du second ordre m duplications des modes de H(z).

En particulier, si la transformation est du type passe bas-passe bas dont l'ordre est l'unité, H(z) et G(z) ont le même gain minimal en bruit, ce qui signifie que le bruit à la sortie des structures optimales est indépendant de la largeur de la bande passante des filtres [18] (Voir chapitre VII).

Par contre, pour les structures directes le gain de bruit dépend de la bande passante du filtre du fait qu'il s'exprime en fonction des pôles de la transformation fréquentielle par [1] :

$$\text{Gain de bruit (forme directe)} = \frac{P(\alpha)}{(1-\alpha^2)^{2N-2}} \qquad (5\text{-}35)$$

où $P(\alpha)$ est un polynôme en α de degrés $4(N-1)$.

5.3.2 Expression compléte du bruit de calcul

Dans certaines structures, il existe d'autres types de variables (ou noeuds) qui n'apparaissent pas dans les équations d'état du filtre (dans les structures en treillis, par exemple) et qui introduisent des erreurs de calcul supplementaires [18].

Donc le bruit de calcul total de la sortie dépendent des N

variables d'etat et des M variables correspondant aux noeuds non étatiques (qui ne sont pas issus d'un élement de retard). L'expression devient alors:

$$\sigma^2_{Total} = \frac{\delta^2 q^2}{12}\left\{\sum_{i=1}^{N} K_{ii} W_{ii} + \sum_{j=1}^{M} \|f_j\|_2^2 \; \|g_j\|_2^2 \right\} \qquad (5\text{-}36)$$

(dans le cas d'un accumulateur de longueur double).
où fi et gi sont les réponses impulsionnelles entrée-variable d'etat et variable d'etat -sortie, respectivement.

5.3.3 Gain minimal et performances des réalisations

Bien que les structures optimales dans l'espace d'etat offrent des gains de bruit minimums, leurs réalisations pratiques s'avérent complexes lorqu'il s'agit des filtres d'ordre N assez grand.

En effet, de telles structures requièrent $(N+1)^2$ multiplications pour calculer chaque échantillon de la sortie; ce qui constitue une augmentation de N^2 multiplications par rapport aux strctures canoniques.

Pour réduire le nombre des multiplications dans les réalisations à bruit minimal, on utilise de multiples procédés, comme la décomposition d'un filtre d'ordre N en sections de structures optimales d'ordre 2 (ou 1) et leurs connexions en paralléle ou en cascades, ce qui réduit le nombre des multiplications à (4N+1) et présente un bon compromis entre le gain de bruit de calcul et la simplicité des calculs.

D'autres procédés ont étè élaborés dans le but de trouver d'autres structures de compromis, comme les procédés de tridiagonalisation de la matrice A globale, qui donnent des structures de (5N-1) multiplications, les structures modulaires qui nécessitent (3N+3) multiplications [19], les structures à coefficients en puissance de deux [20],etc...[21],[22],[23],[24].

Dans cette étude on s'intéressera surtout au sectionnement de l'espace d'etat en cellules du second ordre connectées en paralléles.

5.4 Minimisation de la sensibilité aux coefficients

Il est important dans les considérations pratiques de synthétiser des filtres dont la fonction de transfert présente une faible sensibilité aux variations de ses coéfficients dues à la limitation de leur representation arithmétique.

Pour fonction de transfert H(z)

$$H(z) = \frac{\sum_{i=0}^{N} b_i \; z^{-i}}{\sum_{i=0}^{N} a_i \; z^{-i}} \qquad (5\text{-}37)$$

qui est décrite par un modéle d'etat contrôlable et observable

$$H(z) = D + C\,(zI-A)^{-1}B$$

Le fait de représenter ces coefficients par des mots de longueurs finies va faire dévier H(z) de ses performances souhaitées.

Pour évaluer ces déviations dans l'espace d'etat, on exprime les sensibilités de H(z) par rapport à chaque composantes des paramétres d'etat individuellement par [25][1]:

$$S_{a_{ij}}(z) = \frac{\partial H(z)}{\partial a_{ij}} \;,\; S_{b_j}(z) = \frac{\partial H(z)}{\partial b_j} \;,\; S_{c_j}(z) = \frac{\partial H(z)}{\partial c_j} \qquad (5\text{-}38)$$

et qu'on peut écrire en fonction des fonctions de transfert F et G définies dans la section (4-7):

$$\begin{aligned} S_{a_{ij}}(z) &= G_i(z)\,F_j(z) \\ S_{b_i}(z) &= G_i(z) \\ S_{c_i}(z) &= F_i(z) \end{aligned} \qquad (5\text{-}39)$$

où $\quad G_i(z) = [C\,(zI-A)^{-1}]_i$

et $\quad F_i(z) = [(zI-A)^{-1}B]_i$

et d'aprés les equations (4-41) et (4-55), on définit une mesure de la sensibilité par [25]:

$$M = M_A + M_B + M_C \qquad (5\text{-}40)$$

avec

$$M_B = \frac{1}{2\pi j} \oint G^T(z)\,G(z)\,z^{-1}dz = Tr(W) = \sum_{j=1}^{N} W_{jj}$$

$$M_C = \frac{1}{2\pi j} \oint F^T(z)\,F(z)\,z^{-1}dz = Tr(K) = \sum_{j=1}^{N} K_{jj}$$

$$M_A = M_B\,M_C \geq \frac{1}{2\pi j} \oint \sum_{i=1}^{N}\sum_{j=1}^{N} |S_{a_{ij}}(z)|^2\, z^{-1}\,dz$$

$$= Tr(K)\,Tr(W) = \sum_{i=1}^{N} K_{ii} \sum_{j=1}^{N} W_{jj}$$

Si on applique une transformation de normalisation telle que:

$$Tr(K') = N \qquad (5\text{-}41)$$

alors l'expression de la mesure de sensibilité est:

$$M = (N+1)\,Tr(W') + N$$
$$M = (N+1)\,G' + N \qquad (5\text{-}42)$$

où G' est le gain de bruit de la nouvelle structure du filtre.

On remarque que la minimisation du gain entraine celle de M donc pour un gain à minimal on a (d'aprés l'equation (5-9)).

$$M = \frac{(N+1)}{N}\left[\sum_{i=1}^{N}\mu_i\right]^2 + N \qquad (5\text{-}43)$$

En conclusion, la minimisation du gain de bruit par une transformation T pour un filtre normalisé implique directement la minimisation de la mesure de la sensibilité. En conséquence, une fonction de transfert calculée à partir des coefficients d'une structure optimale normalisée présentera le moins de distorsions possibles dans le cas de l'utilisation de registres de longueurs finies pour représenter les coefficients. Cela a étè confirmé pour l'exemple de simulation dans le chapitre VII.

CHAPITRE VI

STRUCTURES DECOMPOSEES OPTIMALES

CHAPITRE VI

STRUCTURES DECOMPOSEES OPTIMALES

6-1 Introduction

L'inconvenient majeur des structures à gain de bruit minimal étudiées dans le chapitre précédent, est la complexité de leurs réalisations matérielles dûe au grand nombre des multiplieurs qu'elles nécessitent. Pour un filtre d'ordre N, le nombre des multiplications par échantillons de la sortie est $(N+1)^2$ pour les structures optimales, par contre il est de (2N+1) seulement pour les structures directes (canonique).

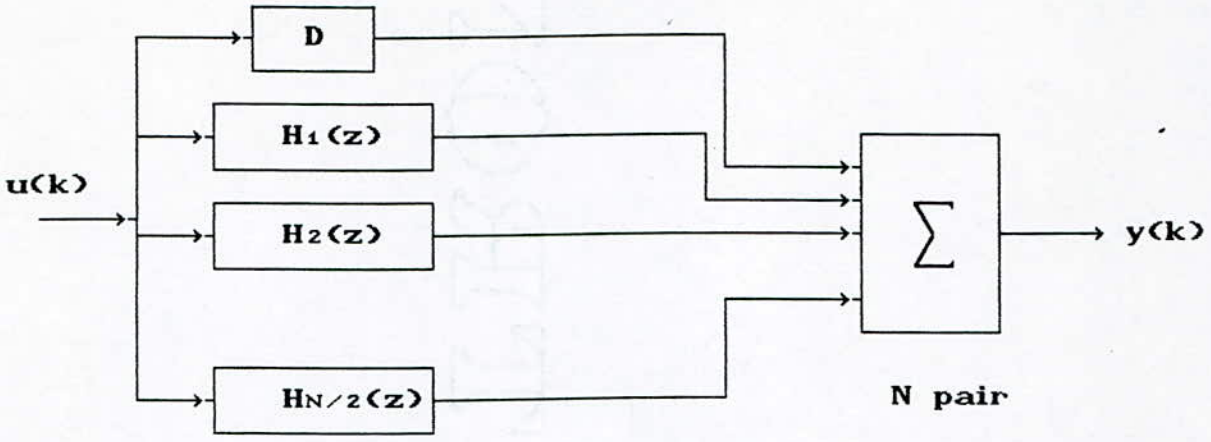
Un moyen efficace pour obtenir des structures de compromis entre le nombre des multiplications et le gain de bruit de calcul est la décomposition du filtre en cellules du second ordre regroupant chacune un couple de pôles complexes conjugués et eventuellement une cellule du premier ordre si N est impair. La connexion entre ces différentes cellules se fait soit en paralléle soit en cascade. Dans ce projet on se limite à l'étude du cas parallèle pour lequel la sortie globale du filtre est simplement la somme des sorties des différentes cellules, il en est de même pour le gain de bruit global. Dans ce cas la fonction de transfert du filtre s'ecrit:(Fig(6-1))

$$H(z) = D + \sum_{i=1}^{N/2} H_i(z) \qquad \text{(pour N pair)} \qquad (6\text{-}1)$$

où $H_i(z)$ est la fonction de transfert de la cellule i du second second ordre de la forme:

$$H_i(z) = C_i(zI - A_i)^{-1} B_i \qquad (6\text{-}2)$$

avec A_i, B_i et C_i les paramétres d'état 2x2, 2x1 et 1x2 (resp.) et D est le chemin direct entrée-sortie du filtre .



Fig(6-1) Décomposition en cellule du second ordr d'un filtre RII

Dans la simulation des structures de compromis (chapitre VII), deux types de structures décomposées sont étudiés:

1- Structure décomposée canonique où chaque cellules a une structure canonique (ou directe) du second ordre; elle nécéssite environ (2N+1) multiplications par échantillon de sortie.

2- Structure décomposée optimale où chaque cellule a une structure à gain minimal, elle nécessite environ (4N+1) multiplications. Bien que cette structure augmente le gain minimal du filtre, elle simplifie considérablement sa conception.

Dans ce présent chapitre, sont developpés les procédès d'optimisation des cellules du second ordre.

Remarque

Il est facile de vérifier que pour la cellule du premier ordre, si elle existe (N impair), elle est optimale.

6-2 Structures optimales du second ordre

On considère le filtre d'ordre 2 dont la fonction de transfert est :

$$H(z) = C\,(zI-A)^{-1}B \qquad (6\text{-}3)$$

où

$$A=\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad , B=\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \quad , \quad C=\begin{bmatrix} c_1, & c_2 \end{bmatrix}$$

et d'aprés les deux conditions nécessaires et suffisantes pour avoir une structure (A, B, C) dont la variance de l'erreur de sortie est minimale (Voir la section (5-1)).

C1 : $K = D_o^{-1} W D_o^{-1}$ (Do diagonale)

et

$$C2 : K_{ii} W_{ii} = K_{jj} W_{jj} \quad \text{pour tous i,j} \qquad (6\text{-}4)$$

et avec la contrainte de normalisation (4-43), on a alors pour C2

$$W_{ii} = W_{jj} \qquad \text{pour tous i,j} \qquad (6\text{-}5)$$

et avec la condition C1, Do doit être de la forme [26]:

$$Do = \rho\, I$$

d'où

$$W = \rho^2\, K \qquad (6\text{-}6)$$

Si on pose la matrice M telle que

$$M = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad (6\text{-}7)$$

on peut écrire (6-6) [26] :

$$W = \rho^2\, M\, K\, M \qquad (6\text{-}8)$$

En éxprimant K et W en fonction des paramétres d'état (A,B,C), l'équation (6-8) entraine:

$$A^T = M\, A\, M$$
$$C^T = \rho\, M\, B \qquad (6\text{-}9)$$

En termes d'élements, les conditions d'optimalité (6-3) deviennent [1]:

- 1- $a_{11} = a_{22}$
  2- $b_1\, c_1 = b_2\, c_2$ (6-10)
  avec la contrainte de normalisation (4-43).

Il existe plusieurs structures du second ordre qui vérifient ces trois contraintes d'optimalité, parmi lesquelles apparaissent:

1- Structures de BOMAR [21]

La fonction de transfert d'un filtre du second ordre donnée par (6-2) peut aussi s'écrire

$$H(z) = \frac{q_1\, z^{-1} + q_2\, z^{-2}}{1 + p_1\, z^{-1} + p_2\, z^{-2}} \qquad (6\text{-}11)$$

où q_1, q_2, p_1 et p_2 sont des constantes réelles.

Pour trouver les élements des paramétres d'etat (A, B, C) de la structure optimale correspondant à H(z); on identifie (6-11) à l'équation (6-2) developpée, on obtient 4 équations à 8 inconnues:

$$q_1 = c_1b_1 + c_2b_2$$
$$q_2 = c_1b_2a_{12} + c_2b_1a_{21} - c_1b_1a_{22} - c_2b_2a_{11} \qquad (6\text{-}12)$$
$$p_1 = -(a_{11} + a_{22})$$
$$p_2 = a_{11}a_{22} - a_{21}a_{12}$$

Ces équations permettent d'avoir 4 réalisations equivalentes qui diffèrent seulement par les signes des coefficients ($a_{11}, a_{12}, a_{21}, a_{22}, b_1, b_2, c_1, c_2$). Deux autres équations sont fournies par les conditions d'optomalité (6-10).
La contrainte de normalisation (4-46) pour laquelle :

$$K_{11} = K_{22} = 1/\delta^2 \qquad (6\text{-}13)$$

donne à partir de la relation:

$$K = A\,K\,A^T + B\,B^T \qquad (6\text{-}14)$$

deux autres équations.

Le systéme de 8 équations à 8 inconnues ainsi formé donne alors les elements des paramétres de la structure optimales du second ordre en fonction des coefficients δ, q_1, q_2, p_1 et p_2. La solution du systéme de la procédure de BOMAR est presentée par l'algorithme de calcul suivant:[1]

1) Calcul des quantités intermediaires $v_1, v_2, \ldots, v_8$ telles que :

$$v_1 = \left[\frac{q_2}{q_1}\right], \quad v_2 = (v_1^2 - p_1v_1 + p_2), \quad v_3 = v_1 - v_2, \quad v_4 = v_1 + v_2$$

$$v_5 = p_2 - 1, \quad v_6 = p_2 + 1, \quad v_7 = v_5\left[v_6^2 - p_1^2\right], \quad v_8 = \left[\frac{p_1}{2}\right]^2 - p_2$$

2) Calcul des élèments des paramétres d'état

$$a_{11} = a_{22} = -p_1/2$$
$$b_1 = [v_7\,\delta^{-2}/(2p_1v_3 - v_6(1+v_3^2))]^{1/2}$$
$$b_2 = [v_7\,\delta^{-2}/(2p_1v_4 - v_6(1+v_4^2))]^{1/2}$$
$$a_{21} = [(\delta^2b_2^2 + v_5)v_8/(\delta^2b_1^2 + v_5)]^{1/2}$$
$$a_{12} = v_8/a_{21}$$
$$c_1 = q_1/2b_1$$
$$c_2 = q_1/2b_2$$

2- Structures de BARNES [27]

A partir de l'expression de la fonction de transfert de (6-11), on construit une structure directe (A,B,C) telle que :

4) Determiner la matrice Ro par la relation

$$Ro\ \Lambda^{-2} Ro^{T} = \begin{bmatrix} 1 & x \\ x & 1 \end{bmatrix} \quad , \text{car } K_1 = I$$

on trouve $Ro = \begin{bmatrix} \sqrt{1/2} & -\sqrt{1/2} \\ \sqrt{1/2} & \sqrt{1/2} \end{bmatrix}$, par exemple.

En considérant le paramétre de normalisation δ et le produit $\Lambda\ Ro^{T}$

on a : $T_1 = \delta\ \Lambda\ Ro^{T} = \frac{\delta}{2} \begin{bmatrix} (1+\mu)^{1/2} & -(1+\mu)^{1/2} \\ (1+1/\mu)^{1/2} & (1+1/\mu)^{1/2} \end{bmatrix}$ où $\mu = \frac{\mu_2}{\mu_1}$

5) Appliquer T_1 à K_2 et W_2 pour obtenir K_{min} et W_{min} :

$$K_{min} = \frac{1}{\delta^2} \begin{bmatrix} 1 & \frac{\mu_1-\mu_2}{\mu_1+\mu_2} \\ \frac{\mu_1-\mu_2}{\mu_1+\mu_2} & 1 \end{bmatrix}$$

$$W_{min} = \frac{\delta^2}{4} \begin{bmatrix} (\mu_1+\mu_2)^2 & \mu_2^2 - \mu_1^2 \\ \mu_2^2 - \mu_1^2 & (\mu_1+\mu_2)^2 \end{bmatrix}$$

6) Calculer la transformation globale de minimisation:

$$T = To\ R_1\ T_1$$

7) Appliquer T à (A, B, C) pour avoir la structure optimale (Amin, Bmin,Cmin) . Le gain minimale est donc :

$$G_{min} = \frac{1}{2} (\mu_1 + \mu_2)^2$$

6.3 Conclusion

Les trois procédures décrites, ci-dessus, permettent chacune un certain nombre de structures optimales du second ordre (par exemple, 4 structures pour la procédure de BOMAR) et aboutissent toutes à un même gain optimal, d'où à une même de mesure de la sensibilité de la fonction de transfert du filtre globale.

Le gain total issu de toutes les cellules de structures optimales du second ordre en paralléles d'un filtre d'ordre N est:

$$G_{total} = \sum_{i=1}^{N/2} [G_{min}]_i$$

Remarque

Pour une cellule du 1er ordre dont la fonction de transfert est $H_1(z) = \frac{\alpha}{z-\beta}$

le gain $G_1 = \frac{\alpha^2}{(1-\beta^2)^2}$, (pour δ=1). Il est invariant.

CHAPITRE VII

RESULTATS ET INTERPRETATIONS

CHAPITRE VII

RESULTATS ET INTERPRETATIONS

7.1 Introduction

Dans ce chapitre, est developpée une étude comparative entre les pérformances de la structure canonique et de la structure décomposée optimale en parallèle d'un filtre numérique RII au moyen des simulations sur ordinateur avec l'arithmétique de la virgule fixe. Pour chaque structure, sont presentés les effets de la limitation de la précision de la représentation des coéfficients du filtre et des résultats des opérations arithmétiques sur la qualité du filtrage, ainsi que le tracé de la variance de l'erreur de sortie en fonction de la longueur des mots binaires utilisée et du type d'accumulation des produits (simple ou double). De même, les tracés de cette variance sont comparés à ceux obtenus théoriquement. La sensibilité de la fonction de transfert (donc de la réponse fréquentielle) à la précision des coefficients du filtre est mise en relief par des graphes illustratifs. La variation du gain de bruit de calcul en fonction de la largeur de la bande passante du filtre est presentée afin de mettre en évidence l'effet de la transformation fréquentielle d'ordre 1 sur la performance de chaque type de structure.

Ce chapitre se termine par une comparaison globale suivie d'une conclusion.

Le filtre numérique RII simulé, dans tout le chapitre, est un filtre passe-bas de BUTTERWORTH d'ordre 8 dont la fréquence de coupure normalisée est $f_c = 0.125$ ($\omega_c = 0.25\ \Pi$).

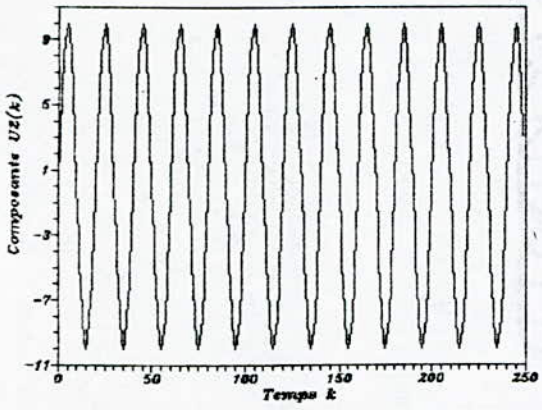
Les signaux d'entrée générés et utilisés sont:

- Un signal S1 qui est la somme de deux sinusoides dont l'une des fréquences se trouve dans la bande passante du filtre et l'autre dans sa bande d'affaiblissement (voir Fig-(7-1-a,b,c)).

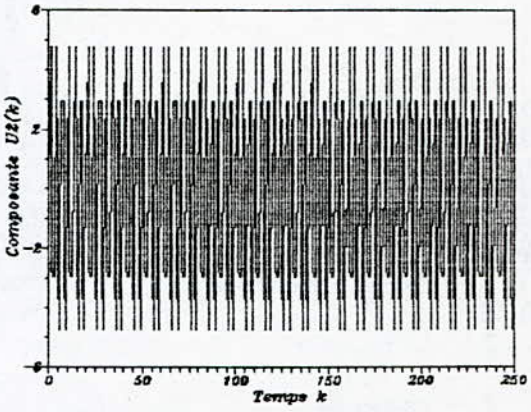
Ce signal est donné par:

$$U(k) = 10 \sin(2\ \Pi\ f_1\ k) + 5 \sin(2\ \Pi\ f_2\ k) \qquad (7-1)$$

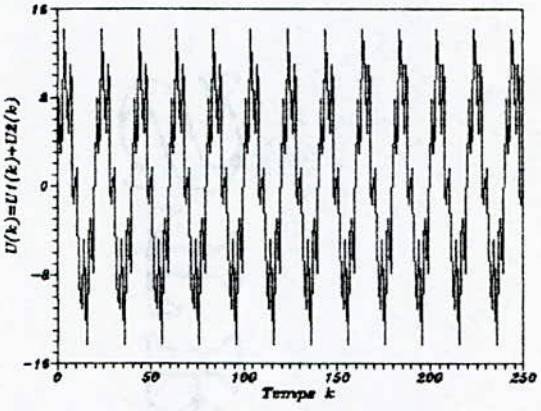
avec $f_2 = 0.05$ ($\omega_1 = 0,1\ \Pi$) et $f_2 = 0.3$ ($\omega_2 = 0.6\ \Pi$)



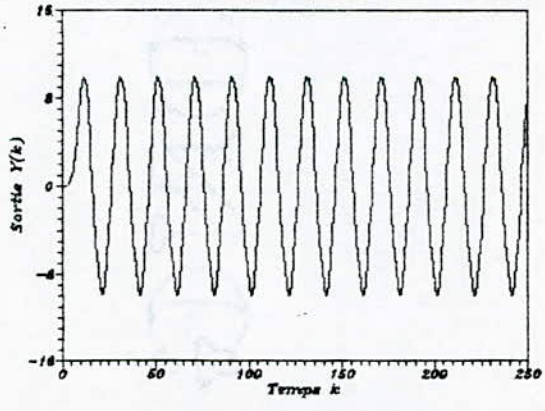
(a) COMPOSANTE DE FREQUENCE NORMALISEE 0.05



(b) COMPOSANTE DE FREQUENCE NORMALISEE 0.3

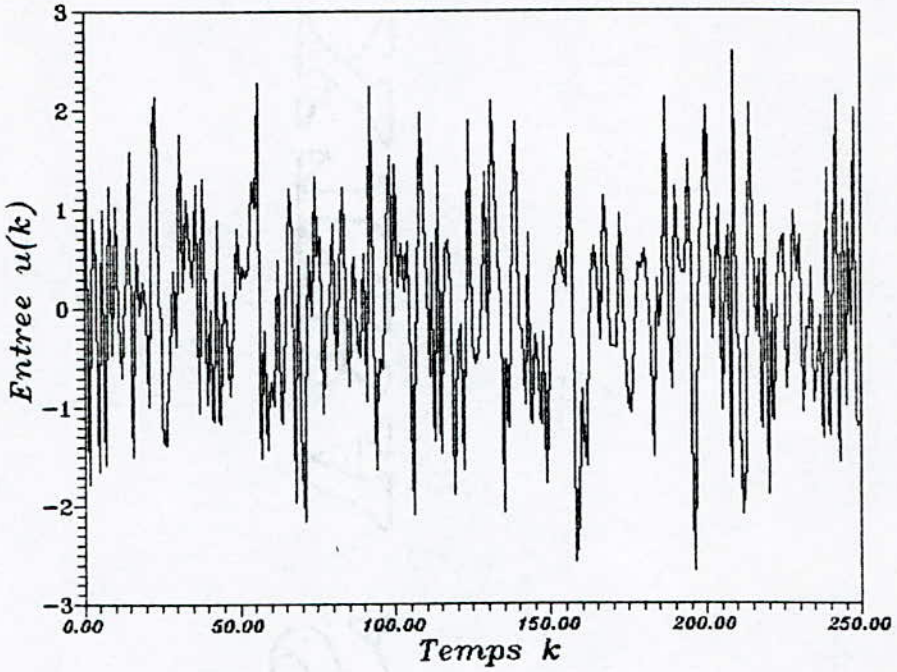


(c) Signal d'entree du filtre
(somme des composantes f1=0.05 et f2=0.3)

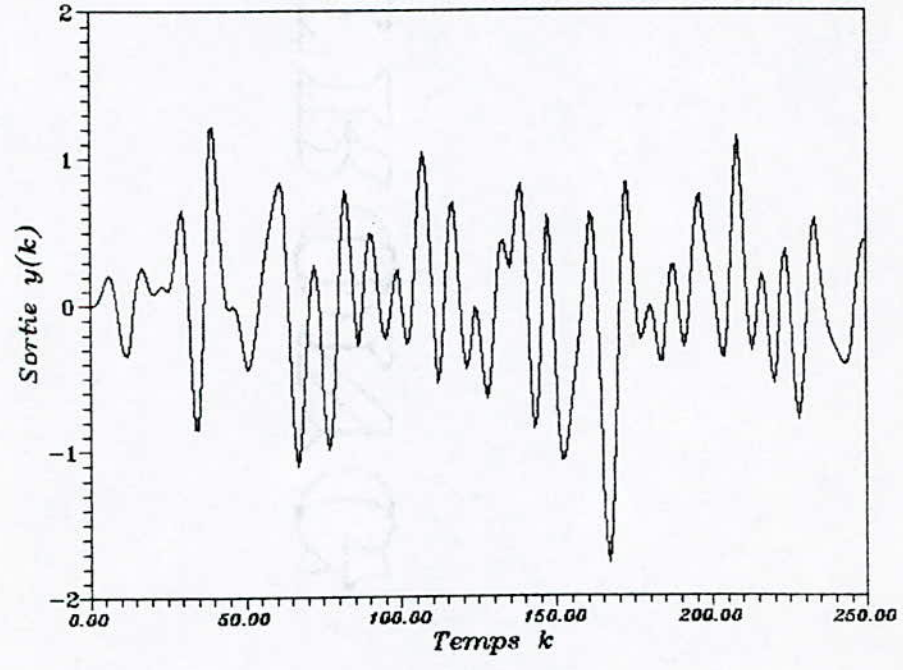


(d) signal de sortie (54 bits)

FIG-(7-2) FILTRAGE D'UN SIGNAL SOMME DE 2 SINUSOIDES
AVEC UNE PRECISION INFINIE (54 Bits)



(a) Signal non filtré



(b) Signal Filtré

FIG-(7- 2) FILTRAGE D'UN SIGNAL BRUIT BLANC CENTRE DE VARIANCE UNITE

- Un signal S2 alèatoire de distribution gaussienne centrée de variance l'unité simulant un bruit blanc (voir Fig(7-2-a)).

7.2 Performances d'un filtre numérique avec une structure canonique

La fonction de transfert du filtre utilise est donnée par:

$$H(z)= \frac{\sum_{i=0}^{8} b_i\, z^{-i}}{1 + \sum_{i=1}^{8} a_i\, z^{-i}} \qquad (7-2)$$

dont les coefficients du numérateur sont:

b_0= 1.0791128792196987E-04
b_1= 8.6329030337575899E-04
b_2= 3.0215160618151565E-03
b_3= 6.0430321236303130E-03
b_4= 7.5537901545378912E-03
b_5= 6.0430321236303130E-03
b_6= 3.0215160618151565E-03
b_7= 8.6329030337575899E-04
b_8= 1.0791128792196987E-04

et les coefficients du dénominateur sont:

a_0= 1.000000000000000
a_1= -3.983784327663337
a_2= 7.536234286077626
a_3= -8.599815333907508
a_4= 6.400154303626428
a_5= -3.156025399574342
a_6= 1.001696629290593
a_7= -0.1863424879766865
a_8= 1.5507616199615062E-02

7.2.1 Simulation avec precision infinie

Le filtrage des signaux S1 et S2 s'effectue suivant la relation d'entrée-sortie suivante:

$$y(k)= \sum_{i=0}^{8} b_i\, u(k-i) - \sum_{i=1}^{8} a_i\, y(k-i) \qquad (7-3)$$

La représentation binaire des coefficients et celle des résultats des opérations (multiplications et additions) est faite sur des mots de 54 bits (précision de l'ordinateur) pour lesquels la précision est supposée infinie.

En effet, on constate que le filtrage de S1 est correctement effectué (voir Fig(7-1-d)); la fréquence f_2 qui se trouve à l'extérieur de la bande passante est bien rejetée, seule la fréquence f_1 est retenue à la sortie.

Le filtrage du signal bruit blanc S2 èlimine les composantes de hautes fréquences c'est à dire celles qui sont superieures à f_c. La sortie du filtre pour ce signal est donnée par Fig(7-2-b).

7.2.2 Simulation avec precision finie

Le filtrage est realisé à partir de la même relation (7-3), mais cette fois-ci en limitant la longueur des mots binaires representant le signal de l'entrée, les coefficients du filtre et les registres de stockage et d'accumulation. Cette quantification est realisée par la fonction **BBIT** dont le programme en Fortran est donné par Fig(7-3). Cette fonction effectue la quantification par l'arrondi avec saturation . La mobilité de la virgule dans cette quantification est destinée à élargir la dynamique afin de ne considèrer que les erreurs de calcul, (la probabilité de dépassement étant alors faible) . Pour simuler le quantificateur réel en virgule fixe, on propose une autre fonction Fortran , **BIT**, donnée par figure Fig(7-3 bis):

On effectue le filtrage pour des longueurs de mots fixées à B=4, 8, 12 et 16 bits. La sortie obtenue pour les quatres cas, est présentée par Fig(7-4).

On constate que le filtrage devient érroné à partir de 12 bits; la sinusoide de la sortie est distordue et cela revient au fait que les coefficients deviennent de moins en moins représentables. Pour 4 bits le signal en sortie est nul car tous les coefficients du numérateur de la fonction de transfert sont représentés par la valeur zero.

7.2.3 Variance de l'erreur de sortie

Pour chaque valeur de B, est calculée l'erreur de la sortie par l'expression :

$$\varepsilon(k) = Y_i(k) - Y_f(k) \qquad (7-4)$$

où $Y_i(k)$: la sortie du filtre avec précision infinie,

$Y_f(k)$: la sortie du filtre avec précision finie.

On calcule la variance de l'erreur de sortie simulée (qu'on

La fonction "BBIT" simule un quantificateur par l'arrondi avec une caractéristique de dépassement par saturation où:

IB= la longueur du mot desirée (bit de signe exclu).

X = la valeur à quantifier (en double précision).

XX= X quantifié sur IB+1 bits.

```
      REAL*8 function BBIT(X,IB)
      REAL*8 A,y,xx,k,k1,N,J,X
      y= Dabs(x)
      A= INT(y)
      K= 2** IB
      J= K - 1
      IF ( A .GE. J )  THEN
          XX= J
          GOTO 20
      ENDIF
      IF ( A .EQ. 0.D0 )  THEN
          xx= NINT( y * K ) / K
          GOTO 20
      ENDIF
      I= -1
10    K= K / 2
      I= I + 1
      N= INT(A / K)
      IF ( N .EQ. 0 )  GOTO  10
      K1= 2 * I
      XX = NINT( (y-A)* K1 ) / K1 + A
20    IF (x.LT.0.d0)  xx = -xx
      bbit= xx
      RETURN
      END
```

FIG-(7- 3) FONCTION DE QUANTIFICATION " BBIT "

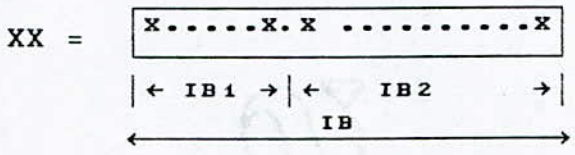
La fonction BIT réalise la quantification en virgule fixe par l'arrondi en signe et valeur absolue et une caractéristique de dépassememt par saturation ; où :

x est la valeur à quantifier (précision infinie) ,

IB est la taille du mot en bits ,

IB2 est la longueur en bits de la partie fractionnaire du mot.

XX est la valeur quantifiée de X qui sera sous forme:




```
      REAL*8 FUNCTION BIT(X,IB,IB2,XX)
      REAL*8 y,xx,k1,X,AX,S
      INTEGER IB,IB1,IB2
      IB2= IB - IB1
      AX = 0.d0
      DO 5 I=1,IB2
         AX= AX + 2.d0**(FLOAT(-I))
5     CONTINUE
      Y=DABS(X)
      K1= 2.d0**FLOAT(IB2)
      S= (2.d0**FLOAT(IB1))- 1.d0+ AX
      IF( Y. GE. S) THEN
       XX = S
       Write(*,*)' OVERFLOW '
       GOTO 20
      ENDIF
      XX= NINT(Y * k1) / k1
20    IF ( X .LT. 0.d0)  XX = - XX
      BIT = XX
      RETURN
      END
```

Fig(7-3 bis) Fonction de quantification en virgule fixe

appellera pratique) par:

$$\sigma^2_{y\,pratique} = \frac{1}{Ne}\sum_{k=0}^{Ne-1}\varepsilon(k)^2 - \left[\frac{1}{Ne}\sum_{k=0}^{Ne-1}\varepsilon(k)\right]^2 \qquad (7-5)$$

où Ne est le nombre d'échantillons du signal de sortie.
pour deux cas:

- Cas de l'accumulateur simple, pour lequel chaque produit intermédiaire est quantifié avant la sommation dans (7-3).

L'expression théorique de la variance des erreurs de calcul en sortie est:

$$\sigma^2_{y\,théorique} = \frac{q^2}{12}(N+1)\delta^2\,(G+1) \qquad (7-6)$$

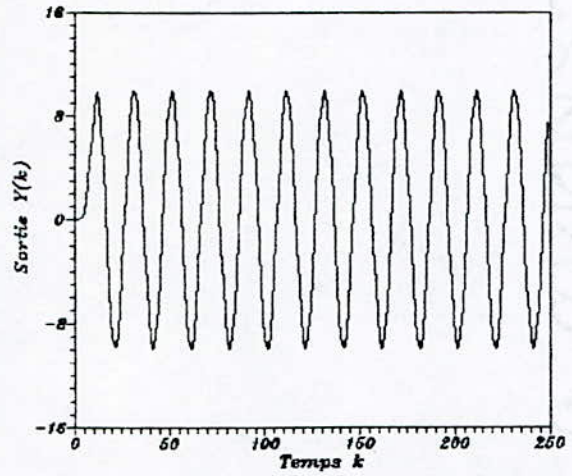
N est l'ordre du filtre ,dans ce cas N =8
q est le pas de quantification, dans ce cas $q=10\ 2^{(-B+1)}$
δ est le paramétre de normalisation, ici $\delta=1$,
G est le gain de bruit de calcul de la structure canonique, il a étè calculé par un programme en Fortran (établi par B.DERRAS à l'ENP en 1991), et pour le filtre utilisé, sa valeur est:

$$G_{canonique} = 8019.683673775$$

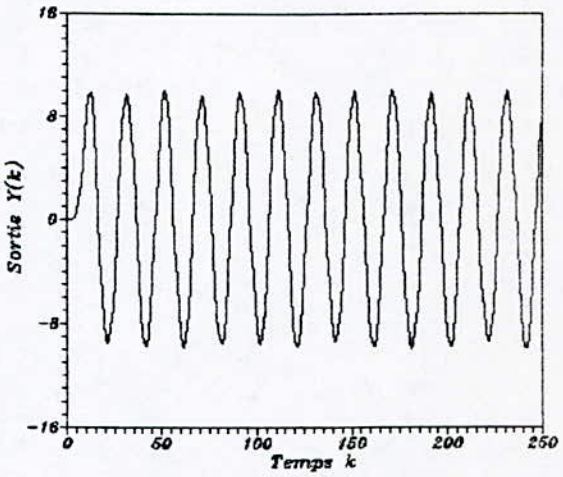
- Cas de l'accumulateur double, pour lequel la quantification s'effectue aprés la sommation de tous les produits de (7-3). L'expression théorique de la variance, dans ce cas, est:

$$\sigma^2_{y\,théoque\ double} = \frac{\sigma^2_{y\ théorique\ simple}}{(N+1)} \qquad (7-7)$$

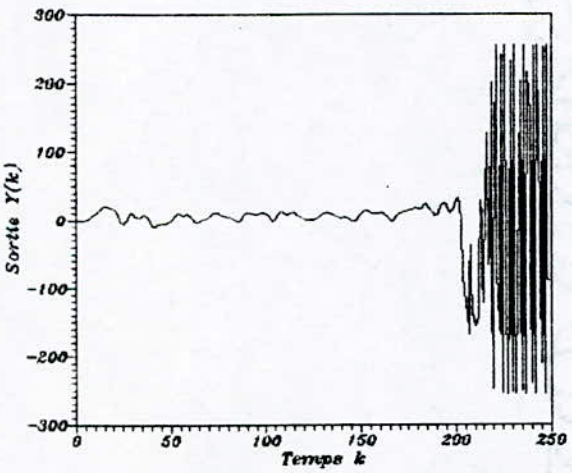
La variance de l'erreur est tracée en fonction de B pour les signaux S1 et S2 (voir Fi (7-5) et Fig(7-6)). On constate que, pour ces deux tracés et dans tous les cas, la variance est décroissante et correspond bien à aux expressions théoriques données par (7-6) et (7-7). On peut aussi remarquer que la variance pour le cas d'un accumulateur double est infèrieure à celle pour un accumulateur simple. La variance relative au bruit blanc S2 se rapproche le plus de la variance théorique que celle relative au signal S1.



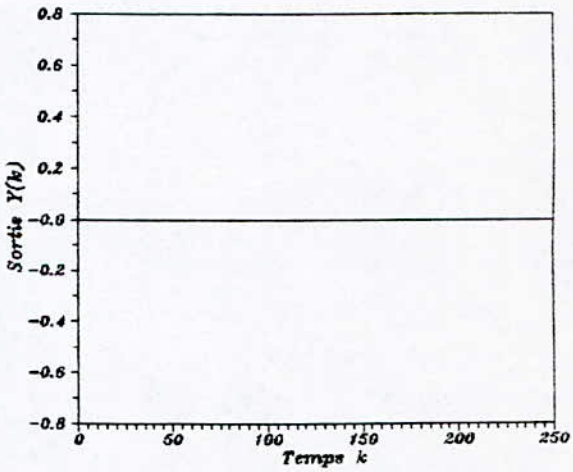
(a) Pour Accumulateur 16 BITS



(b) Pour Accumulateur 12 BITS

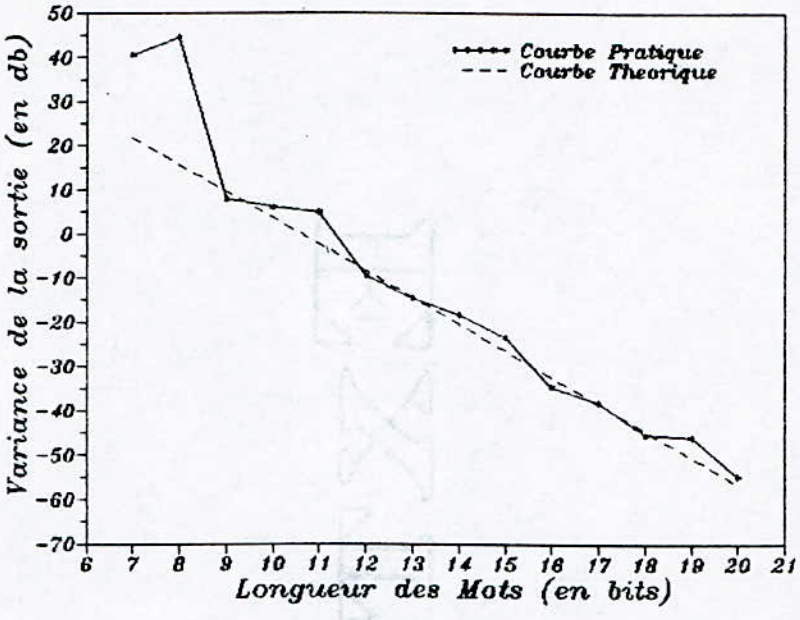


(c) Pour Accumulateur 8 BITS

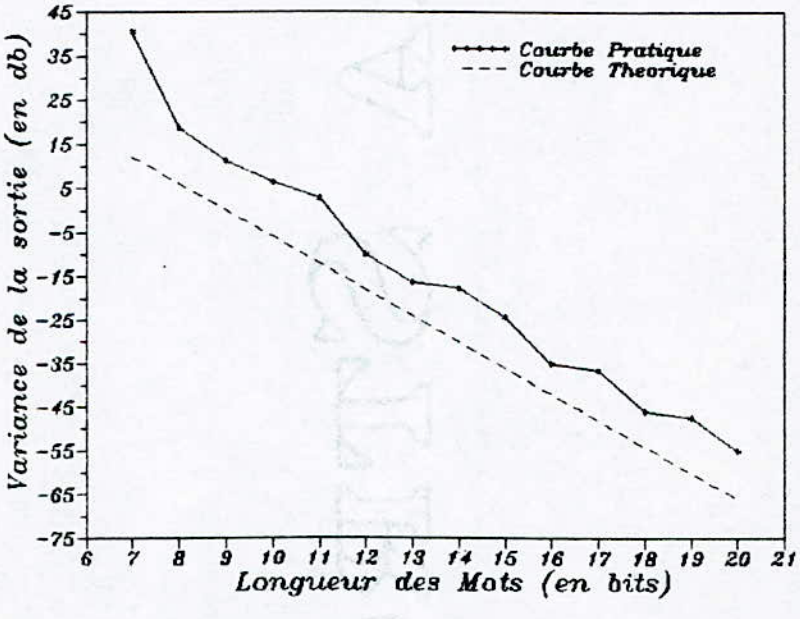


(d) Pour Accumulateur 4 BITS

FIG-(7-4) SORTIE DE LA STRUCTURE CANONIQUE
POUR DES DIFFERENTES LONGUEURS D'ACCUMULATEURS

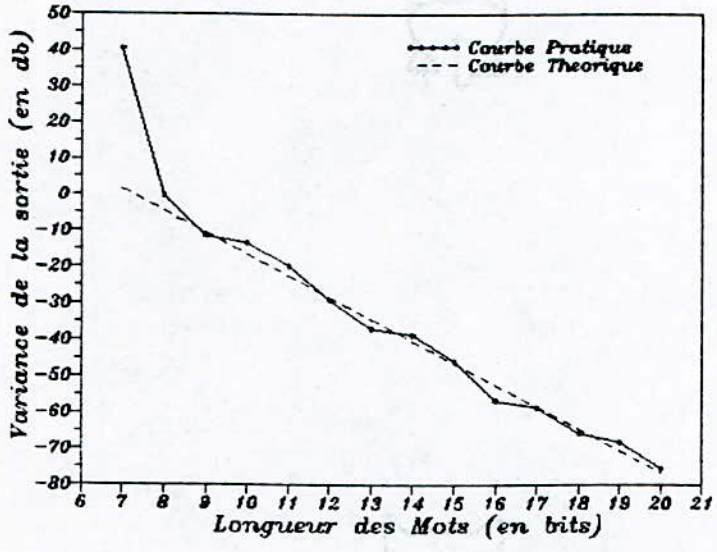


(a) Accumulateur Simple

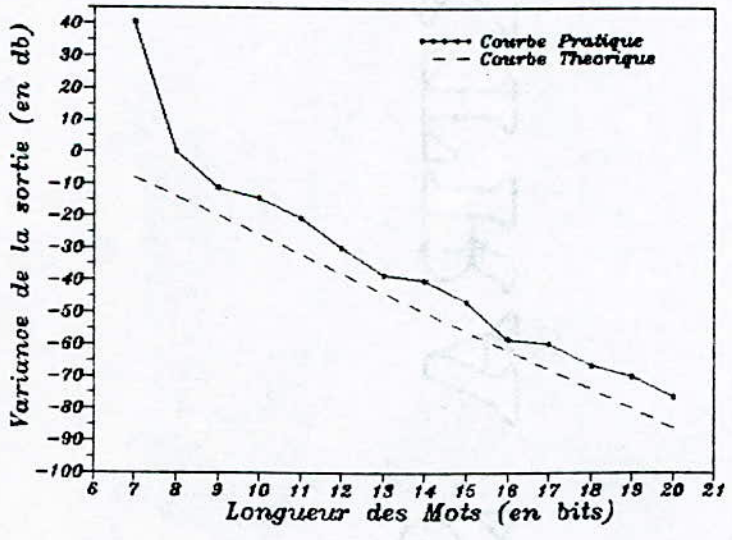


(b) Accumulateur Double

FIG-(7-5) ENTREE SOMME DE DEUX SINUSOIDES
(f1= 0.05 et f2= 0.3)



(a) Accumulateur Simple



(b) Accumulateur Double

FIG-(7-6) ENTREE BRUIT BLANC CENTREE
DE VARIANCE UNITE

appellera pratique) par:

$$\sigma^2_{y\,pratique} = \frac{1}{Ne}\sum_{k=0}^{Ne-1}\varepsilon(k)^2 - \left[\frac{1}{Ne}\sum_{k=0}^{Ne-1}\varepsilon(k)\right]^2 \quad (7-5)$$

où Ne est le nombre d'échantillons du signal de sortie.

pour deux cas:

- Cas de l'accumulateur simple, pour lequel chaque produit intermédiaire est quantifié avant la sommation dans (7-3).

L'expression théorique de la variance des erreurs de calcul en sortie est:

$$\sigma^2_{y\,théorique} = \frac{q^2}{12}(N+1)\delta^2(G+1) \quad (7-6)$$

N est l'ordre du filtre ,dans ce cas N =8

q est le pas de quantification, dans ce cas $q=10\ 2^{(-B+1)}$

δ est le paramétre de normalisation, ici $\delta=1$,

G est le gain de bruit de calcul de la structure canonique, il a étè calculé par un programme en Fortran (établi par B.DERRAS à l'ENP en 1991), et pour le filtre utilisé, sa valeur est:

$$G_{canonique} = 8019.683673775$$

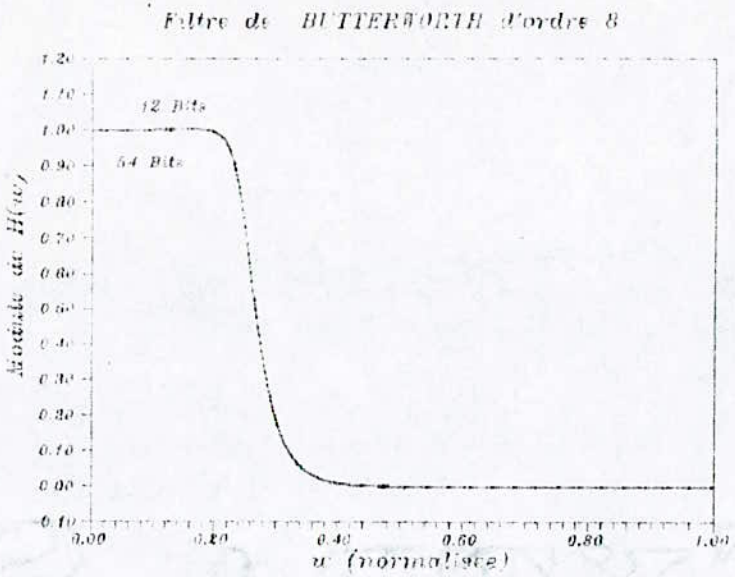
- Cas de l'accumulateur double, pour lequel la quantification s'effectue aprés la sommation de tous les produits de (7-3). L'expression théorique de la variance, dans ce cas, est:

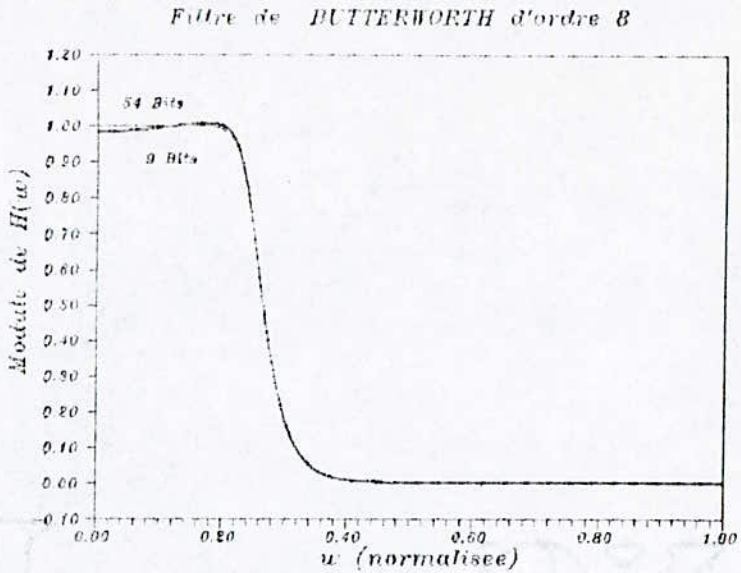
$$\sigma^2_{y\,théoque\ double} = \frac{\sigma^2_{y\ théorique\ simple}}{(N+1)} \quad (7-7)$$

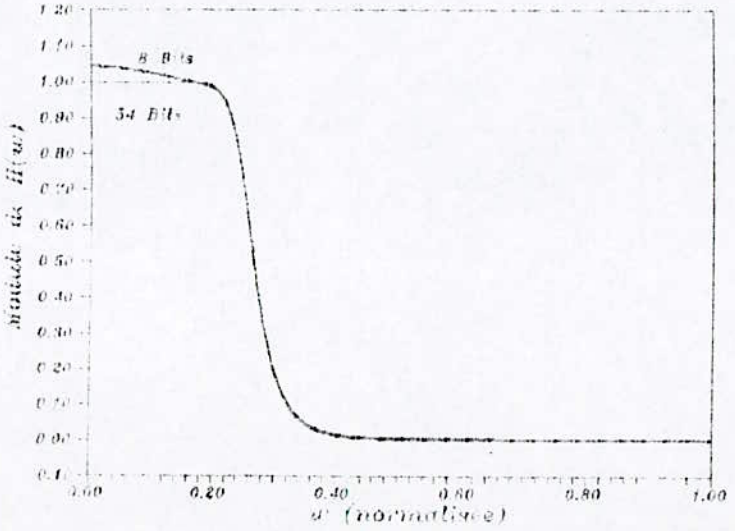
La variance de l'erreur est tracée en fonction de B pour les signaux S1 et S2 (voir Fi (7-5) et Fig(7-6)). On constate que, pour ces deux tracés et dans tous les cas, la variance est décroissante et correspond bien à aux expressions théoriques données par (7-6) et (7-7). On peut aussi remarquer que la variance pour le cas d'un accumulateur double est infèrieure à celle pour un accumulateur simple. La variance relative au bruit blanc S2 se rapproche le plus de la variance théorique que celle relative au signal S1.

7.2.4 Sensibilté de la rèponse fréquentielle

L'effet de la variation des coefficients de la structure canonique dûe à la limitation de la longueur des registres sur la fonction de transfert du filtre est présenté par Fig (7-7). Cette figure montre que, pour 12 bits, une distorsion de la


Filtre de BUTTERWORTH d'ordre 8
12 Bits
54 Bits
Module de H(w)
w (normalisee)


Filtre de BUTTERWORTH d'ordre 8
54 Bits
9 Bits
Module de H(w)
w (normalisee)


Filtre de BUTTERWORTH d'ordre 8
8 Bits
54 Bits
Module de H(w)
w (normalisee)

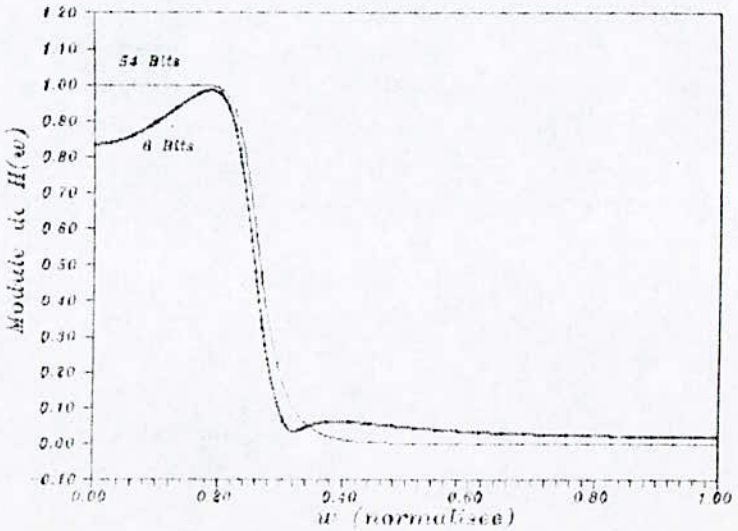

Filtre de BUTTERWORTH d'ordre 8
54 Bits
8 Bits
Module de H(w)
w (normalisee)

Fig (7-12)

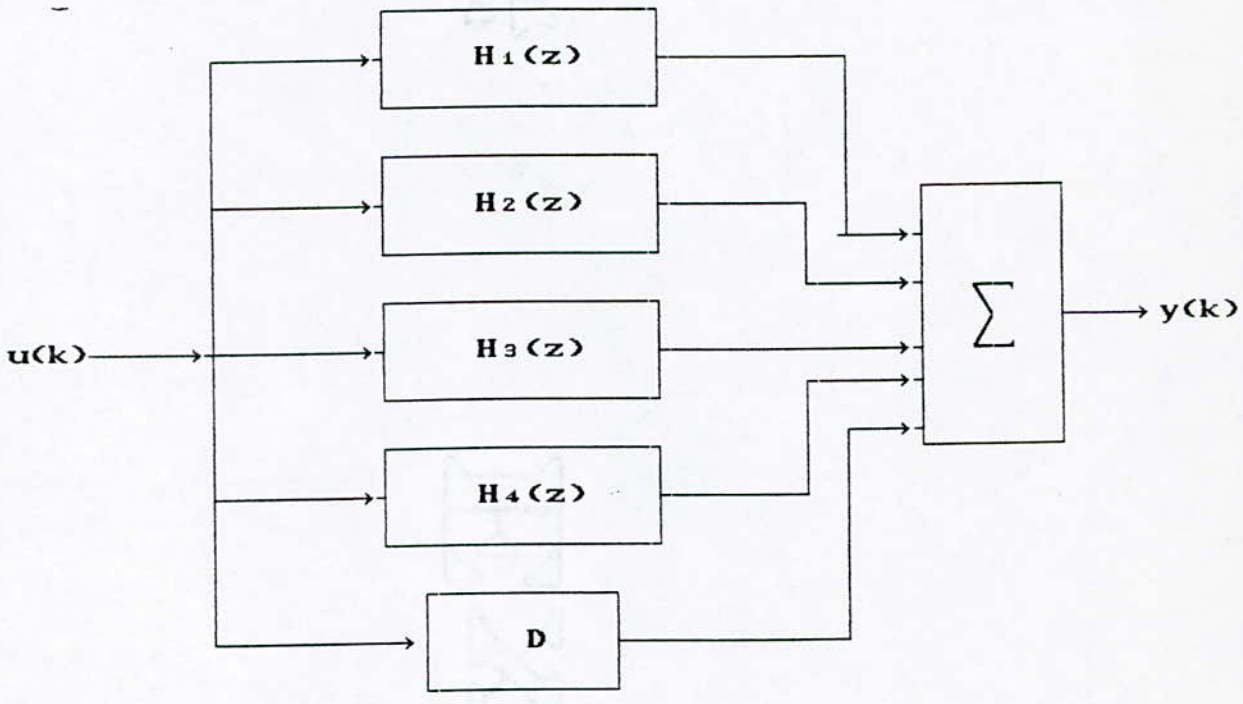
fonction de transfert apparait. Elle devient plus importante au fur et à mesure que la longueur des registres contenant les coefficients devient petite.

Pour 6 bits, la fonction de transfert est nulle car les coéfficients ne sont plus représentables dans cette gamme. En conséquence, la qualité du filtrage se dètériore graduellement pour des registres moins de 12 bits. La sensibilité de la fonction de transfert se traduit par l' ampleur de cette dègradation.

7.3 Performances d'un filtre numérique avec une structure décomposée en parallèle

Dans cette partie, il s'agit de mettre en œuvre la théorie explicitée dans le chapitre prècèdent par une simulation sur ordinateur. A cet effet, la décomposition du filtre en sections du second ordre regroupant chacune un couple de pôles complexes conjugués et disposées en parallèle, ainsi que la construction des structures canoniques et optimales suivant les trois méthodes exposées dans le chapitre V (BOMAR, BARNES et HWANG), sont éxècutées par le programme STRUCMIN se trouvant en annexe.

Le filtre d'ordre 8 simulé est ainsi decomposé en 4 cellules d'ordre 2 que l'on connectera en parallèle (Fig (7-8)).



Fig(7-8) Dècomposition du filtre d'ordre 8 en cellules parallèles

Chaque cellule est de la forme :

$$H(z)= \frac{q_1 z + q_2}{1 + p_1 z + p_2 z^2} \qquad (7-8)$$

où q_1, q_2, p_1 et p_2 sont des constantes réelles spécifiques pour chaque cellule. Les pôles de H(z) sont de la forme:

$$\lambda = \alpha \pm j\beta$$

Les resultats de la décomposition sont donnés par le tableau suivant :

	$H_1(z)$	$H_2(z)$	$H_3(z)$	$H_4(z)$
q1	-3.2868476268	-0.6181903311	0.4356772381	3.4706539054
q2	1.9459927643	-0.4018617159	-0.3054473169	-1.1160909524
p1	-0.8905976013	-1.0153398695	-1.2427733740	-0.8350734828
p2	0.2594951881	0.4359073931	0.7575469450	0.1809722346
α	0.4452987909	0.5076699257	0.6213867068	0.4175367355
β	0.2473947555	0.4221121073	0.6094468832	0.0814573765

Le coefficient du chemin direct entrée/ sortie est:

D= 1.0791128792196987E-04

On construit les structures d'etats directes (canoniques) des 4 cellules et en calculant le gain de bruit pour chacune,on obtient:

Cellule	1	2	3	4
Matrice A	0.000 1.000 -0.259 0.891	0.000 1.000 -0.436 1.015	0.000 1.000 -0.757 1.243	0.000 1.000 -0.181 0.835
Vecteur B	0.000 1.000	0.000 1.000	0.000 1.000	0.000 1.000
Vecteur C	1.946 -3.287	-0.402 -0.618	-0.305 0.436	-1.116 3.471
Gain de bruit	35.334463	6.892163	3.728628	37.074119

Le gain total pour la forme décomposée directe est donc :

$$G_{\text{déc. can}} = 83.0293732064$$

On calcule ensuite les structures d'etat optimales selon une des trois procédures étudiée dans le chapitre VI et on procéde de la même maniére que pour les structures canoniques. En utilisant la procédure de **BOMAR**, par exemple , on obtient les resultats suivant:

Cellule	1	2	3	4
Matrice A	0.445 -0.486 0.126 0.445	0.508 -0.279 0.639 0.508	0.621 -0.700 0.530 0.621	0.417 -0.035 0.191 0.417
Vecteur B	0.938 0.839	0.886 0.237	0.495 0.491	0.923 0.797
Vecteur C	-1.752 -1.958	-0.349 -1.304	0.440 0.444	1.879 2.178
Gain de bruit	8.7138762	4.327928	1.610885	11.126727

Le gain total de la structure décomposée optimale est :

$$G_{\text{déc. optimale}} = 25.77941741187027$$

De même , il est intéressant de voir le gain de bruit de la structure optimale globale du filtre à partir de ses modes du second ordre et dont la valeur est [19] :

$$G_{optimal} = 0.8597278167$$

D'aprés ces resultats, on déduit les hiérarchies qui existent entre les gains des différentes structures :

$$G_{opt.\,global} < G_{déc.\,optimal} < G_{déc.\,can} < G_{can.\,global}$$

et entre les nombres d'opérations NO par échantillon de la sortie pour chaque structure :

$$NO_{opt.\,global} > NO_{déc.\,optimal} > NO_{déc.\,can} = NO_{can.\,global}$$

où $NO_{opt.\,global} = (N+1)^2$, $NO_{déc.\,optimal} = 4N+1$ et $NO_{déc.\,can} = 2N+1$

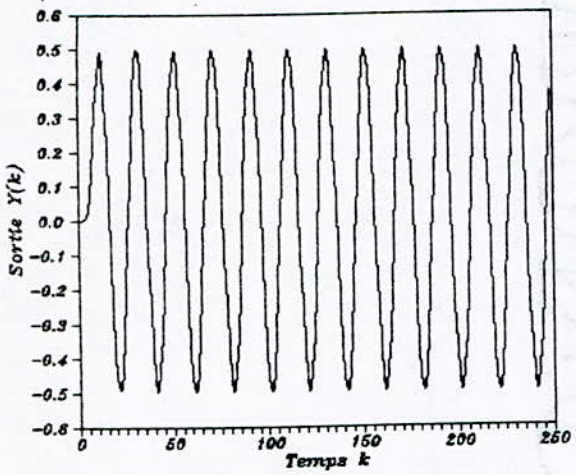
On peut conclure que les structures décomposées canonique et optimale présentent un certain équilibre par rapport aux structures optimale et canonique globales du point de vue des erreurs de calcul et nombres d'opérations. Le choix de la structure convenable dépend du cahier de charge exigé pour une application donnée, et plus précisément du materiel disponible (multiplieurs), du coût de la réalisation et du rapport signal sur bruit desié (qualité du filtrage).

7.3.1 Simulation avec précion finie

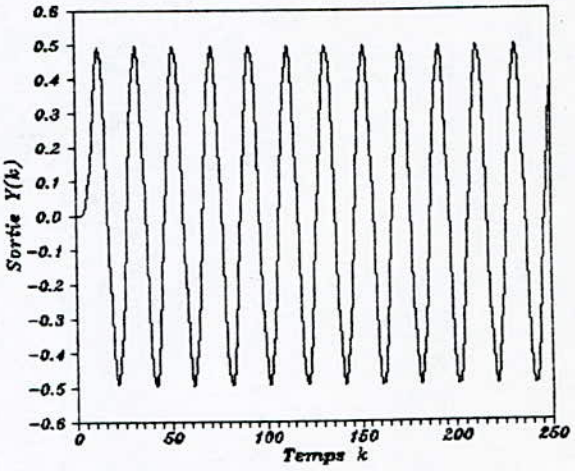
Dans cette partie ,on refait la même opération de filtrage sur le signal S1 qu'on normalise en le mettant dans la gamme [-1,1]. on fait varier la longueur des mots binaires mais cette fois avec une structure décomposée optimale; On obtient la Fig(7-9) sur laquelle on remarque que la sinusoide de la sortie est conservée jusqu'à environ 8 bits, et que pour 6 bits le signal n'est plus nul bien qu'il est distordu relativement au cas canonique (voir Fig(7-4)).

a- variance de l'erreur de sortie

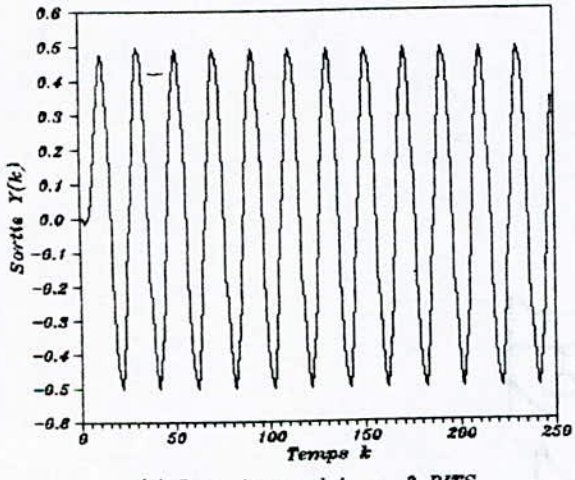
De la même maniére que pour le cas de la structure canonique, on calcule la variance de l'erreur de sortie en fonction du nombre de bits des registres simulés , on obtient pour le signal S1,la figure Fig(7-11) et pour le signal S2 la figure Fig (7-10).



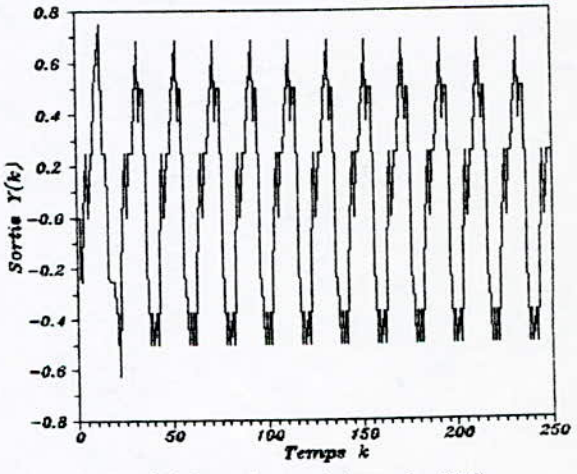
(a) Pour Accumulateur 16 BITS



(b) Pour Accumulateur 12 BITS



(c) Pour Accumulateur 8 BITS



(d) Pour Accumulateur 4 BITS

FIG-(7- 9) SORTIE DE LA STRUCTURE DECOMPOSEE OPTIMALE OPTIMALE POUR DES DIFFERENTES LONGUEURS D'ACCUMULATEURS

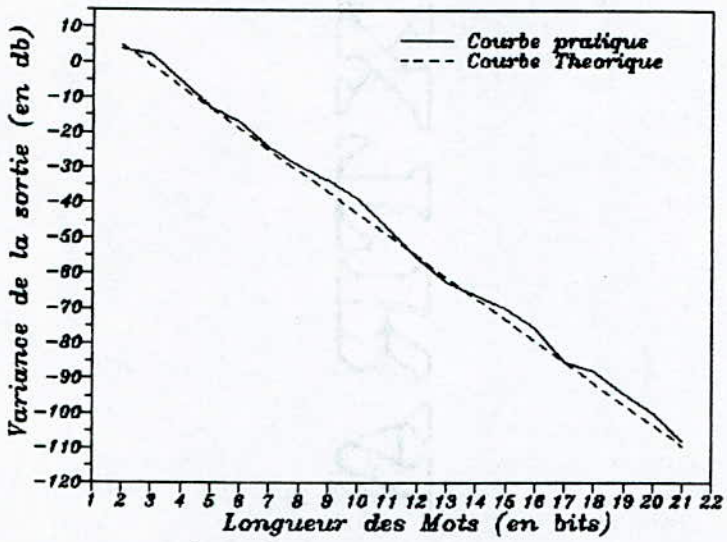


FIG-(7-10 a) ENTREE BRUIT BLANC
Accumulateur SIMPLE

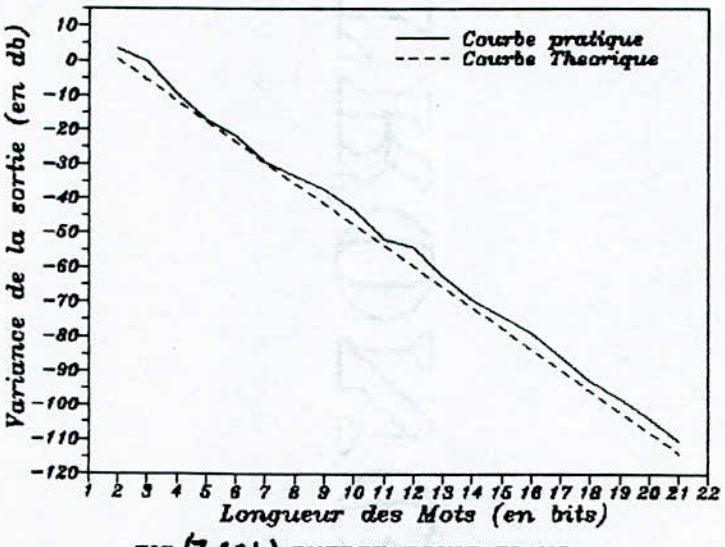


FIG-(7-10 b) ENTREE BRUIT BLANC
Accumulateur de longueur DOUBLE

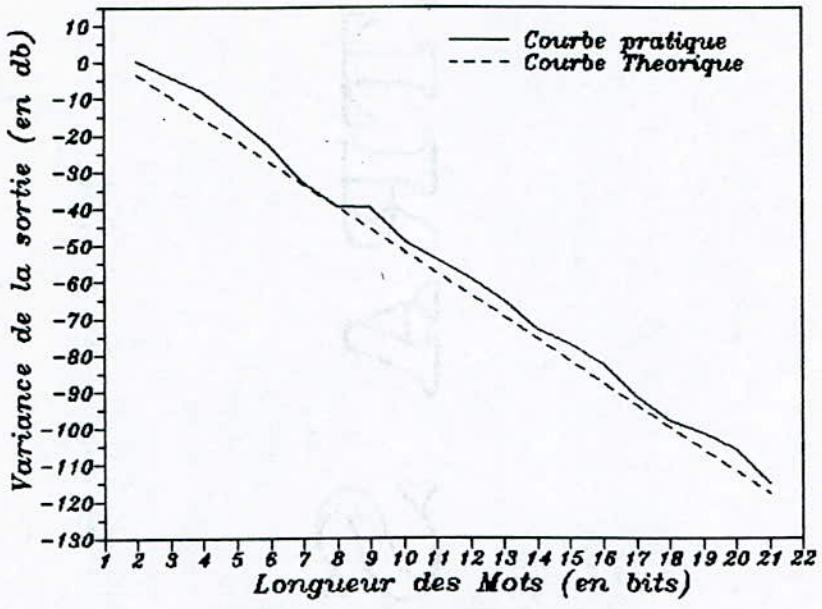


FIG-(7-11 a) *ENTREE SINUSOIDALE*
Accumulateur SIMPLE

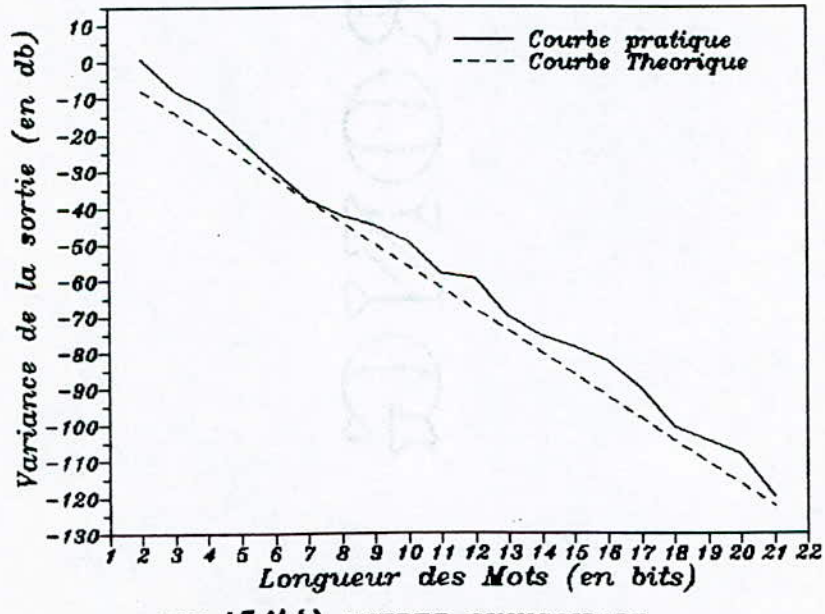


FIG-(7-11 b) *ENTREE SINUSOIDALE*
Accumulateur de longueur DOUBLE

On remarque pour chaque signal le rapprochement entre la variance théorique donnée par :

$$\sigma^2_{y\,\text{théorique}} = \frac{q^2}{12}\,(3\,\delta^2 G_{\text{déc. opt}} + 1) \qquad (7\text{-}8)$$

pour le cas d'un accumulateur simple et par :

$$\sigma^2_{y\,\text{théorique}} = \frac{q^2}{12}\,(\delta^2 G_{\text{déc. opt}} + 1) \qquad (7\text{-}10)$$

pour le cas d'un accumulateur double; et entre la variance pratique . Ce qui confirme l'applicabilité de la théorie de minimisation par les structures d'etat de MULLIS et de HWANG[16],[17]. Pour le cas de l'accumulateur double, on remarque que la variance est moindre que celle du cas de l'accumulateur simple.

b- Sensibilité de la réponse fréquentielle

En utilisant les paramétres de la structure décomposée optimale, on calcule pour chaque cellule la réponse fréquentielle par:

$$H_i\,(e^{j\omega}) = C_i\,(zI - A_i)^{-1} B_i \qquad (7\text{-}11)$$

où A_i ,B_i, C_i sont les paramétres d'etat de la cellule i d'ordre 2.

Ensuite on calcule la réponse fréquentielle globale :

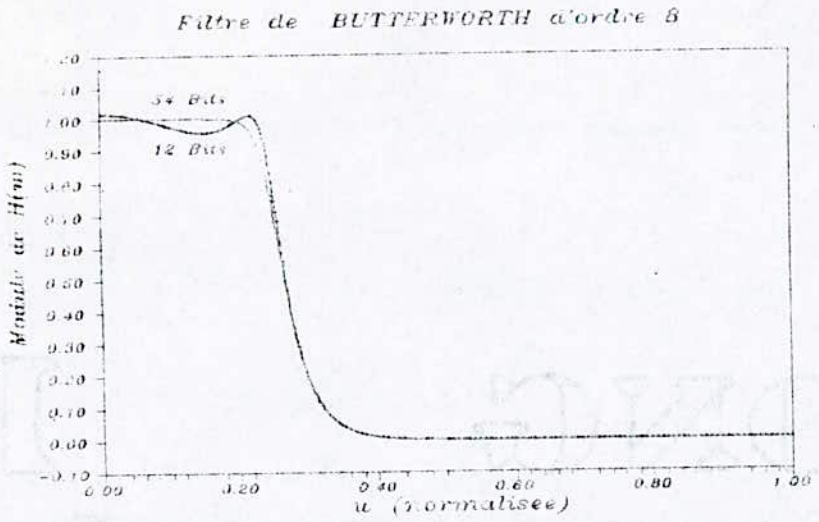
$$H\,(e^{j\omega}) = \sum_{i=1}^{4} H_i\,(e^{j\omega}) + D \qquad (7\text{-}12)$$

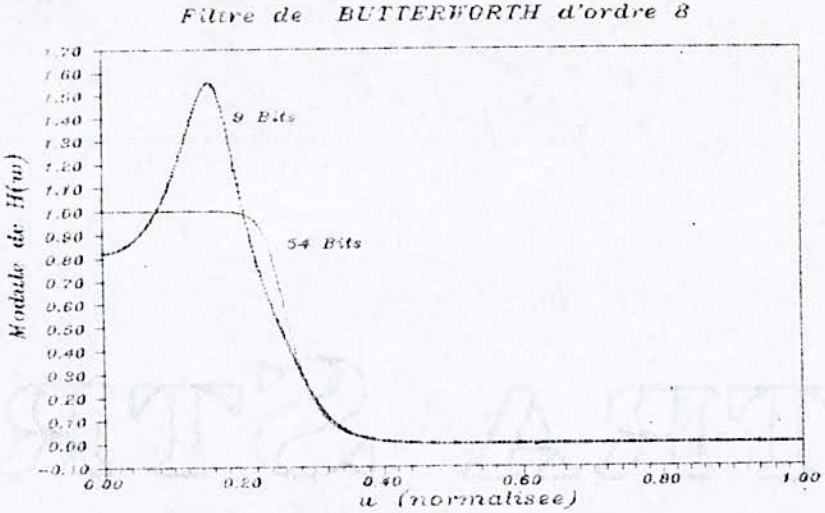
On obtient pour les différentes valeurs de longueur des registres, les courbes données par Fig(7-12).

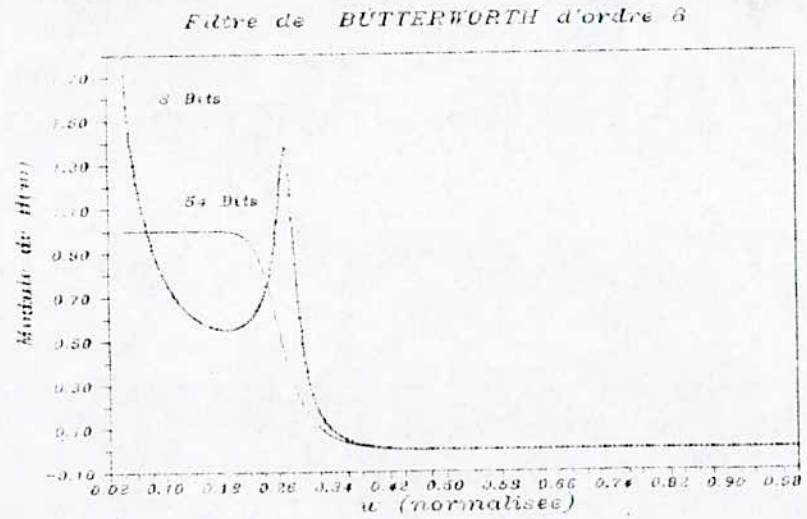
On constate une nette amélioration de la réponse fréquentielle pour cette structure en comparaison avec la structure canonique en précision finie Fig(7-7)) . En effet , la sensibilité de la fonction de transfert a diminué; cela revient au fait que la décomposition en sections d'ordre 2 a permis de séparer les pôles qui sont proches dans le cercle unité , avant la quantification.

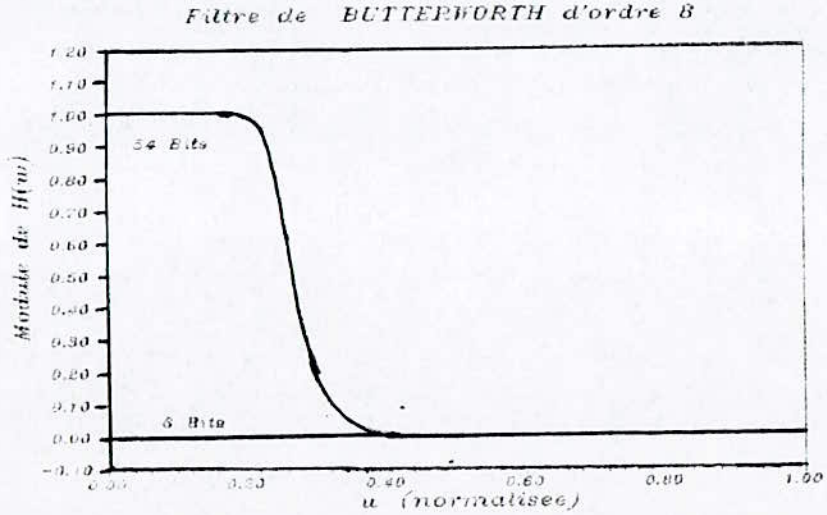
La distorsion de la courbe commence à apparaitre à partir de 8 bits. Cette réponse n'est plus nulle pour 6 bits comme il a étè vu pour la structure canonique globale; (comparer Fig(7-7)et Fig(7-12)).

Fig (7-7)


Filtre de BUTTERWORTH d'ordre 8
54 Bits
12 Bits
Module de H(w)
u (normalisee)


Filtre de BUTTERWORTH d'ordre 8
9 Bits
54 Bits
Module de H(w)
u (normalisee)


Filtre de BUTTERWORTH d'ordre 8
8 Bits
54 Bits
Module de H(w)
u (normalisee)


Filtre de BUTTERWORTH d'ordre 8
54 Bits
8 Bits
Module de H(w)
u (normalisee)

7.4 Effet d'une transformation fréquentielle sur le gain de bruit de calcul

Dans cette secion, On etudie le comportement des gains de bruit de calcul des quatres types de structures vis-à-vis - d'une transformation fréquentielle d'ordre 1, en synthétisant des filtres passe-bas de **BUTTERWORTH** d'ordre 8 de fréquences de coupure différentes. On obtient le tableau suivant:

Fréquence de coupure	Gain de Bruit de Calcul pour la Structure :			
	Canonique	Décomposée Canonique	Décomposée Optimale	Optimale
0.1	3.6697E+009	434.747	25.7794	0.859728
0.2	196390	120.161	25.7794	0.859728
0.3	590.243	63.4288	25.7794	0.859728
0.4	12.7886	45.8975	25.7794	0.859728
0.5	10.92	41.5147	25.7794	0.859728
0.6	157.627	45.8975	25.7794	0.859728
0.7	8697.39	63.4288	25.7794	0.859728
0.8	2927340	120.161	25.7794	0.859728
0.9	5.4965E+010	434.747	25.7794	0.859728

La figure Fig(7-13) illustre la variation du gain en fonction de la largeur de la bande : le gain de la structure canonique varie sensiblement avec la fréquence de coupure ,il est important pour les filtres à bandes étroites. Pour la structure décomposée canonique le gain est relativement moins sensible à la largeur de la bande. Le gain de bruit est insensible à la transformation fréquentielle d'ordre 1 pour les structures optimales et décomposées optimales.

Conclusion

La simulation d'un filtre RII a permis d'étudier les performances des quatre structures d'etat : canonique globale, décomposée canonique, décomposée optimale et optimale globales; et de vérifier la correspondance entre la théorie et la pratique .

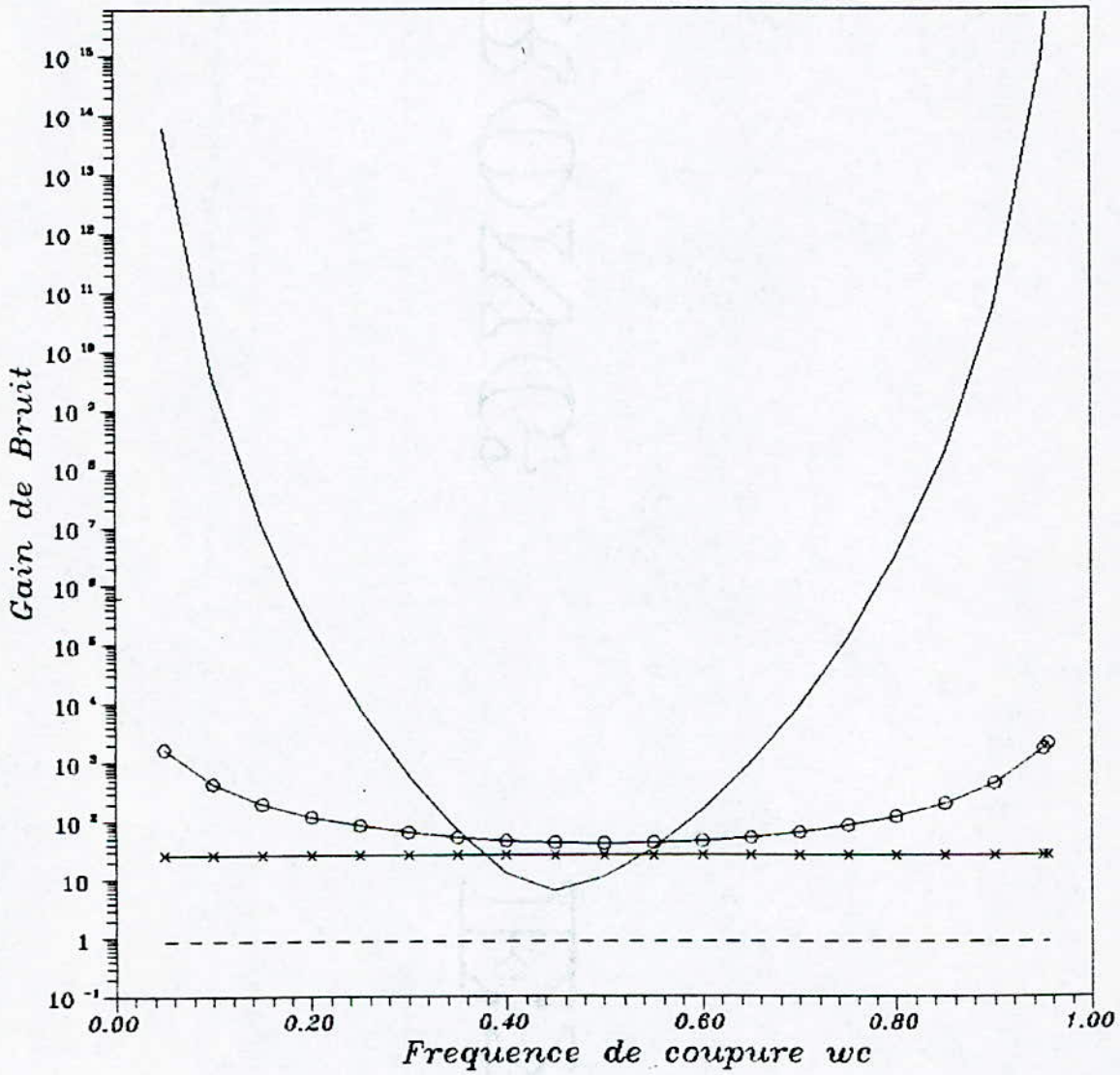


FIG-(7-13) EFFET DE LA TRANSFORMATION FREQUENTIELLE PB-PB SUR LE BRUIT DE CALCUL POUR DIFFERENTES STRUCTURES

——— *Structure Canonique*
ooooo *Structure Decomposee Canonique*
xxxxx *Structure Decomposee Optimale*
- - - - *Structure Optimale*

CHAPITRE VIII

CONCLUSION GENERALE

CHAPITRE VIII

CONCLUSION GENERALE

Aprés avoir exposé brièvement les différentes méthodes de synthèse des filtres numériques ainsi que les distorsions et les effets non linéaires introduits par la quantification des coefficients et des résultats des opérations arithmétiques du filtre numérique RII en virgule fixe, la caractérisation classique de tels filtres par leurs fonctions de transfert ou leurs réponses impulsionnelles s'est révélée insuffisante pour décrire et analyser leurs comportements surtout dans le cas de l'utilisation de mots de longueur finie.

La représentation d'état du filtre est utilisé pour apporter plus de détails et de précision sur le déroulement des opérations et sur la façon avec laquelle s'effectue le filtrage.

Grâce à la propriété de changement de coordonnées dans l'espace d'état par une simple transformation non singulière, des structures éliminant les dépassements, des structures normalisées et des structures à gain de bruit minimale sont possibles pour un même filtre numérique.

Pour trouver les structures à gain de bruit minimal, deux méthodes d'approche ont étè utilisées (**MULLIS-ROBERTS** [16] et **HWANG** [17]). L'inconvenient de ces structures est leur complexité du point de vue du nombre d'opérations arithmétiques nécessaires dont, en particulier, celui des multiplications qui est de $(N+1)^2$ pour un filtre d'ordre N et ce nombre n'est que (2N+1) pour les structures canoniques.

La théorie de la décomposition du filtre en cellules de second ordre [1],[28] connectées en parallèle a permis d'étudier deux structures de compromis (canonique et optimale) entre le gain de calcul et la complexité des réalisations. Pour la structure décomposée optimale, trois procédures d'optimisation du gain de bruit (**BOMAR, BARNES** et **HWANG** [1],[21],[27]) pour les filtres numériques RII du second ordre ont étè étudiées et simulées, elles permettent d'obtenir des cellules du second ordre à gain de bruitminimal, ce qui réduit le nombre des multiplications pour le filtre globale à (4N+1). Toutefois, le gain de bruit

cette structure est superieur à celui de la structure optimale globale.

Le nombre des multiplications pour les structures décomposées canoniques est le même que celui des structures canonique globale, c'est à dire (2N+1), mais leurs gains en bruit sont beaucoup plus faible; ce qui rend ce type de structures intéressant.

La simulation d'un filtre numérique RII du type passe-bas (CHAPITRE VII) a permis de vérifier les résultats élaborés en théorie ainsi que la validité des hypothèses faites sur les erreurs de calcul. Une comparaison entre les deux structures canonique et décomposée optimale a étè établie en termes de variance de l'erreur de sortie, de la sensibilité de la fonction de transfert à la quantification de ses coefficients pour chaque structure et de la qualité du filtrage.

En conclusion, la réalisation d'un filtre numérique RII en virgule fixe n'est pas unique; elle dépend de la structure d'état qui doit être choisie de façon à répondre aux spécifications techniques (tolérances en bruit de calcul) et économiques (nombre de multiplieurs nécessaires).

BIBLIOGRAPHIE

[1] C.T. MULLIS & R.A. ROBERTS, Digital Signal Processing, Addison Wesley, Reading, MA, 1987.

[2] M.KUNT, Traitement Numérique des Signaux, Dunod 1981.

[3] J.LIFERMANN, Les méthodes Rapides de Transformation du signal: Fourier, Walsh, Hadamard; Masson 1980.

[4] J.MAX, Méthodes et Techniques du Traitement du signal, Tome 1, Masson 1987.

[5] D.ADDOU, " Le filtrage Numérique: Application au Traitement de la Parole, " Projet de Fin D'etudes, ENPA 1987.

[6] M.LABARRERE et al., Le Filtrage et ses applications, CEPAD 1988.

[7] M.S.MOAD, "Synthése des Filtres numériques RII par la méthode des moindres carrés modifiés," Projet de fin d'etudes, ENPA 1990.

[8] M.BELLANGER, Traitement Numérique du signal, Masson 1987.

[9] T.M.CLAASEN et al., "Effects of quantization and overflow in recursive digital filters, "IEEE trans. Acous, speech & signal processing, Vol.ASSP-26, mars 1979.

[10] L.R.RABINER & B.GOLD, Theory and Application of Digital Signal Procesing, Prentice Hall, Englewood Cliffs, NJ, 1975.

[11] A.ANTONIOU, Digital Filters:Analysis and Design, Mac Graw Hill, NY, 1979.

[12] L.B.JACKSON, "Limit Cycles in State-Space Structures for Digital Filters," IEEE Trans. Circuits & Systems, Vol. CAS-26, Jan 1979.

[13] P.S.DINIZ & A.ANTONIOU, " More Economical State-Space Digital-Filter Structures Which are Free of Constant-Input Limit Cycles," IEEE Trans. Acous, Speech & Signal Processing, Vol. ASSP-34, Aout 1986.

[14] W.L.MILLS et al.,"Digital Filter Realisations without Overflow Oscillations," IEEE Trans. Acous,Speech & Signal Processing,Vol.ASSP-26,Aout 1978.

[15] C.W.BARNES,"Roundoff Noise and Overflow in Normal Digital Filters," IEEE Trans. Circuits & Systems, Vol.CAS-23, Mars 1979.

[16] C.T.MULLIS & R.A.ROBERTS,"Synthesis of Minimum Roundoff Noise Fixed Point Digital Filters," IEEE Trans. Circuits & Systems, Sept 1976.

[17] S.Y.HWANG, " Minimum Uncorrelated Unit Noise in State-Space Digital Filtering," IEEE Trans. Acous, Speech & Signal processing, Vol.ASSP-25, Aout 1977.

[18] C.T. MULLIS & R.A. ROBERTS," Roundoff Noise in Digital Filters : Frequency Transformations and Invariants," IEEE Trans. Acous., Speech & Signal Processing, Vol.ASSP-24 , Dec 1976.

[19] B. DERRAS, " New Efficient State-Space Structures for Realisation of Recursive Digital Filters," Thèse de Master, Université du Colorado, 1985.

[20] B.W.BOMAR, "Minimum Roundoff Noise Digital Filters with Some Power of Two Coefficients," IEEE Trans. Acous., Speech & Signal Processing, Vol.ASSP-34, Oct 1984.

[21] B.W.BOMAR, " New Second Order State-Space Structures for Realising Low Roundoff Noise Digital Filters," IEEE Trans. Acous., Speech & Signal Processing,Vol.ASSp-33,fev 1985.

[22] C.W.BARNES, "computationally Efficient Second Order Digital Filter Sections with Low Roundoff Noise Gain," IEEE Trans. Circuits & Systems,Vol. CAS-31, Oct 1984.

[23] M. ARJMAND & R.A.ROBERTS, "Reduced Multiplier,Low Roundoff Noise Digital Filters," Department of Electrical Engineering,University of Colorado, IEEE 1979.

[24] C.W.BARNES, "A parametric Approach to the realisation of Second Order Digital Filter Sections," IEEE Trans. Circiuts & Systems, Vol. CAS-32,Juin 1985.

[25] V. TAVSANOGLU & L. THIELE, " Optimal Design of State-Space Digital Filters by Simultaneous Minimisation of Sensitivity and Roundoff Noise," IEEE Trans. Circuits & Systems, Vol. CAS-31,Oct 1984.

[26] L.B.JACKSON et Al., "Optimal Synthesis of Second Order State-Space Structures for Digital Filters," IEEE Trans. Circuits & Systems,Vol. CAS-26,Mars 1979.

[27] C.W.BARNES, "On the Design of optimal State-Space Realisations of Second Order Digital Filters," IEEE Trans. Circuits & Systems ,Vol. CAS-31,Juillet 1984.

[28] C.H.CHEN (ed), Signal Processing Handbook, Marcel DEKKER, INC, NY & BASEL 1988.

ANNEXE

Le programme **STUCMIN** *effectue le sectionnement du filtre en cellules du second ordre et construit leurs structures canoniques et optimales, leurs gains de bruit et ainsi que leurs matrices K et W respctifs. Les structures optimales sont établies suivant le choix d'une des trois procédures suivantes:*

* *Procédure de* **BOMAR** *qui est éxécutée par la subroutine* **MINOISE**
* *Procédure de* **BARNES** *qui est éxécutée par la subroutine* **MINOR**.
* *Procédure de* **HWANG** *qui est éxécutée par la subroutine* **HWAN**.

```
      PROGRAM STRUCMIN
      REAL*8 P(20),Q(20),Q1(20),D,AA(10,3),BB(10,2),BI(2),CI(2)
      Real*8 K(2,2),KI(2,2),WI(2,2),W(2,2),TR(2,2),A(2,2),DEL,G,G1
      Real*8 tn(2),td(3)
      COMPLEX*16 POL(10,2),pl
      Real*8 GM(10),GMN,alp,bet,AI(2,2),B(2),C(2),WK,X,AA1,BB1,CC1
      Open(2,file='Natej.dat',Status='NEW')
      Open(1,file='Buthba.DAT',status='OLD')
      Do 1 i=1,5
        Write(*,*)
    1 Continue
      Write(*,*)'      QUEL L''ORDRE DU FILTRE (N=<18) ? '
      Read(*,*) NO
      N=NO+1
      read(1,*) (Q(i),i=1,N)
      read(1,*) (P(i),i=1,N)
      WRITE(2,*)'                                          '
      Write(2,*)'       FILTRE D''ORDRE',NO,'              '
      WRITE(2,*)'                                          '
      WRite(2,*)
      Write(2,*)'** Les coefficients du numerateurs sont:'
      Write(2,*)
      DO 5 I=1,N
         Write(2,*) Q(i)
    5 Continue
      Write(2,*)
      Write(2,*)'** Les coefficients du denumerateur sont:'
      Do 6 I=1,N
        Write(2,*) P(I)
    6 Continue
C     IN :NOMBRE DE CELLULES DU 2nd ET 1er ORDRE
      IN=NO/2
      if (mod(NO,2).eq.1) Then
         IN=(NO+1)/2
         IMP=1
      Endif
C
```

```
C     DIVISION EUCLIDIENNE DE LA FONCTION DE TRANSFERT
C              G= Q/P = d+ Q1/P
      D=Q(1)/P(1)
      DO 11 I=2,N
      Q1(I-1)=Q(I)-P(I)*D
   11 CONTINUE
      Call decompos(P,Q1,BB,AA,POL,IN,NO)
      write(2,*)'****LE COEFF. DU CHEMIN DIRECT ENTREE/SORTIE EST:
      write(2,*)' D=',D
      Write(*,*)'    Quelle le facteur de normalisation (.GE.1.)?'
      READ(*,*)DEL
      Write(2,*)'
      write(2,*)'  Decomposition en cellules de structures directes
      Write(2,*)'
      Write(2,*)
      G=0.d0
      IR=IN
      IF(IMP.eq.1) IR=IN-1
      Do 10 i=1,IR
           Call Fcan(BB(i,1),BB(I,2),AA(i,2),AA(i,3),A,B,C,K,W)
           GM(i)=K(1,1)*W(1,1)+K(2,2)*W(2,2)
           Call Sortie(I,A,B,C,K,W,GM(i))
           G=G+GM(i)
   10 Continue
      G1=0.d0
      IF(imp.eq.1) Then
         x=(1-AA(IN,3)*AA(IN,3))
         WK=BB(IN,2)*BB(IN,2)/(x*x)
         BB1=BB(IN,2)
         AA1=-AA(IN,3)
         CC1=1.d0
         Call ORDRE1(AA1,BB1,CC1)
         G1=WK
      Endif
      Write(2,*)'  ***   GAIN TOTAL POUR LA FORME DIRECTE ***'
      write(2,12)  G
   12 Format(5x,'  GT =',F16.10)
C     Determination de la structure donnant un bruit de calcul minime
C     pour chaque cellule du 2nd ordre du filtre.
      write(*,*)'
      write(*,*)'
      write(*,*)'
      write(*,*)'
      write(*,*)'
      write(*,*)'
      write(*,*)'
      write(*,*)'
      write(*,*)'
      write(*,*)'
      write(*,*)'
      Write(*,*)
```

```
******* STRUCTURE OPTIMALE ********
        ~~~~~~~~~~~~~~~~~~~~~~~~

( 1 )  Methode de minimisation de BOMAR .

( 2 )  Methode de minimisation de BARNES ,
       (ou structure normale)

( 3 )  Methode de minimisation de HWANG .
```

```
      Write(*,*)
      write(*,*)'Quel est votre choix?'
      read(*,*) Ichoix
      If(Ichoix.eq.1) then
            Write(2,*)
            Write(2,*)'                                                  '
            Write(2,*)'                STRUCTURE OPTIMALE DE             '
            Write(2,*)'           ***       BOMAR        ***             '
            Write(2,*)'                                                  '
            Write(2,*)
            Do 20 i=1,IR
        Call MINOISE(I,BB(i,1),BB(i,2),AA(i,2),AA(i,3),GM(i),DEL)
 20         Continue
            Goto 50
      Endif
      If(Ichoix.eq.2) then
            write(2,*)
            Write(2,*)'                                                  '
            Write(2,*)'                STRUCTURE OPTIMALE DE             '
            Write(2,*)'           ***       BARNES       ***             '
            Write(2,*)'                                                  '
            Write(2,*)
            Write(2,*)
            Do 30 i=1,IR
               Alp=dreal(POL(i,1))
               Bet=dimag(POL(i,1))
               Call MINOR(I,BB(i,1),BB(i,2),alp,bet,GM(i),DEL)
 30         continue
            Goto 50
       Else
            Write(2,*)
            Write(2,*)'                                                  '
            Write(2,*)'                STRUCTURE OPTIMALE DE             '
            Write(2,*)'           ***       HWANG        ***             '
            Write(2,*)'                                                  '
           Do 40 i=1,IR
           Call Fcan(BB(i,1),BB(I,2),AA(i,2),AA(i,3),AI,BI,CI,KI,WI)
               Call Hwan(KI,WI,K,W,GM(i),TR,DEL)
               Call Trans(TR,AI,BI,CI,A,B,C)
               Call Sortie(i,A,B,C,K,W,GM(I)
 40        continue

      Endif
 50   GMN=0.d0
      Do 60 i=1,IR
      GMN=GMN+GM(i)
 60   Continue
      IF(imp.eq.1) THEN
         BB1=BB(IN,2)/DEL
         AA1=-AA(IN,3)
         CC1=DEL
         Call ORDRE1(AA1,BB1,CC1)
      Endif
```

```
Write(2,*)
Write(2,*)'***  GAIN TOTAL DE LA FORME OPTIMALE ***'
Write(2,*)'   GT=',GMN
write(*,*)'Les resultats sont dans le fichier'
write(*,*)'*****   " NATEJ.DAT"     ******** '
Stop
End

 SUBROUTINE MINOISE(I,q1,q2,p1,p2,GNM,DEL)
 Real*8 A(2,2),B(2),C(2),q1,q2,p1,p2,W(2,2),K(2,2)
 Real*8 v1,v2,v3,v4,v5,v6,v7,v8,GNM,tn(2),td(3)
 Real*8 G,DEL,D1
 Integer I
 D1=1.d0/DEL
 v1=q2/q1
 v2=dsqrt(v1*v1-p1*v1+p2)
 v3=v1-v2
 v4=v1+v2
 v5=p2-1.d0
 v6=p2+1.d0
 v7=v5*(v6*v6-p1*p1)
 v8=p1*p1/4.d0-p2
 A(1,1)=-p1/2.d0
 A(2,2)=A(1,1)
 B(1)=dsqrt(v7/(2.d0*p1*v3-v6*(1.d0+v3*v3)))*D1
 B(2)=dsqrt(v7/(2.d0*p1*v4-v6*(1.d0+v4*v4)))*D1
 A(2,1)=dsqrt((B(2)*B(2)+v5)*v8/(B(1)*B(1)+v5))
 A(1,2)=v8/A(2,1)
 C(1)=q1/(2.d0*B(1))*DEL
 C(2)=q1/(2.d0*B(2))*DEL
 Call Gauss(A,B,K,0)
 CAll Gauss(A,C,W,1)
 GNM=K(1,1)*W(1,1)+K(2,2)*W(2,2)
 Call Sortie(I,A,B,C,K,W,GNM)
 Return
 End

 SUBROUTINE MINOR(i,q1,q2,alp,bet,GNM,DEL)
 Real*8 A(2,2),A1(2,2),B(2),B1(2),C(2),C1(2),K(2,2)
 REAL*8 K1(2,2),W(2,2),W1(2,2),DEL,GF
 Real*8 alp,bet,q1,q2,phi,r,v,d1,d2,GNM
 Integer I
 v=(q2+alp*q1)/bet
 phi=datan2(-q1,v)
 r=dsqrt(q1*q1+v*v)
 A1(1,1)=alp
 A1(2,2)=A1(1,1)
 A1(2,1)=bet
 A1(1,2)=-A1(2,1)
 B1(1)=dsin(phi/2.d0)
 B1(2)=dcos(phi/2.d0)
 C1(1)=r*B1(2)
```

```
      C1(2)=r*B1(1)
      call Gauss(A1,B1,K1,0)
      d1=dsqrt(K1(1,1))
      d2=dsqrt(K1(2,2))
      A(1,1)=A1(1,1)
      A(2,2)=A(1,1)
      A(2,1)=bet*d1/d2
      A(1,2)=-bet*d2/d1
      B(1)=B1(1)/(d1*del)
      B(2)=B1(2)/(d2*del)
      C(1)=C1(1)*d1*del
      C(2)=C1(2)*d2*del
      Call Gauss(A,B,K,0)
      CAll Gauss(A,C,W,1)
      GF=K(1,1)*w(1,1)+k(2,2)*w(2,2)
      GNM=2*r*r*d1*d1*d2*d2
      Call Sortie(I,A,B,C,K,W,GNM)
      Return
      End

      Subroutine Hwan(K0,W0,K,W,GNM,TR,DEL)
      Real*8 K0(2,2),K(2,2),K1(2,2),W0(2,2),W(2,2)
      Real*8 TR(2,2),TC(2,2),T(2,2),R(2,2),U1,U2,DEL
      Real*8 T1(2,2),V,THETA,GNM,W1(2,2),W2(2,2)
C
C     TC represente la transformation de CHOLESKY .
C
      TC(2,1)=0.d0
      TC(2,2)=dsqrt(K0(2,2))
      TC(1,2)=K0(1,2)/TC(2,2)
      TC(1,1)=dsqrt(K0(1,1)*K0(2,2)-K0(1,2)*K0(1,2))/TC(2,2)
      Call Xmat2(TC,W0,W1)
      Theta=Datan(1.d0)
      If(W1(1,1).ne.W1(2,2)) Then
          Theta=Datan(2.d0*W1(1,2)/(W1(1,1)-W1(2,2)))/2.d0
      Endif
      R(1,1)=dcos(Theta)
      R(2,2)=R(1,1)
      R(2,1)=Dsin(Theta)
      R(1,2)=-R(2,1)
      Call Xmat2(R,W1,W2)
      U1=Dsqrt(W2(1,1))
      U2=Dsqrt(W2(2,2))
      V=U2/U1
      K(1,1)=1.d0/(Del*Del)
      K(2,2)=K(1,1)
      K(1,2)=(U1-U2)/((U2+U1)*K(1,1))
      K(2,1)=K(1,2)
      W(1,1)=(U1+U2)*(U1+U2)/(4.d0*K(1,1))
      W(2,2)=W(1,1)
      W(1,2)=(W2(2,2)-W2(1,1))/(4.d0*K(1,1))
      W(2,1)=W(1,2)
```

```
      GNM=K(1,1)*W(1,1)+K(2,2)*W(2,2)
C
C     T transformation de normalisation
C
      T(1,1)=dsqrt(1.d0+V)*Del/2.d0
      T(1,2)=-T(1,1)
      V=1.d0/V
      T(2,1)=DSQRT(1.d0+V)*Del/2.d0
      T(1,2)=T(2,1)
C
C     Calcul de la transformation globale TR=TC*R*T
C
      Call Mult2(Tc,R,T1)
      Call Mult2(T1,T,TR)
      return
      End

      Subroutine Trans(TR,AI,BI,CI,A,B,C)
      REAL*8 TR(2,2),Ai(2,2),A(2,2),BI(2,2),B(2,2),CI(2,2),C(2,2)
      REAL*8 TRI(2,2),AH(2,2)
      Call INVMAT2(TR,TRI)
      Call Mult2(TRI,AI,AH)
      Call Mult2(AH,TR,A)
      Call Vecmat(TRI,BI,B,1)
      Call vecmat(TR,CI,C,0)
      Return
      End

      Subroutine INVMAT2(A,B)
      Real*8 A(2,2),B(2,2),D
      D=A(1,1)*A(2,2)-A(1,2)*A(2,1)
      B(1,1)=A(2,2)/D
      B(1,2)=-A(1,2)/D
      B(2,1)=-A(2,1)/D
      B(2,2)=A(1,1)/D
      Return
      End

      Subroutine Xmat2(F,X,T)
      Real*8 X(2,2),F(2,2),T(2,2),S(2,2)
      Do 30 I=1,2
         Do 20 J=1,2
            S(I,J)=0.D0
            Do 10 K=1,2
               S(I,J)=S(I,J)+F(K,I)*X(K,J)
   10       Continue
   20    Continue
   30 Continue
      Do 60 I=1,2
         Do 50 J=1,2
            T(I,J)=0.D0
            Do 40 K=1,2
```

```
            T(I,J)=T(I,J)+S(I,K)*F(k,j)
40        Continue
50     Continue
60  Continue
    Return
    End

    Subroutine Fcan(q1,q2,p1,p2,A,B,C,K,W)
    REAL*8 q1,q2,p1,p2,tn(2),td(3),A(2,2),K(2,2),W(2,2)
    REAl*8 B(2),C(2)
    tn(1)=q1
    tn(2)=q2
    td(1)=1.d0
    td(2)=p1
    td(3)=p2
     Call State2(Tn,Td,A,B,C)
     Call Gauss(A,B,K,0)
     Call Gauss(A,C,W,1)
    Return
    End

    Subroutine State2(Q,P,A,B,C)
    Real*8 Q(2),P(3),A(2,2),B(2),C(2)
    A(1,1)=0.d0
    A(1,2)=1.d0
    A(2,1)=-P(3)
    A(2,2)=-P(2)
    B(1)=0.d0
    B(2)=1.d0
    C(1)=Q(2)
    C(2)=Q(1)
    Return
    End

    Subroutine Sortie(i,A,B,C,K,W,GNM)
    REAL*8 A(2,2),K(2,2),W(2,2),B(2),C(2),GNM
    write(2,*)'          ~~~~~~~~~~~~~~~~~~~~~~~~~~ '
    Write(2,*)'            CELLULE', I
    write(2,*)'          ~~~~~~~~~~~~~~~~~~~~~~~~~~'
    Write(2,*)'** MATRICE  A'
    Write(2,*)
    Write(2,*) A(1,1),A(1,2)
    Write(2,*) A(2,1),A(2,2)
    WRite(2,*)
    Write(2,*)'** VECTEUR B'
    WRite(2,*)
    Write(2,*) B(1)
    Write(2,*) B(2)
    Write(2,*)
    Write(2,*)'** VECTEUR C'
    Write(2,*)
    Write(2,*) C(1),C(2)
```

```
      Write(2,*)
      Write(2,*)'** MATRICE  K'
      Write(2,*)
      Write(2,*) K(1,1),K(1,2)
      Write(2,*) K(2,1),K(2,2)
      Write(2,*)
      Write(2,*)'** MATRICE  W'
      Write(2,*)
      Write(2,*) W(1,1),W(1,2)
      Write(2,*) W(2,1),W(2,2)
      Write(2,*)
      Write(2,*)'** GAIN  G',i
      Write(2,*)
      Write(2,*) GNM
      Write(2,*)
c     Write(2,*)' G1=',G1
      Return
      END

      Subroutine Vecmat(Mat,Vec,Res,Icas)
c
c     Produit D'une matrice par un vecteur:
c     Icas= 0   =>  Vect*Mat
c     Icas= 1   =>  Mat*Vec
c
         Real*8 Mat(2,2),Vec(2),Res(2),Temp
         Do 20 I=1,2
            Res(I)=0.d0
            Do 10 K=1,2
               Temp=Mat(I,K)
               If(Icas.eq.0) Temp=Mat(K,I)
               RES(I)=Res(I)+Vec(K)*Temp
10          Continue
20       Continue
         Return
         End

         Subroutine Mult2(A,B,C)
          Real*8 A(2,2),B(2,2),C(2,2)
          Do 10 I=1,2
             Do 20 j=1,2
                C(i,j)=0.d0
                Do 30 k=1,2
                   C(i,j)=C(i,j)+A(i,k)*B(k,j)
30              Continue
20           continue
10        Continue
      Return
      End

      SUBROUTINE Gauss(F,E,Res,Iflag)
C  Sous-programme de résolution d'un système d'equations linéaire
```

```
C                              [T][x]=[R]
C    Iflag=0 =>  Calcul de K
C    Iflag=1 =>  Calcul de w
C
      Real*8 T(3,3),R(3),Pivot,Temp,Cof
      Real*8 Res(2,2),Temp1,Temp2
      Real*8 F(2,2),E(2)
        Temp1=F(1,2)
        Temp2=F(2,1)
      IF(Iflag.eq.1) Then
        Temp1=F(2,1)
        Temp2=F(1,2)
      Endif
        T(1,1)=F(1,1)*F(1,1)-1.d0
        T(3,3)=F(2,2)*F(2,2)-1.d0
        T(2,2)=Temp1*Temp2+F(1,1)*F(2,2)-1.d0
        T(1,2)=2.d0*F(1,1)*Temp1
        T(1,3)=Temp1*Temp1
        T(2,1)=F(1,1)*Temp2
        T(2,3)=Temp1*F(2,2)
        T(3,1)=Temp2*Temp2
        T(3,2)=2.d0*Temp2*F(2,2)
        R(1)=-E(1)*E(1)
        R(2)=-E(1)*E(2)
        R(3)=-E(2)*E(2)
        N=3
        Ier=0
        Do 50 K=1,N-1
        Pivot=DABS(T(K,K))
        IP=K
         Do 10 I=K+1,N
           If (Pivot.Lt.DABS(T(I,K))) Then
              Pivot=DABS(T(I,K))
              IP=I
           Endif
10       Continue
         If (Pivot.EQ.0.D0) Goto 80
         If (IP.Ne.K) Then
            Do 20 I=K,N
              Temp=T(K,I)
              T(K,I)=T(IP,I)
              T(IP,I)=Temp
20          Continue
           Temp=R(K)
           R(K)=R(IP)
           R(IP)=Temp
         Endif
         Do 40 I=K+1,N
           Cof=T(I,K)/T(K,K)
            Do 30 J=K+1,N
              T(I,J)=T(I,J)-Cof*T(K,J)
30          Continue
```

```
            R(I)=R(I)-Cof*R(K)
   40     Continue
   50  Continue
       If (T(N,N).Eq.0.D0) Goto 80
       R(N)=R(N)/T(N,N)
       Do 70 I=1,N-1
         K=N-I
         Temp=0.D0
          Do 60 J=K+1,N
            Temp=Temp+T(K,J)*R(J)
   60     Continue
         R(K)=(R(K)-Temp)/T(K,K)
   70  Continue
         Res(1,1)=R(1)
         Res(1,2)=R(2)
         Res(2,1)=R(2)
         Res(2,2)=R(3)
       Return
   80  Ier=1
       write(*,*)'Pivot nul'
       Return
       End

       Subroutine DECOMPOS(A,B1,BB,AA,POL,IN,N0)
       REAL*8 A(20),B1(20),D,AA(10,3),BB(10,2)
       COMPLEX*16 POL(10,2)
C
C    CALCUL DES POLES.
C
       CALL BAIRSTOW(A,N0,IN,POL,AA)
C
C    DECOMPOSITION EN SECTIONS DU 2nd ORDRE.
C
       CALL RESIDU(POL,B1,N0,IN,BB)
       CALL CELL(N0,IN,POL,AA,BB)
       Return
       End

       SUBROUTINE BAIRSTOW(A,M,IN,POL,AA)

*-------------------------------------------------------------------*
*     SUB. BAIRSTOW RECHERCHE LES COUPLES DE POLES  D'UN FILTRE     *
*     PAR LA METHODE DE BAIRSTOW.                                   *
*     LES PARAMETRES TRANSMIS SONT:                                 *
*       M    : DEGRES DU POLYNOME.                                  *
*     A(M+1): LES COEFFICIENTS DU DENOMINATEUR DE LA FONCTION       *
*             DE TRANSFERT DU FILTRE D'ORDRE  M.                    *
*       IN   : NOMBRE DE CELLULES DU  2nd  ORDRE;                   *
*              IN=M/2     SI M EST PAIR,TOUTES LES CELLULES        *
*                          SONT DU 2eme ORDRE.                      *
*              IN=(M-1)/2 SI M EST IMPAIR,LA (IN+1) CELLULE EST    *
*                          DU 1er ORDRE.                            *
```

```
*     LES RESULTATS DU CALCUL SONT:                                    *
*        POL(IN,2) : MATRICE DES POLES DONT CHAQUE  LIGNE              *
*                    REPRESENTE UNE SECTION DU 2nd ORDRE ET            *
*                    CONTIENT 2 POLES (qui pourraient etre             *
*                    complexes conjugues).                             *
*        AA(IN,3)  : MATRICE DONT CHAQUE  LIGNE REPRESENTE UNE         *
*                    SECTION DU 2nd ORDRE ET CONTIENT LES              *
*                    COEFFICIENTS DU DENOMINATEUR CORRESPONDANT.       *
*--------------------------------------------------------------------*

      COMPLEX*16 POL(10,2)
      REAL*8  A(20),B(20),AA(10,3),R(20),S,S1,P,P1,C,D,X,Y,E
      CALL EPSILON(E)
      E=E**(2.d0/3.d0)
      K=M+1
      j=1
  100 S=-A(2)/A(1)
      P=A(3)/A(1)
C
C     INITIALISATION DE LA SOMME S ET DU PRODUIT P  DES RACINES.
C
      B(1)=A(1)
      L=0
   60 L=L+1
      B(2)=A(2)+S*B(1)
      DO 20 I=3,K
      B(I)=A(I)+S*B(I-1)-P*B(I-2)
   20 CONTINUE
C
C    LE VECTEUR B CONTIENT LES COEF. DU POLYNOME QUOTIENT.
C
      R(1)=0.D0
      R(2)=B(1)
      DO 30 I=3,K
      R(I)=B(I-1)+S*R(I-1)-P*R(I-2)
   30 CONTINUE
C
C    LE VECTEUR R CONTIENT LES DERIVEES PARTIELLES DES Bi PAR RAPPORT
C    A S ET P.
C
      D=R(K)*R(K-2)-R(K-1)*R(K-1)
      S1=S-(B(K)*R(K-2)-B(K-1)*R(K-1))/D
      P1=P-(B(K)*R(K-1)-B(K-1)*R(K))/D
C
C    ITERATION DES VALEURS DE S ET P.
C
      IF((ABS(S1-S)+ABS(P1-P)).LT.E)  GOTO 50
      S=S1
      P=P1
      GOTO 60
   50 D=S1*S1-4*P1
      IF(D.LT.0.D0) GOTO 70
```

```
C
C      C   SERT SEULEMENT A ORDONNER LES POLES TROUVES.
C
       C=-1.D0
       IF(S1.GT.0.D0) C=1.D0
       D=DSQRT(D)
       X=(S1+C*D)/2.D0
       Y=P1/X
       POL(J,1)=CMPLX(X,0.D0)
       POL(J,2)=CMPLX(Y,0.D0)
       AA(J,1)=1.D0
       AA(J,3)=P1
       AA(J,2)=-S1
       GOTO 80
    70 D=DSQRT(-D)
       POL(J,1)=CMPLX(S1/2,D/2)
       POL(J,2)=CMPLX(S1/2,-D/2)
       AA(J,1)=1.D0
       AA(J,3)=P1
       AA(J,2)=-S1
    80 K= K-2
       J=J+1
       DO 90 I=1,K
       A(I)=B(I)
    90 CONTINUE
       IF(K.GT.3) GOTO 100
       IF(K.EQ.2) GOTO 110
       IF(K.EQ.1) GOTO 120
       S1=-A(2)/A(1)
       P1= A(3)/A(1)
       GOTO 50
   110 Continue
       POL(in,1)=CMPLX(-A(2)/A(1),0.D0)
       AA(IN,1)=0.D0
       AA(IN,2)=1.D0
       AA(IN,3)=A(2)/A(1)
   125 FORMAT(2X,4F16.10)
   120 Continue
       do 121 i=1,in
         do 122 j=1,2
          write(*,*)'pol(',i,',',j,')=',pol(i,j)
   122   continue
   121 continue
       RETURN
       END

       SUBROUTINE EPSILON(EPS)
*-------------------------------------------------------------*
*   Cette subroutine calcule le plus petit reel pouvant etre  *
*   represente par la machine.                                *
*-------------------------------------------------------------*
       Real*8 EPS
```

```
      EPS=1.D0
   10 If ((1.D0+EPS).GT.1.D0) then
         EPS=EPS/2.D0
         GOTO 10
      Endif
      EPS=EPS*2.D0
      Return
      End

      SUBROUTINE RESIDU(POL,B1,N0,IN,BB)
      Complex*16 pol(10,2),R,Num,Den,v(20)
      Real*8 B1(20),BB(10,2)
      Integer IN
      Do 5 i=1,IN
c
c     Calcul de la valeur du polynome numerateur de la fonction de
c     transfert par l' algorithme de HORNER . Le resultat est Num.
c
         V(1)=cmplx(0.d0,0.d0)
         Do 15 k=1,N0
              V(k+1)=B1(k)+V(k)*pol(i,1)
   15    Continue
         Num=v(N0+1)
         Den=cmplx(1.d0,0.d0)
         Do 25 j=1,IN
              if(i.eq.j) goto 25
              Den=Den*(pol(i,1)-pol(j,1))
   25    Continue
         Do 35 j=1,IN-1
                Den=Den*(pol(i,1)-pol(j,2))
   35    Continue
         If((2*IN).EQ.N0) then
            Den=Den*(pol(i,1)-pol(IN,2))
         Endif
C
C       R  est le residu du 1er pole de la cellule i.
C
         R=Num/Den
         If(DIMAG(POL(IN,1)).EQ.0.d0) goto 65
C
C     Les BB sont les coefficients du numerateur (polynome du 1er
C     ordre: BB1*z+BB2 )
C
         BB(i,1)= 2*dreal(R)
         BB(i,2)=-2*dreal(pol(i,2)*R)
    5 Continue
      Goto 75
C
C    Consideration du cas d'un pole reel simple :
C
   65  BB(IN,2)=dreal(R)
       BB(IN,1)=0.d0
```

```
 75 return
    End

    SUBROUTINE CELL(NO,IN,POL,AA,BB)
    Complex*16 pol(10,2)
    Real*8 AA(10,3),BB(10,2)
    write(2,*)
    Write(2,*)'
    Write(2,*)'     ** DECOMPOSITION EN CELLULES DU 2nd ORDRE **
    Write(2,*)'
    Do 20 i=1,IN
       WRITE(2,*)'┌──────────────────────┐'
       Write(2,*)  │  CELLULE',i        ' │'
       write(2,*)'└──────────────────────┘'
       Write(2,*)
       write(2,*) ' ***Les coefficients du numerateur sont :'
       write(2,45) BB(i,1)
       write(2,45) BB(i,2)
       write(2,*) ' ***Les coefficients du denominateur sont :'
       do 55 k=1,3
          write(2,45) AA(i,K)
55     Continue
       IF(I.eq.IN) Then
          If(Mod(NO,2).eq.1) then
             Write(2,*)'     **** UN POLE SIMPLES REEL:'
             Write(2,*) Dreal(POL(IN,1))
           Else
-             GOTO 30
          Endif
       Endif
30     write(2,*)'  ***Les 2 poles complexes sont :'
       write(2,46) Dreal(pol(i,1)),Dimag(pol(i,1))
45     Format(5x,f16.10)
46     format(5x,f16.10,'+/-j',F16.10)
20  Continue
    Return
    end

    Subroutine ORDRE1(AA1,BB1,CC1)
    Real*8  AA1,BB1,CC1
       Write(2,*)'   ~~~~~~~~~~~~~~~~~~~~~~~~~~'
       write(2,*)'    CELLULE DU 1er ORDRE    '
       Write(2,*)'   ~~~~~~~~~~~~~~~~~~~~~~~~~~'
       Write(2,*)
       WRITE(2,*)'*** A = ',AA1
       Write(2,*)
       WRITE(2,*)'*** B = ',BB1
       Write(2,*)
       Write(2,*)'*** C = ',CC1
       WRITE(2,*)
     Return
     End
```