REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère des Universités et de la Recherche Scientifique

ECOLE NATIONALE POLYTECHNIQUE

Département d'Electronique

Laboratoire Traitement du Signal

THESE DE MAGISTER

Présentée par: Djamel ADDOU

Ingénieur Diplômé de l'E.N.P.

SYNTHESE OPTIMALE DES FILTRES NUMERIQUES RECURSIFS REALISES PAR DES STRUCTURES D'ETAT GLOBALE ET DECOMPOSEE AVEC UNE ARITHMETIQUE DE CALCUL A VIRGULE FIXE

Soutenue en Juin 1992 devant le jury composé de:

MM.	A. CHEKIMA	Professeur	Président
	B. DERRAS	PHD	Rapporteur
	A. FARAH	Maître de conférences	Examinateur
	C. BENMEHREZ	PHD	Examinateur
	F. CHIGARA	Maître Assistant	Examinateur

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère des Universités et de la Recherche Scientifique

ECOLE NATIONALE POLYTECHNIQUE

Département d'Electronique

Laboratoire Traitement du Signal

THESE DE MAGISTER

*Présentée par: **Djamel ADDOU***

Ingénieur Diplômé de l'E.N.P.

SYNTHESE OPTIMALE DES FILTRES NUMERIQUES RECURSIFS REALISES PAR DES STRUCTURES D'ETAT GLOBALE ET DECOMPOSEE AVEC UNE ARITHMETIQUE DE CALCUL A VIRGULE FIXE

Soutenue en Juin 1992 devant le jury composé de:

MM.	A.	CHEKIMA	Professeur	Président
	B.	DERRAS	PHD	Rapporteur
	A.	FARAH	Maître de conférences	Examinateur
	C.	BENMEHREZ	PHD	Examinateur
	F.	CHIGARA	Maître Assistant	Examinateur

بسم الله الرحمان الرحيم

الإهداء

- إلى التي حملتني وهنا على وهن
- إلى والدي الكريم -حفظه الله تعالى- الذي غرس في نفسي روح التضحية و نكران الذات .
- إلى إخواني و أخواتي
- إلى كل الذين امنوا بالعلم و تعركوا بالعلم و صبروا في سبيل العلم من أجل العلم
- إلى كل أولئك أهدي ثمرة جهدي.

جمال عدو

REMERCIEMENTS

Il m'est particulièrement agréable d'exprimer à Monsieur B. DERRAS, PhD et directeur de recherche, ma profonde gratitude pour sa sollicitude, ses conseils efficaces et ses encouragements permanents qu'il m'a prodigués tout le long de ce travail. Que ce document témoigne de ma reconnaissance qui lui est due.

J'exprime mes plus vifs remerciements à Monsieur A. CHEKIMA, Professeur à l'E.N.P. pour m'avoir fait l'honneur de présider ce jury de thèse.

J'adresse mes remerciements les plus sincères à Messieurs A. FARAH, Maître de conférences, C. BENMEHREZ, PhD et F. CHIGARA Maître assistant, membres du jury pour avoir examiné avec toute l'attention voulue ce travail.

Syhthése Optimale Des Filtres Numériques Recursifs Réalisés Par Des Structures D'Etat Globale Et Décomposée Avec Une Arithmétique De Calcul A Virgule Fixe

par
Djamel Addou
Ingénieur électronique, Ecole Nationale Polytechnique
Juin 1987

RÉSUMÉ

Dans cette étude, on s'intéresse principalement à des filtres numériques récursif représentés par les variables d'état et réalisés avec des structures qui minimisent le bruit d'arrondi et reduisent le nombre de multiplieurs. Aussi, afin d'analyser le comportement de ces filtres vis-à-vis de la limite de précision de toutes les valeurs intervenant dans l'opération du filtrage, deux types de structures sont étudiées:

- Structures avec minimum de bruit de calcul et $(N+1)^2$ multiplieurs.
- Structures de compromis basées sur la décomposition du filtre en des sections de second ordre, avec (4N+1) multiplieurs.

SOMMAIRE

S.O.M.M.A.I.R.E

CHAPITRE

CHAPITRE I

INTRODUCTION

CHAPITRE I

INTRODUCTION

1.1 Généralités

La pénétration du numérique dans les sciences de l'ingénieur , une fois passée le stade de la transposition du filtre analogique en algorithme de calcul, a conduit à une théorie propre au traitement du signal; en vue d'extraire une information mieux adaptée et ce avant même sa quantification. Alors le pas et la finesse de la numérisation peuvent être optimisés en vue de la charge du calculateur ou la taille du microprocesseur du filtrage.

Le filtrage numérique représente l'ensemble des opérations, calculs arithmétiques et manipulation de nombres, qui sont éffectués sur un signal à traiter, représenté par une suite ou un ensemble de nombres, en vue de fournir une autre suite ou un autre ensemble de nombres, qui représente le signal traité.

En traitement du signal, les techniques numériques apportent des possibilités avantageuses et prodigieuses: La conception rigoureuse des systèmes, une grande reproductibilité des équipements, une grande stabilité dans le temps, la précision et la miniaturisation. Mais, est-ce dire qu'il n'y a aucune contrainte liée à la réalisation d'un filtre numérique ? Certes Non, et on en citera essentiellement deux, rencontré dans ce travail:

- Le problème de la précision des calculs, lié à la longueur finie des mots binaires sur lesquels travaille l'unité de calcul et les convertisseurs analogique-numérique (C.A.N).
- Le problème de la rapidité de calcul, lié à la conception des structures du filtre numérique.

En fait, pour mettre en oeuvre un filtre numérique, on exécute les différentes étapes suivantes:

1°- Spécification prototype du filtre.

2°- Synthèse du filtre.

3°- Choix de la structure ayant une bonne performance et un faible coût.

4°- Simulation et test de la structure choisie.

5°- Réalisation et implantation sur un " Hardware spécialisé ".

Une fois spécification et synthèse du filtre obtenues, l'objectif de notre étude se limite aux étapes 3 et 4 sus-cités. Ainsi, lors de la réalisation du filtre sur " Hardware spécialisé " utilisé en temps réel (étape 5) à l'aide d'une unité de calcul travaillant en virgule fixe sur des mots de longueur finie, on est obligé de tenir compte de l'inextricabilité des erreurs de calculs d'ailleurs indésirées. Tandis qu'avec l'utilisation d'un ordinateur à usage général, pour un traitement en temps différé, le problème de ces erreurs ne se pose pas vu la grande précision de l'ordinateur.

1.2 Description synoptique de la thèse

Ce travail traite l'étude des filtres numériques récursifs linéaires invariants dans le temps, pour la raison que ces derniers présentent des réalisations efficaces et ont plusieurs propriétés intéressantes caractérisées par la forme raide de la réponse du filtre et par la haute qualité du signal de sortie. Cependant, cette étude peut être approchée par deux méthodes différentes: La première est basée sur la synthèse et l'analyse de la fonction de transfert (description externe), malheureusement, celle-ci présente quelques problèmes imprévisibles dues au manque d'informations dans la fonction de transfert. Ceci peut être remédier par l'utilisation du seconde approche qui utilise les réalisations d'espace d'état, en spécifiant les structures du filtre avec plusieurs représentations. Il présente une information complète, concernant le comportement interne du filtre et un modèle mathématique traitable dans l'ensemble.

Par l'utilisation de la réalisation d'espace d'état, des nouvelles structures des filtres numériques récursifs sont synthètisées et évaluées par leur performances dans cette étude. En particulier, ces structures sont caractérisées avec quelques propriétés souhaitables, telle que l'efficacité de calcul et faible bruit d'arrondi vis-à-vis des effets de la longueur finie du mot.

Cependant, les structures optimales ainsi obtenues présentent une certaine complexité dans leur réalisation, vu le nombre élevé d'opérations à effectuer. Ce qui nous amène à chercher des structures de compromis qui ont pour but de maintenir la réduction du bruit de calcul et simplifier la complexité de réalisation matérielle. De ce fait, on étudiera d'autres structures qui peuvent surélever ce problème; ceux sont les structures décomposées en des cellules de second ordre, connectées en parallèle ou en cascade.

Actuellement, il existe deux méthodes de base permettant de construire les structures à faible bruit de calcul: la méthode de S. Hwang [1] et celle de Mullis-Roberts [2]. Pour le cas des structures optimales de second ordre, en dehors de ces deux dernières méthodes d'optimisation , il existe celle de B. Bomar [3] et W. Barnes [4]. Ces méthodes diffèrent du point de vue traitement mathématique, mais aboutissent toutes au même gain minimum de bruit de calcul. De ce fait, dans notre travail l'approche méthodologique utilisée pour la synthèse des nouvelles structures avec minimum d'erreurs de calcul est celle de S. Hwang [1]. Avec cette méthode, les structures dérivées (optimale et décomposée optimale) de la canonique ou forme directe, sont évaluées et comparées selon leur complexités et performances.

1.3 Organisation générale de la thèse

Après le chapitre d'introduction, le chapitre II portera sur des généralités de réalisations des filtres numériques récursifs, en particulier leur représentations par les variables d'état. Cette théorie préliminaire de représentation est utile pour la suite du travail.

Dans le chapitre III, après avoir rappelé quelques possibilités de codage des nombres, on consacrera entièrement une partie pour donner quelques principes qui permettent de calculer les erreurs propres aux structures de réalisations choisies des filtres numériques.

Les chapitres IV et V, sont consacrés au développement en détail de la procédure d'optimisation des erreurs de calcul à la

sortie du filtre, on trouvera donc, les structures optimales et décomposées optimales. Dans chaque chapitre, des exemples numériques sont utilisés, pour présenter et interpréter les résultats obtenus par simulation.

Et enfin, on termine ce travail par une conclusion générale au chapitre VI.

CHAPITRE II

REPRESENTATION DES FILTRES NUMERIQUES

CHAPITRE II

REPRÉSENTATION DES FILTRES NUMÉRIQUES

2.1 Introduction

Un filtre numérique est un algorithme de calcul par lequel une séquence de nombre u(k) dite séquence d'entrée est transformée en une autre séquence de nombre y(k) dite séquence de sortie.

D'une manière générale, un filtre numérique est constitué par (Fig 2.1) :

-Un ou plusieurs organes de retard (registres à décalage).
-Des sommateurs et des multiplicateurs.
-Des registres fournissant les coefficients de pondération du filtre.

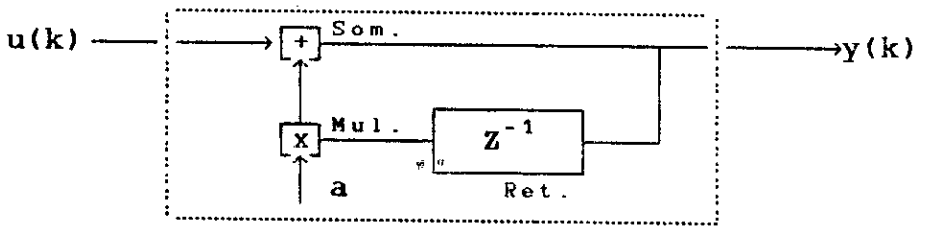


Fig 2.1: Organisation d'un filtre numérique

On distingue parmi un grand nombre de famille de ces filtres, les filtres récursifs à réponse impulsionnelle infinie (R.I.I) et non récursifs à réponse impulsionnelle finie (R.I.F) [5], [6], dont les principales propriétés sont résumées dans le tableau ci-dessous (Tab 2.1).

2.2 La formulation externe

La relation qui lie la suite des valeurs d'entrée u(k) aux valeurs de sortie y(k) constitue une opération linéaire discrète

Tab 2.1: Propriétés des filtres "R.I.F" et "R.I.I"

MODE	FILTRE RECURSIF (R.I.I)	FILTRE NON RECURSIF (R.I.F)
foction de transfert H(Z)	des pôles et des zéros	uniquement des zéros
coéfficient h(n) de la réponse impulsionnelle	infinité de valeurs non nulles	tous nuls, sauf pour un nombre fini de valeurs
longueur de mémoire	infinie	finie
stabilité	stable (si les pôles de H(Z) sont à l'intérieur du cercle unité)	toujours stable

utilisant des coefficients de pondération (a_i,b_i), cette opération est donnée par l'équation de récurrence suivante:

$$y(k) = \sum_{i=0}^{N} b_i u(k-i) - \sum_{i=1}^{N} a_i y(k-i) \qquad (2.1)$$

où N représente l'ordre du filtre. L'analyse des systèmes discrets peut s'effectuer grâce à la transformée en Z. Cette dernière appliquée à (2.1), fournit la fonction de transfert du filtre numérique, donnée par:

$$H(Z) = \frac{\sum_{i=0}^{N} b_i Z^{-i}}{1 + \sum_{i=1}^{N} a_i Z^{-i}} \qquad (2.2)$$

Si l'on développe H(Z) sous la forme d'une série en Z^{-1}, qui apparait comme un facteur de retard, la sortie y(k) peut s'exprimer par la convolution discrète des coefficients h(k) avec les valeurs de l'entrée u(k), i.e:

$$y(k) = \sum_{i=0}^{\infty} h(i)\ u(k-i) = \sum_{i=-\infty}^{k} h(k-i).u(i) = h(k)*u(k)$$

et

$$H(Z) = \sum_{k=0}^{\infty} h(k).Z^{-k} \qquad (2.3)$$

L'inconvénient de la formulation externe réside dans le fait qu'on ne peut rien prévoir sur le comportement des différentes variables internes du filtre, d'où recours à la formulation d'état.

2.3 La formulation d'état

Dans une représentation d'état, on recherche donc l'information nécessaire et suffisante qu'il convient de connaître à l'instant k pour prédire le comportement futur du système dans l'intervalle de temps (k,k+1) connaissant les causes dans un même intervalle. Cet ensemble de paramètres x(k) qui est une mémoire sélective condensée du système, s'appele l'état du système. On aura alors la relation suivante:

$$y(k) = f\ [\ x(k),u(k)\ ] \qquad (2.4)$$

Si on applique cette relation à l'état lui même, on aura l'équation d'état du système représentée par:

$$\begin{cases} x(k+1) = f_1\ [\ x(k),u(k)\ ] \\ y(k) = f_2\ [\ x(k),u(k)\ ] \end{cases} \qquad (2.5)$$

Lorsque f_1 et f_2 sont des relations linéaires à paramètres constants il vient par suite:

$$\begin{cases} x(k+1) = A\ x(k) + B\ u(k) \\ y(k) = C\ x(k) + D\ u(k) \end{cases} \qquad (2.6)$$

où x(k) est le vecteur d'état qui donne à tout instant l'état du système. Aussi, la dimension minimale de ce vecteur représente l'ordre du filtre (système) donnée par N.

* y(k) et u(k) représentent la sortie et l'entrée, respectivement.

* A La matrice d'évolution (NxN).

* B La matrice de commande (Nx1).

* C La matrice d'observation (1xN).

* D La matrice de transmission (1x1).

La représentation par les variables d'état, permet de donner une structure qui lie les différentes relations entre les variables internes du filtre. Pour chaque fonction de transfert, on peut trouver un nombre infini de représentations par les variables d'état. Toutes les représentations sont liées par des transformations non singulières.

Exemple: Représentation Canonique

A partir de la fonction de transfert (2.2), on donne la représentation temporelle du filtre :

$$y(k) = \sum_{i=0}^{N} b_i u(k-i) - \sum_{i=1}^{N} a_i y(k-i) \qquad (2.7)$$

en utilisant la transformée en Z, on obtient la fonction de transfert suivante:

$$H(Z) = \frac{Y(Z)}{U(Z)}$$

où U(Z) et Y(Z) sont les transformées en Z de l'entrée et de la sortie du filtre respectivement. On pose:

$$\frac{Y(Z)}{U(Z)} = \frac{N(Z)}{D(Z)} \quad , \text{ et } W(Z) = \frac{U(Z)}{D(Z)}$$

ce qui donne:

$$u(k) = w(k) + a_1 w(k-1) + \ldots + a_N w(k-N). \qquad (2.8)$$

Par linéarité on trouve:

$$y(k) = \sum_{i=0}^{N} b_i w(k-i) \qquad (2.9)$$

Les deux équations (2.8) et (2.9) donnent le schéma canonique présenté par la figure (fig 2.2).

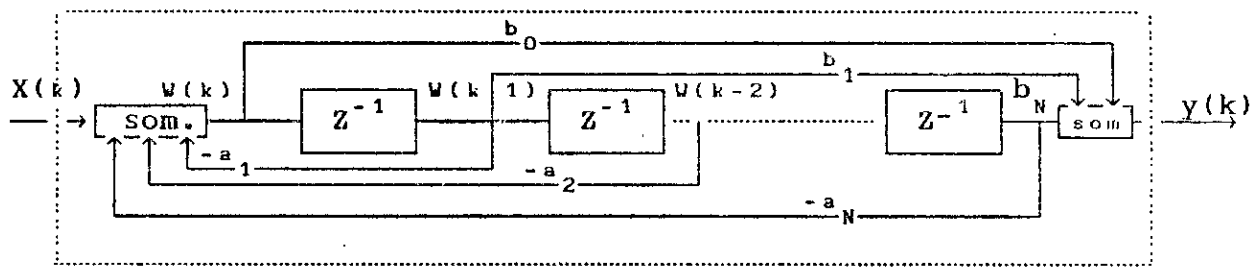
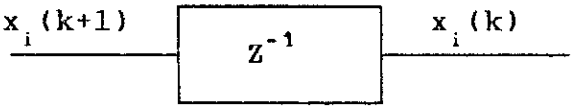


Fig 2.2: Structure canonique d'un filtre numérique

Si on choisit chaque élément de retard Z^{-1} comme variable interne:



On trouve la formulation d'état suivante:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ \vdots \\ x_N(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & & 1 \\ -a_N & \cdots & & -a_2 & -a_1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_N(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} u(k) \quad (2.10a)$$

$$y(k) = \begin{bmatrix} (b_N - a_N b_0) & \cdots & (b_2 - a_2 b_0) & (b_1 - a_1 b_0) \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_N(k) \end{bmatrix} + b_0 . u(k) \quad (2.10b)$$

On retrouve la forme (2.6). En termes de (A,B,C,D), la réponse impulsionnelle causale h(k) pour un tel filtre est donnée par:

$$h(k) = \begin{cases} 0 & k<0 \\ D & k=0 \\ C\,A^{k-1}\,B & k>0 \end{cases} \quad (2.11)$$

et

$$H(Z) = C\,(Z\,I - A)^{-1}\,B + D \quad (2.12)$$

Remarque: Si on applique une transformation quelconque (non singulière) au vecteur d'état x(k), on trouve une autre représentation d'état mais caractérisée par la même fonction de transfert. Soit x'(k) un autre vecteur d'état tel que:

$$x'(k) = T^{-1} x(k) \quad (2.13)$$

ou T désigne une transformation d'état non singulière (NxN). La nouvelle représentation d'état est donnée par:

$$\begin{cases} x'(k+1) = (T^{-1}A\ T)\ x'(k) + (T^{-1}B)\ u(k) \\ Y(k) \quad\ = (CT)\ x'(k) + D\ u(k) \end{cases} \tag{2.14}$$

La structure (A,B,C,D) devient après cette transformation:

$$(T^{-1}A\ T,\ T^{-1}B,\ CT,\ D) \tag{2.15}$$

mais la fonction de transfert H(Z) et la réponse impulsionnelle h(k) restent invariantes.

NOTE: La stabilité du filtre, mis sous la forme d'état, est vérifiée lorsque les modules des valeurs propres de la matrice A dans l'expression x(k+1)=A x(k)+B u(k), sont inférieurs à l'unité. Aussi, les valeurs propres de A sont les pôles du filtre.

2.4 Conclusion

Etant donné, que tout système numérique n'effectue que trois types d'opérations: les additions, les multiplications et les décalages; la réalisation d'un filtre numérique exige en plus de la détermination de la fonction de transfert (formulation entrée-sortie), la connaissance de l'hiérarchie des opérations élémentaires à exécuter. Cependant, il devient impératif de concevoir une représentation plus détaillée du filtre, afin de décrire les opérations internes du filtrage, d'où recours à la formulation d'état, qui permet de connaître la manière dont s'effectuent les opérations à l'intérieur du filtre, de mieux analyser les erreurs internes, et constituer en plus d'un outil mathématique puissant, un moyen d'analyse souple et efficace. De plus, le fait qu'il y a une infinité de représentations d'état, on peut choisir la meilleur représentation vis-à-vis des performances souhaitées.

CHAPITRE III

ANALYSE DES ERREURS DE CALCUL DANS LES FILTRES NUMERIQUES

CHAPITRE III

ANALYSE DES ERREURS DE CALCUL DANS LES FILTRES NUMERIQUES

3.1 Introduction

Une fois la synthèse effectuée et la structure choisie pour un filtre numérique, l'implantation de ce dernier sur un hardware nécessite l'écriture et l'exploitation d'un programme de calcul correspondant à une succession de taches telles que la conversion des coefficients en nombres binaires, leur stockage en mémoire et l'exécution d'opérations arithmétiques (additions, multiplications, décalage). On peut juger les performances de ce programme et du filtre qu'il représente selon des critères tels que :

-La taille du programme.

-Le temps d'exécution.

-Le nombre de coefficients.

-La précision des résultats obtenus.

Pour apprécier a priori la qualité d'une mise en oeuvre, il devrait y avoir un compromis entre tous ces critères et aboutir à une implantation possible à faibles erreurs, autrement dit le dernier mot reste donc à la simulation. A ce titre, nous donnons quelques principes qui permettent de calculer les erreurs propres aux structures de réalisations choisies.

3.2 Bruit de quantification

La quantification est une règle de correspondance entre le nombre infini de valeurs possibles du signal d'entrée u(t) et un nombre fini de valeurs assignées au signal de sortie $u_q(t)$. La règle de correspondance se traduit par une approximation $u_q(k)$ de la valeur de l'entrée u(kT) à une valeur voisine que l'on peut obtenir par arrondi ou par troncature avec un pas de quantifica-

tion q qui joue un rôle analogue au bit de poids le plus faible dans le codage binaire des nombres . Toutes les valeurs d'entrée appartenant au même intervalle sont donc représentées par le même niveau quantifié, qui correspond généralement à la valeur médiane de l'intervalle pour la quantification par arrondi ou à sa valeur minimale pour la quantification par troncature (Fig 3.1). Un tel processus introduit naturellement une distorsion intrinsèque qui

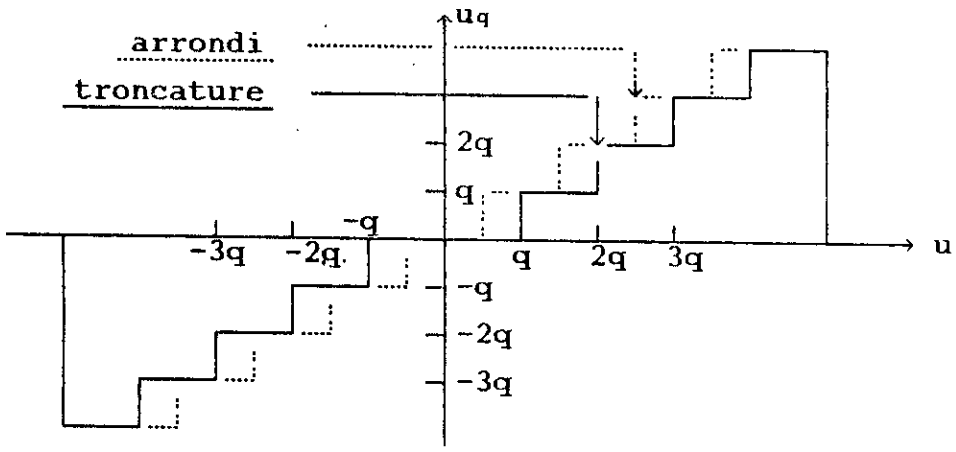


Fig 3.1: Effet de l'arrondi et de la troncature sur un signal

dépend autant de la nature du signal que de la loi de quantification adoptée. L'effet de cette approximation est de superposer au signal d'origine, un signal d'erreur e(k) désignée par distorsion ou bruit de quantification, il vient ainsi:

$$u_q(k) = u(k) + e(k) \qquad (3.1)$$

On formule ainsi le problème en termes statistiques, en interprétant l'erreur de mesure comme étant une variable aléatoire additionnelle (Fig 3.2). L'hypothèse de calcul formulée est que, moyennant un pas de quantification suffisamment petit, la densité de probabilité associée est supposée uniformément répartie sur une plage de valeurs q. On admet de plus, que l'erreur commise n'est pas corrélée avec le signal. Ces hypothèses permettant d'exprimer

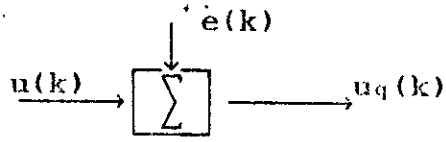


Fig 3.2: Interprétation statistique d'un quantificateur

le lien entre les propriétés statistiques du signal quantifié et du signal de départ. De ce fait, l'erreur due à l'arrondi ou la troncature peut être statistiquement modélisée comme une source de bruit additive. La moyenne m_e et la variance σ_e^2 de ce bruit dépend de la distance entre deux nombres adjacents quantifiés. Si q représente le pas de quantification, on aura dans le cas d'arrondi:

$$\begin{cases} m_e = 0. \\ \sigma_e^2 = q^2/12. \end{cases} \qquad (3.2)$$

et dans le cas de la troncature:

$$\begin{cases} m_e = q / 2. \\ \sigma_e^2 = q^2/ 12. \end{cases} \qquad (3.3)$$

où m_e et σ_e^2 représentent la moyenne et la variance, respectivement, du bruit de quantification.

3.3 Effets du codage des nombres sur calculateurs

Il existe divers 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 général, positives et négatives; les représentations préférées sont celles qui conservent l'information du signe. En ce qui concerne la position de la virgule, le calculateur utilise deux arithmétiques:

Arithmétique à virgule fixe

Le mot binaire représente soit un nombre entier, soit un nombre fractionnaire selon que par convention, on fait l'hypothèse d'une virgule à droite ou à gauche du mot binaire. Une fois la position de la virgule est choisie, on peut adopter pour les opérations le principe de l'arrondi ou la troncature sur le résultat. L'arrondi convertit le résultat en nombre codé le plus proche. La

troncature abandonne les bits de poids les plus faibles non représentatifs. Si b désigne la base, n le nombre de bits significatifs, un majorant e_m de l'erreur peut être donné dans le cas des nombres fractionnaires de la façon suivante [6],[7]:

$$e_m = \frac{b^{-n}}{2} \quad \text{cas de l'arrondi}$$

$$e_m = b^{-n} \quad \text{cas de la troncature}$$

Arithmétique à virgule flottante

La précédente arithmétique présente l'inconvénient d'avoir une gamme dynamique faible. Pour remédier à ce problème, on utilise l'arithmétique à virgule flottante, où le nombre s'écrit sous la forme suivante :

$$a = m\, b^e$$

m représente un nombre fractionnaire désignant la mantisse et e un nombre entier désignant l'exposant. La base étant représentée par b. Dans cette arithmétique, les erreurs de troncature ou d'arrondi n'affectent que la mantisse. De même si n désigne le nombre de bits de la mantisse, un majorant de l'erreur est donné par :

$$e_m = b^{-n} \quad \text{cas de l'arrondi}$$

$$e_m = 2.b^{-n} \quad \text{cas de la troncature}$$

Ce type de représentation permet une extension de la gamme dynamique, du fait de l'effet multiplicatif introduit par l'exposant. Cependant, il entraîne une complication des opérations arithmétiques et des circuits [7]. Par contre, les performances de l'arithmétique à virgule fixe sont inverses, c.à.d elle est plus rapide et demande moins de mémoire.

3.4 Effets de la longueur finie du mot

Dans les réalisations aussi bien matérielles que logicielles des filtres numériques, on est amené à stocker les nombres dans des registres de longueurs finies. Par conséquent, les coefficients doivent être quantifiés par arrondi ou troncature avant qu'ils soient stockés dans les registres. Le procédé de cette quantification entraînera donc trois types d'erreurs:

1- L'inexactitude de la réponse du filtre due à la quantification des coefficients.

2- Le bruit arrondi causé par la quantification des produits à l'intérieur de la réalisation.

3- les cycles limites dues aux effets non linéaires de la quantification [6].

En plus des erreurs citées, il existe un autre type d'erreur due à l'effet de la conversion analogique-numérique. Seulement, cette erreur concerne l'extérieur du filtre (cf. section 3.2); par conséquent, elle est indépendante de la réalisation du filtre. Les autres effets, cités précédemment, affectent la performance du filtre et dépendent essentiellement:

* du type d'arithmétique utilisé dans l'algorithme du filtre.

* du type de la quantification utilisée pour réduire les mots à la longueur désirée (arrondi ou troncature).

* de la structure choisie du filtre utilisé.

Erreurs de quantification des coefficients

Les erreurs de quantification des coefficients, introduisent des perturbations dans les pôles et zéros de la fonction de transfert, qui à leur tours introduisent des erreurs dans la réponse fréquentielle. Dans ces conditions, le filtre peut ne pas satisfaire les spécifications désirées, cet effet est appelé sensibilité du filtre à la quantification. C'est un effet déterministe qui peut être négligé devant le bruit de calcul [8]. Soient a_i et b_i les valeurs exactes (non quantifiées) des coefficients, a'_i et b'_i leur valeurs réelles après quantification, à savoir:

$$a'_i = a_i + \alpha_i$$

$$b'_i = b_i + \beta_i$$

α_i et β_i sont les erreurs introduites par la quantification. L'équation de récurrence du filtre à coefficients non quantifiés est donnée par (2.1) et celle du filtre à coefficients quantifiés est:

$$y'(k) = \sum_i b'_i\, u(k-i) - \sum_i a'_i\, y'(k-i) \qquad (3.4)$$

Ainsi, l'erreur e(k) s'écrit:

$$e(k) = y'(k) - y(k)$$

$$= \sum_i \beta_i . u(k-i) - \sum_i a_i [y'(k-i) - y(k-i)] - \sum_i \alpha_i y'(k-i)$$

il vient par suite:

$$e(k) + \sum_i a_i e(k-i) = \sum_i \beta_i u(k-i) - \sum_i \alpha_i [y(k-i) + e(k-i)] \qquad (3.5)$$

Si on pose: $$H(Z) = \frac{\sum_i b_i Z^{-i}}{\sum_i a_i Z^{-i}} = \frac{N(Z)}{D(Z)}$$

$$A(Z) = \sum_i \alpha_i Z^{-i}$$

$$B(Z) = \sum_i \beta_i Z^{-i}$$

et on prend la transformée en Z de l'équation (3.5), on obtient:

$$E(Z) = \frac{B(Z)\,D(Z) - A(Z)\,N(Z)}{D(Z)\,[1 + D(Z) + A(Z)]}\,U(Z) \qquad (3.6)$$

Où E(Z) est la transformée en Z de e(k). On pose:

$$\Delta(Z) = \frac{B(Z)\,D(Z) - A(Z)\,N(Z)}{D(Z)\,[1 + D(Z) + A(Z)]}$$

On substitue $\Delta(Z)$ dans (3.6):

$$E(Z) = Y'(Z) - Y(Z) = \Delta(Z).U(Z) \qquad (3.7)$$

Par conséquent:

$$\frac{Y'(Z)}{U(Z)} = \frac{Y(Z)}{U(Z)} + \Delta(Z)$$

$$\Rightarrow \quad H'(Z) = H(Z) + \Delta(Z) \qquad (3.8)$$

H(Z) est la fonction de transfert du filtre à coefficients non quantifiés, H'(Z) celle du filtre à coefficients quantifiés, et $\Delta(Z)$ celle du "filtre d'erreur". Ainsi le filtre à coefficients quantifiés apparait, comme la mise en parallèle de deux filtres H(Z) et $\Delta(Z)$.

Erreurs de quantification de produit

Il s'agit de l'erreur la plus délicate, celle causée par la quantification des opérations arithmétiques. A chaque instant, dans un filtre numérique, un signal représenté par b_1 bits est multiplié par un coefficient représenté 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 fixée pour tout le système, chaque sortie des multiplieurs doit être quantifiée avant la prochaine opération. Cependant, on peut accommoder l'accroissement de la longueur des mots par l'utilisation de registres plus longs (accumulateurs à précision infinie) que ceux du signal original afin de stocker les résultats d'opérations arithmétiques, Seulement en pratique, il est d'ordinaire que la taille des mots soit réduite. Si on considère le sous-bloc représenté par la figure (3.3), cette erreur additive e(k), doit figurer après chaque produit dans chaque élément multiplicateur. En général, dans la structure d'un

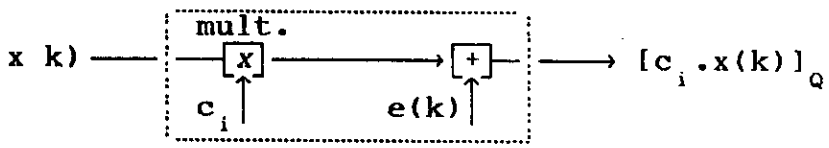


Fig 3.3: Introduction des erreurs de quantification de produit.

filtre représenté par l'implantation à virgule fixe, l'erreur arrondie ou troncaturée apparait à chaque noeud de calcul, où chaque multiplieur représenté par le modèle de la figure (3.3), peut être remplacé comme dans la figure (3.4). A savoir, chaque

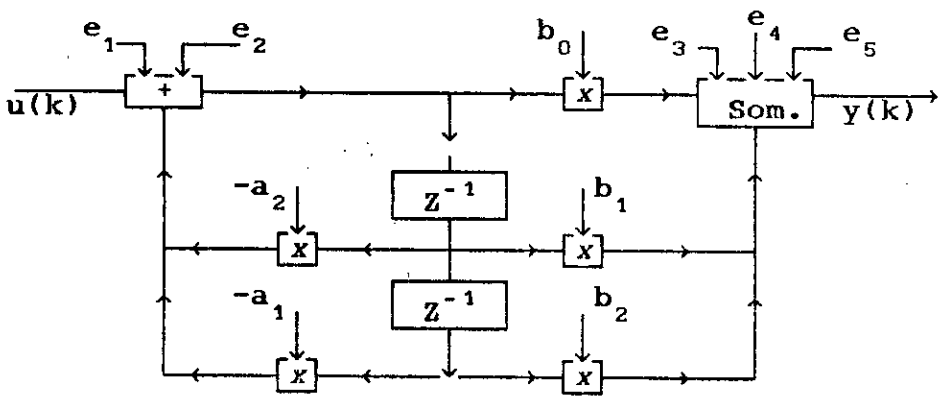


Fig 3.4: Modéle de bruit pour une section canonique de second ordre.

signal bruit $e_i(k)$ peut être représenté, dans le cas d'arrondi par exemple, comme un processus de bruit avec une densité de probabi-

lité uniforme [8],i.e.,

$$p(e_i,k) = \begin{cases} 1/q & \text{pour } -q/2 < e_i(k) < +q/2. \\ 0 & \text{ailleurs.} \end{cases} \qquad (3.9)$$

avec la considération des deux hypothèses suivantes:

1°- $e_i(k)$ et $e_i(k+n)$ sont statistiquement indépendants pour toute valeur de n ($n \neq 0$).

2°- $e_i(k)$ et $e_j(k+n)$ sont statistiquement indépendants pour toute valeur de k ou n ($i \neq j$).

Phénomènes d'oscillations

Dans l'étude du bruit de calcul, on suppose que les valeurs des échantillons du signal d'entrée du filtre numérique sont de même ordre de grandeur que les différentes multiples du pas de quantification q. Ce qui nous permet d'admettre que les échantillons du bruit de calcul sont statistiquement incorrélés aussi bien entre eux qu'avec la suite du signal d'entrée. Mais dans le cas où ces satisfactions ne sont pas obtenues, les oscillations qui peuvent apparaître sont de deux sortes:

- Oscillations à faibles amplitudes dites: " Oscillations de cycles limites ".
- Oscillations à fortes amplitudes dites: " Oscillations de dépassements ".

Oscillations de cycles limites

Si le signal à travers le filtre numérique peut atteindre des valeurs constantes ou nulles dans un certain intervalle de temps, alors les erreurs de quantification tendent à devenir fortement correlées aussi bien entre eux qu'avec le signal d'entrée et peuvent causer la déviation ou la distorsion du filtre par l'apparition d'une auto-oscillation appelée " cycle limite " pour une entrée constante ou nulle [6], [9].

Oscillations de dépassements

Si l'amplitude d'un signal interne d'une réalisation en virgule fixe excéde la gamme dynamique, un dépassement se produit. Aussi, après un dépassement interne la sortie du filtre peut , en

dépendant des pôles de celui-ci, devenir indépendante du signal d'entrée, donc distordue: Ce qui provoque des " Oscillations de dépassements " [8], [9]. Toutefois, la représentation du nombre dépassant la gamme est obtenue selon la caractéristique de dépassement choisie, parmi lesquelles on cite:

- La caractéristique de complément à 2 (fig 3.5a): Elle est périodique et a pour propriété d'annuler les dépassements intermédiaires dans un accumulateur et d'obtenir des résultats de somme qui soient dans la gamme [8], [10].

- La caractéristique de saturation (fig 3.5b): L'erreur de dépassement de cette caractéristique est inférieure à celle de complément à deux mais elle est difficile à réaliser matériellement [8], [10].

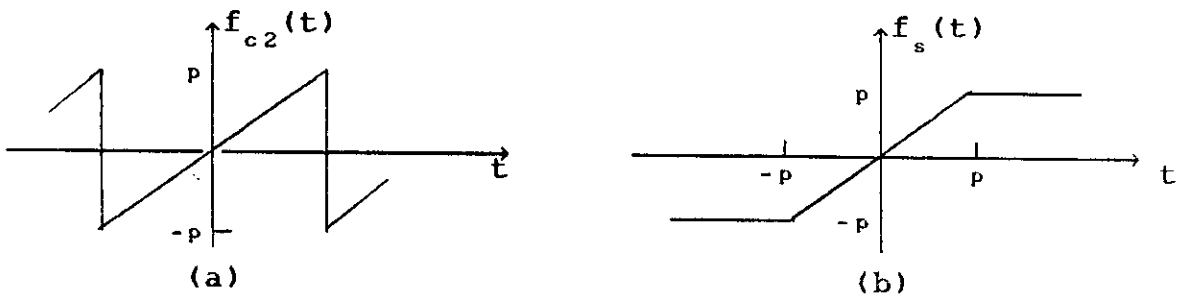


Fig 3.5: Caractéristiques de dépassements
(a) Complément à 2 et (b) Saturation

Remarques:

1- Les cycles limites peuvent être minimisées par l'utilisation de la troncature au lieu de l'arrondi [9]. Si le nombre de bits de mémoire est suffisamment grand c.à.d le pas q suffisamment petit, ces cycles peuvent être réduites ou éliminées [8].

2- les oscillations de dépassement, peuvent être minimisées en utilisant une normalisation appropriée ou en élargissant la gamme des représentations possibles, et ceci en augmentant la valeur du pas q, qui entraîne en conséquence l'augmentation de l'erreur de quantification.

3.5 Conclusion

Le présent chapitre nous a montré l'origine multiple des erreurs de mise en oeuvre d'un filtre numérique. Les conséquences sont de trois ordres: divergences, oscillations et imprécision de la sortie filtrée. Le phénomène de divergence (sensibilité des coefficients) peut être aisément maîtriser par l'analyse des pôles (vu la stabilité du filtre) après codage. La quantification du signal d'entrée ne présente pas vraiment de difficultés, l'erreur introduite par le codage pouvant être représentée par un bruit additif dont la taille dépend du pas de quantification. En revanche, l'analyse des erreurs de calcul et leur propagation s'avère fastidieuse, d'autant plus pour les filtres R.I.I vu la contre réaction qui présentent, c.à.d les valeurs calculées précédemment par le filtre servent à la détermination des valeurs suivantes, d'où l'obligation de tenir compte des erreurs de calcul ainsi propagées.

CHAPITRE IV

STRUCTURES DE REALISATIONS D'ETATS OPTIMALES

CHAPITRE IV

STRUCTURES DE RÉALISATIONS D'ÉTAT OPTIMALES

4.1 Introduction

Le problème des filtres numériques à virgule fixe avec un minimum de bruit d'arrondi revient à déterminer les structures du filtre qui minimisent l'inexactitude causée par la longueur finie du mot arithmétique. Dans l'absence de cet effet, la synthèse est triviale. Par exemple, la connaissance des coefficients du numérateur et dénominateur présente une réalisation de forme directe, cependant une telle réalisation qui est rarement utilisée, peut produire l'imprécision à laquelle l'ordre de grandeur est nettement plus grand que celui des autres réalisations. Du fait qu'en pratique, la réalisation d'un filtre numérique est basée sur des registres de longueur finie, cette limite du nombre de bits introduit une dégradation de la performance du filtre due à l'effet de la quantification des coefficients, ainsi que les différents produits rencontrés dans le calcul. On développe dans ce chapitre, un algorithme de calcul celui de S.Hwang [1], qui permet de réaliser une structure de filtre numérique récursif avec une variance de l'erreur de calcul à la sortie du filtre minimum.

4.2 Normalisation des filtres numériques en virgule fixe

Le but de la règle de normalisation est de limiter la probabilité de dépassement dans les registres internes du filtre. Normaliser revient donc à mettre toutes les valeurs numériques des variables internes dans une gamme appropriée à la réalisation matérielle. Dans la représentation à virgule fixe, les nombres, une fois normalisés, sont inférieurs à l'unité. Ceci signifie que le point décimal est positionné après le 1^{er} bit (le bit le plus significatif). Dans un tel cas, on essaie de limiter tous les nombres possibles qu'on peut représenter par la gamme dynamique.

Dans l'étude qui suit, on opte pour la norme L_2 [8], elle permet de conserver moyennement la gamme dynamique (i.e diminuer la probabilité d'overflow). Elle est définit par:

$$\delta \quad \|f\|_2 = \delta \left[\sum_{k=0}^{\infty} f^2(k)\right]^{1/2} = 1 \qquad (4.1)$$

où f(k) représente la réponse impulsionnelle entre l'entrée u(k) et la variable d'état x(k), et δ un paramètre de normalisation choisi subjectivement afin d'obtenir une bonne représentation des valeurs dans la gamme souhaitée, d'éviter ainsi les oscillations de dépassements et de réduire les erreurs d'arrondi. En général δ=1, mais les résultats de simulation ont montré que δ=4 [7] [8], représente un compromis optimal entre les erreurs de calcul et les erreurs de dépassement. Si on augmente δ, on fait diminuer la probabilité de dépassement mais on augmente les erreurs de calcul, car l'augmentation de δ introduit une mauvaise correspondance entre la gamme dynamique et les nombres à représenter. Etant donnée la norme L_2, on définit la mesure suivante:

$$\lambda_i^2 = \left[\sum_{k=0}^{\infty} f_i^2(k)\right] \qquad (4.2)$$

et par l'égalité de Parseval, on obtient:

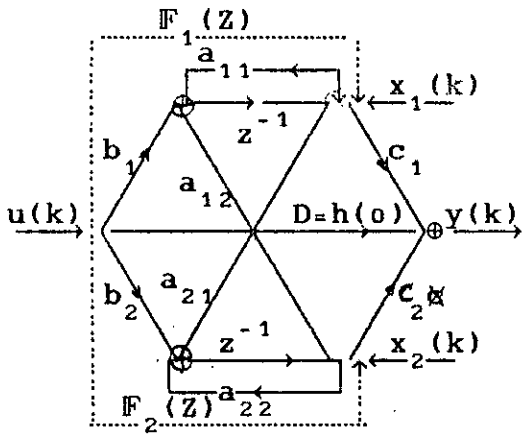
$$\lambda_i^2 = \| f \|_2^2 = \frac{1}{2\pi j} \oint_{|z|=1} F_i(Z)\, F_i(Z^{-1})\, Z^{-1}\, dz$$

$$= \frac{1}{2\pi} \int_{-\pi}^{\pi} | F_i(e^{j\Phi}) |^2 \, d\Phi$$

où $f_i(k)$ est la réponse impulsionnelle du $i^{ème}$ mode d'état (variable d'état) et $F_i(Z)$ est la fonction de transfert correspondante entre l'entrée u(k) et le $i^{ème}$ mode d'état $x_i(k)$ (Fig 4.1).
Si l'entrée u(k) du filtre est un processus de bruit blanc avec une moyenne nulle et une variance σ_u^2, la $i^{ème}$ variable d'état est aussi aléatoire avec une moyenne nulle et une variance σ_i^2 donnée par:

$$\sigma_i^2 = \sigma_u^2 \sum_{k=0}^{\infty} f_i^2(k) = \sigma_u^2 \| f_i \|_2^2 = \sigma_u^2 \lambda_i^2 \qquad (4.3)$$

où λ_i^2 n'est autre que la mesure définie par l'équation (4.2). Par conséquent on définit le vecteur réponse impulsionnelle f(k) [Nx1] entrée-variable d'état par:



$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

$$B = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

$$C = \begin{bmatrix} c_1 & c_2 \end{bmatrix}$$

Fig 4.1 Structure d'espace d'état d'un filtre de second ordre.

$$f^T(k) = \begin{bmatrix} f_1(k) & f_2(k) & \cdots & f_N(k) \end{bmatrix} \qquad (4.4)$$

f^T désigne la transposée de f et $f_i(k)$ i = 1...N, sont les réponses impulsionnelles [7]. Pour une réalisation à variable d'état, en termes de (A,B,C,D), on exprime le vecteur d'état par [8]:

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

et la réponse impulsionnelle séquentielle par:

$$f_i(k) = \begin{cases} 0 & k \leq 0 \\ (A^{k-1}B)_i & k > 0 \end{cases} \qquad (4.5)$$

Ainsi, la fonction de transfert correspondante est donnée par:

$$F_i(Z) = [\ (\ ZI - A\ )^{-1}B\ ]_i \qquad (4.6)$$

A partir du vecteur f(k) (4.4), on construit une matrice K [NxN] de la façon suivante:

$$K = \sum_{k=0}^{\infty} f(k)\ f^T(k) \qquad (4.7)$$

Par développement de (4.7), on obtient:

$$K = \sum_{k=0}^{\infty} \begin{bmatrix} f_1^2(k) & f_1(k)f_2(k) & \cdots & f_1(k)f_N(k) \\ f_2(k)f_1(k) & f_2^2(k) & \cdots & f_2(k)f_N(k) \\ \vdots & & & \\ f_N(k)f_1(k) & f_N(k)f_2(k) & \cdots & f_N^2(k) \end{bmatrix}$$

On remarque que les éléments diagonaux de la matrice K sont ceux de la mesure λ_i^2. Par conséquent, avec une normalisation de type (4.1b), les éléments diagonaux

$$K_{ii} = \sum_{k=0}^{\infty} f_i^2(k) = \| f_i \|_2^2$$

sont tous égaux à $1/\delta^2$. Dans le cas où $\delta = 1$, $K_{ii} = 1$ $(i=1,\ldots,N)$. Par conséquent, pour normaliser un filtre, il faut lui appliquer une transformation d'état T de normalisation. Vu la nouvelle représentation d'état donnée par (2.15), celle du vecteur réponse sera:

$$f'(k) = T^{-1} f(k) \qquad (4.8)$$

ce qui induit celle de la matrice K:

$$K' = T^{-1} K T^{-T} \qquad (4.9)$$

avec une contrainte de normalisation:

$$\delta^2 K'_{ii} = 1 \qquad (4.10)$$

Il suffit donc de prendre comme matrice de normalisation T satisfaisant à la contrainte (4.10), celle donnée par:

$$T = \text{Diag} \left(\delta\sqrt{K_{11}}, \delta\sqrt{K_{22}}, \ldots, \delta\sqrt{K_{NN}} \right) \qquad (4.11)$$

Diag(.) indique la matrice diagonale. Il vient par suite:

$$\begin{cases} K'_{ij} = \dfrac{K_{ij}}{T_{ii}\, T_{jj}} \\ K'_{ii} = 1 / \delta^2 \end{cases} \qquad (4.12)$$

Remarque:

L'équation (4.7) montre que la matrice K est définie positive symétrique, peut être interprétée comme matrice de covariance du vecteur d'état du filtre, pour une entrée de bruit blanc centré normalisé, c.-à-d.:

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

En fonction des paramètres (A,B), elle s'exprime comme suit:

$$K = \sum_{k=0}^{\infty} (A^k B)\, (A^k B)^T \qquad (4.13)$$

Par conséquent, elle vérifie l'équation de Lyapunov [8]:

$$K = A\, K\, A^T + B\, B^T \qquad (4.14)$$

La résolution de cette équation nous permet de déterminer la matrice K pour un filtre donné. Le calcul de K est résumé dans l'organigramme présenté par la figure (Fig 5.7).

4.3 Analyse de bruit de calcul dans le filtre numérique RII

Après avoir posé le problème des erreurs de calcul dans un filtre numérique dans le chapitre précédent, on établira dans cette section l'expression générale de la variance de bruit de calcul à la sortie d'un filtre numérique récursif. En présence d'un signal d'entrée, c.-à-d., pour des valeurs non nulles de u(k), l'opération d'arrondi ou de troncature avant mise en mémoire avec un pas de quantification q, est équivalente à la superposition au signal d'entrée un signal d'erreur qui, appliqué à l'entrée du filtre, subit la fonction filtrage, dont la variance à la sortie du filtre s'exprime par:

$$\sigma_y^2 = \sigma_e^2 \sum_{k=0}^{\infty} h^2(k) \qquad (4.15)$$

Aussi, d'autres arrondis interviennent dans les multiplications, donc il apparait clairement que les signaux d'erreurs produits sont à ajouter à chaque noeud de calcul dans le filtre (cf. section 3.4), et au signal de sortie de ce filtre, comme le montre la figure ci-dessous (Fig 4.2). On suppose dans cette analyse, le modèle de bruit arrondi remplit les deux hypothèses suivantes:

* Les sources d'erreurs de différents accumulateurs sont incorrélées.

* Chaque source supposée blanche est incorrélée avec l'entrée.

En ce qui concerne les erreurs de dépassements, celles-ci sont très faibles après une normalisation appropriée des registres internes du filtre; elles n'apparaissent pas dans l'expression du bruit de calcul. Par conséquent pour déterminer le bruit de calcul à la sortie du filtre, il est nécessaire d'identifier la fonction de transfert variable d'état-sortie G(Z) (i=1,...,N) (Fig 4.2). Comme pour la construction de $F_i(Z)$ en termes de (A,B), $G_i(Z)$ s'exprime en termes de (A,C) comme suit:

$$G_i(Z) = \left(C(ZI - A)^{-1} \right)_i \qquad (4.16)$$

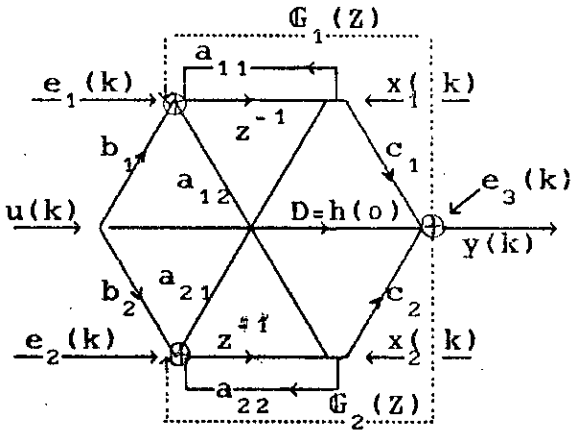


Fig4.2:Erreurs arrondis dans une structured'espace d'état de second ordre.

ainsi, la réponse impulsionnelle correspondante est donnée par:

$$g_i(k) = \begin{cases} 0 & k \leq 0 \\ (C\ A^{k-1})_i & k > 0 \end{cases} \qquad (4.17)$$

On remarque sur la figure (4.2), qu'il existe trois sources de bruit arrondi dans le filtre. Elles occupent les noeuds 1, 2 et 3 (registre de sortie), dont chacun subit la contribution de trois sources de bruit. Chaque produit est remplacé donc par son modèle de bruit équivalent, par conséquent la variance de bruit de sortie totale due aux quantifications des produits est obtenue par la sommation des variances des bruits de sortie associés avec toutes les sources de bruit internes.

Pour le noeud 1, par exemple, la variance de bruit de sortie due aux trois sources de bruit blanc est donnée par:

$$\sigma_1^2 = \sigma_{e_{11}}^2 \sum_{k=0}^{\infty} g_1^2(k) + \sigma_{e_{12}}^2 \sum_{k=0}^{\infty} g_1^2(k) + \sigma_{e_{13}}^2 \sum_{k=0}^{\infty} g_1^2(k)$$

$$= 3\ \sigma_{e_1}^2 \sum_{k=0}^{\infty} g_1^2(k) = 3\ \sigma_{e_1}^2 \parallel g_1 \parallel_2^2$$

De façon similaire, le bruit causé par la quantification du noeud 2 donne:

$$\sigma_2^2 = 3\ \sigma_{e_2}^2 \parallel g_2 \parallel_2^2$$

et celui, du noeud de sortie 3 donne:

$$\sigma_3^2 = 3\ \sigma_{e_3}^2$$

La variance du bruit de sortie arrondi est donc, donné par la

somme des trois supposés incorrélés:

$$\sigma^2_{bruit} = 3\,\sigma^2_e \left(1 + \| g_1 \|^2_2 + \| g_2 \|^2_2 \right)$$

où $\sigma^2_e = \sigma^2_{e_1} = \sigma^2_{e_2} = \sigma^2_{e_3}$ représentent la variance de bruit de calcul aux noeuds 1, 2 et 3, respectivement.

En général, une structure de variable d'état d'ordre N contient N+1 noeuds qui génèrent des bruits arrondis ou troncaturés. De ce fait, diverses multiplications sont effectuées pour le calcul de la $i^{ème}$ variable d'état, donc on a divers arrondis incurés dans le même calcul. Si ν_i est le nombre de ses arrondis, et si chacun est supposé être non correlé avec les autres, alors le calcul de la $i^{éme}$ variable d'état produit un bruit de sortie arrondi donné par:

$$\sigma^2_i = \nu_i \sigma^2_e \sum_{k=0}^{\infty} g^2_i(k) = \nu_i \sigma^2_e \| g_i \|^2_2 \qquad (4.18a)$$

avec

$$\sigma^2_e = E\left[e^2_i(k) \right] = q^2/12 \qquad (4.18b)$$

Si les arrondis sont aussi non correlés d'un noeud à un noeud à l'intérieur du filtre, alors le bruit arrondi à la sortie du filtre est donné par:

$$\sigma^2_y = \sigma^2_e \left[\nu_0 + \sum_{i=1}^{N} \nu_i \left(\sum_{k=0}^{\infty} g^2_i(k) \right) \right] + \sigma^2_e \sum_{k=0}^{\infty} h^2(k) \qquad (4.19)$$

Le terme $\sigma^2_e \sum_{k=0}^{\infty} h^2(k)$, généralement faible, est dû à la conversion analogique-numérique de l'entrée (4.15). ν_0 désigne le nombre d'arrondis pour le calcul de la sortie du filtre. Si on utilise un accumulateur à double précision, c.-à-d. l'arrondi s'effectue après l'addition de toutes les branches qui aboutissent au noeud correspondant, on aura par conséquent, $\nu_i = 1$ si non $\nu_i = N+1$; c'est le cas d'accumulateur à simple précision, où on suppose que la matrice A est complète (le cas le plus défavorable). A partir de l'équation (4.19), le rapport σ^2_y/σ^2_e donne:

$$G' = \nu_0 + \sum_{i=1}^{N} \nu_i \| g_i \|^2_2 \qquad (4.20)$$

où le terme de la quantification de l'entrée est négligé. G' est appelé le gain de bruit de calcul. Il est clair, que le terme $\sum \nu_i \| g_i \|^2_2$ soit aussi faible que possible, afin que σ^2_y donnée par

(4.19) le soit aussi, ainsi avoir une structure à faible bruit de calcul. De ce fait, on construit une matrice W de la manière suivante:

$$W = \sum_{k=0}^{\infty} g(k)\ g^T(k) \tag{4.21}$$

avec

$$g^T(k) = \begin{bmatrix} g_1(k) & g_2(k) & \dots & g_N(k) \end{bmatrix} \tag{4.22}$$

qui représente le vecteur réponse impulsionnelle variable d'état-sortie de dimension [1xN], par conséquent W s'écrit:

$$W = \sum_{k=0}^{\infty} \begin{bmatrix} g_1^2(k) & g_1(k)g_2(k) & \dots & g_1(k)g_N(k) \\ g_2(k)g_1(k) & g_2^2(k) & \dots & g_2(k)g_N(k) \\ \vdots & & & \\ g_N(k)g_1(k) & g_N(k)g_2(k) & \dots & g_N^2(k) \end{bmatrix}$$

Il est évident que les éléments diagonaux de W apparaissent dans l'expression du gain (4.20). Par substitution on obtient:

$$G' = \nu_0 + \sum_{i=1}^{N} \nu_i\ W_{ii} \tag{4.23}$$

autrement dit, pour tout i= 1,...,N on a dans le cas le plus défavorable:

$$G'_{(\text{simple précision})} = (N+1)\ \big(1 + Tr(W)\big) \tag{4.24a}$$

et dans le cas le plus favorable:

$$G'_{(\text{double précision})} = 1 + Tr(W) \tag{4.24b}$$

Tr(.) désigne la trace de la matrice.

Si on applique une transformation d'état T non singulière au vecteur d'état (2.13), alors le vecteur g(k) (4.22) sera transformée de la façon suivante:

$$g'(k) = T^T\ g(k) \tag{4.25}$$

la matrice W devient [8]:

$$W' = T^T\ W\ T \tag{4.26}$$

En particulier, si T est la transformation de normalisation définie par (4.11), W' aura pour composantes:

$$W'_{ij} = \delta^2 W_{ij} \sqrt{K_{ii} K_{jj}}$$

Par conséquent, après normalisation l'expression globale du gain de bruit de calcul en fonction des paramètres du filtre non normalisé est donnée par:

$$G' = \nu_0 + \delta^2 \sum_{i=1}^{N} \nu_i K_{ii} W_{ii} \qquad (4.27)$$

Remarque

L'équation (4.21) montre que la matrice W est définie positive symétrique, peut être interprétée comme matrice de l'élément bruit du filtre numérique, en fonction des paramètres (A,C), elle s'exprime comme suit:

$$W = \sum_{k=0}^{\infty} (CA^k)^T (CA^k) \qquad (4.28)$$

Par conséquent, elle vérifie l'équation de Lyapunov [8]:

$$W = A^T W A + C^T C \qquad (4.29)$$

La résolution de cette équation nous permet de déterminer la matrice W pour un filtre donné. Le calcul de W est résumé dans un organigramme présenté à la figure (fig 5.7) en remplaçant K par W, A par A^T et B par C^T.

4.4 Sensibilité du filtre

La limite 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'intérieur du cercle unité. De ce fait, si la fonction de transfert H(Z) du filtre numérique a été calculée d'abord et que la limitation des nombres de coefficients intervient ensuite, H(Z) se trouve modifiée par l'introduction des polynômes parasites $e_{num}(Z)$ et $e_{den}(Z)$ à savoir:

$$H'(Z) = \frac{N(Z) + e_{num}(Z)}{D(Z) + e_{den}(Z)}$$

Il est donc important dans des considérations pratiques de synthétiser des filtres dont la fonction de transfert présente une faible sensibilité aux variations de ses coefficients dues à la limite de leur représentation arithmétique. Pour évaluer ainsi la sensibilité de H(z) dans l'espace d'état, on exprime les dérivées partielles de H(Z) par rapport à chaque élément des paramètres d'état (A,B,C,D) du filtre. On obtient le système suivant:

$$\begin{cases} \dfrac{\partial H(Z)}{\partial b_i} = \left[C(ZI - A)^{-1} \right]_i = G_i(Z) \\ \dfrac{\partial H(Z)}{\partial c_i} = \left[(ZI - A)^{-1} B \right]_i = F_i(Z) \\ \dfrac{\partial H(Z)}{\partial a_{ij}} = G_i(Z)\, F_j(Z) \end{cases} \tag{4.30}$$

On définit ainsi, les différentes sensibilités indexées ψ_i^b, ψ_i^c, et ψ_{ij}^a à partir des équations (4.7) et (4.21), de la manière suivante [11]:

$$\begin{cases} \psi_i^b = \| G_i \|_2^2 \\ \psi_i^c = \| F_i \|_2^2 \\ \psi_{ij}^a = \| G_i F_j \|_2^2 \le \| G_i \|_2^2 \, \| F_j \|_2^2 \end{cases} \tag{4.31}$$

en fonction des paramètres K et W, le système (4.31) devient:

$$\begin{cases} \psi_i^b = W_{ii} \\ \psi_i^c = K_{ii} \\ \psi_{ij}^a \le W_{ii} K_{jj} \end{cases} \tag{4.32}$$

à partir du système (4.32), on définit la mesure de sensibilité de H(Z) par rapport à chaque paramètre (A,B,C,D) comme suit:

$$\begin{cases} S_b = \sum_{i=1}^{N} \psi_i^b = Tr(W) \\ S_c = \sum_{i=1}^{N} \psi_i^c = Tr(K) \\ S_a = \sum_{i=1}^{N} \sum_{i=1}^{N} \psi_{ij}^a \leq \sum_{i=1}^{N} \sum_{i=1}^{N} W_{ii} K_{jj} = Tr(W)\ Tr(K) \end{cases} \qquad (4.33)$$

Par conséquent, la mesure de sensibilité globale s'exprime par la somme suivante [6]:

$$\begin{aligned} M &= S'_a + S_b + S_c \\ &= Tr(K)\ Tr(W) + Tr(K) + Tr(W) \end{aligned} \qquad (4.34)$$

où S'_a est la borne inférieure de S_a. Si on applique une transformation de normalisation telle que:

$$Tr(K) = N \qquad (K_{ii} = 1\ ,i = 1,\ldots,N)$$

l'équation (4.34) devient alors:

$$M = (N+1)\ Tr(W) + N \qquad (4.35)$$

On remarque, le terme Tr(W) apparait à la fois dans l'expression de la mesure de sensibilité et celle du gain de bruit de calcul. Par conséquent, des structures qui minimisent le bruit d'arrondi, minimisent aussi la sensibilité du filtre.

4.5 Structures avec minimum d'erreur de calcul

Minimiser le gain de bruit de calcul d'un filtre numérique décrit par ses paramètres d'état (A,B,C,D) et les matrices (K,W), revient à trouver une transformation d'état T non singulière, qui donne la nouvelle structure ($T^{-1}AT$,$T^{-1}B$,CT,D) à faible bruit sous la contrainte de normalisation.

Le procédé de minimisation de S. Hwang est basé sur la décomposition polaire de la matrice T, qui reste à déterminer.

Borne inférieure du gain de bruit

Soit T une matrice arbitraire réelle non singulière, alors il existe une matrice unique orthogonale R, et une matrice définie positive symétrique S telles que la matrice T s'écrit par décom-

position polaire, comme suit:

$$\begin{aligned} T &= R\ S \\ &= R\ (R_0\ P\ R_0^T) \qquad \text{(par diagonalisation de S)} \\ &= R_1\ P\ R_0^T \end{aligned}$$

où R_0 et R_1 sont des matrices orthogonales et P une matrice diagonale donnée par:

$$P = \text{Diag}\left[\ \lambda_i\ \right] \qquad i = 1,\ldots,N$$

Etant donné les matrices K_0 et W_0 du système original (A,B,C,D), en appliquant la transformation T à la matrice W_0 , l'équation du gain à minimiser devient:

$$\begin{aligned} G &= G' - 1 \\ &= \text{Tr}(T^T\ W_0 T) \end{aligned} \qquad (4.36)$$

On substitue T , dans (4.36) on obtient:

$$\begin{aligned} G &= \text{Tr}(T\ T^T\ W_0) \\ &= \text{Tr}(P^2\ R_1^T\ W_0\ R_1) \\ &= \sum_{i=1}^{N} \lambda_i^2\ \mu_i^2 \end{aligned} \qquad (4.37)$$

où μ_i^2 est le $i^{ème}$ élément diagonal de $(R_1^T W_0 R_1)$. De même T substituée dans (4.9) donne la contrainte de normalisation comme suit:

$$K = R_0\ P^{-1}\ R_1^T\ K_0\ R_1\ P^{-1}\ R_0^T = \begin{bmatrix} 1 & & X \\ & \ddots & \\ X & & 1 \end{bmatrix} \qquad (4.38)$$

Par ailleurs, on obtient [18]:

$$\begin{cases} \prod_{i=1}^{N} \mu_i^2 \geq \text{Det}(W_0) \\ \prod_{i=1}^{N} \lambda_i^2 \geq \text{Det}(K_0) \end{cases} \qquad (4.39)$$

où Det(.) désigne le déterminant de la matrice. On applique l'inégalité de la moyenne arithmétique-géométrique [18] à (4.37), il vient par suite de calcul:

$$\text{Tr}(\ T^T\ W_0\ T) \geq N \prod_{i=1}^{N} (\ \lambda_i^2\ \mu_i^2\ ) = N \left[\prod_{i=1}^{N} \lambda_i^2 \prod_{i=1}^{N} \mu_i^2\right]^{1/N}$$

ce qui implique:

$$Tr(\ T^T\ W_0\ T\ ) \geq N\ \left[\ Det(K_0)\ Det(W_0)\ \right]^{1/N} = N\ \left[\ Det(K_0W_0)\ \right]^{1/N}$$

On note par:

$$BI = N\ \left[\ Det(K_0W_0)\ \right]^{1/N} \qquad (4.40)$$

la borne inférieure du gain de bruit de calcul. Elle est atteinte si et seulement si [1]:

- K_0W_0 est symétrique.
- K_0^{-1} et W_0 sont équivalentes à une constante près.

Cette borne inférieure est exprimée en termes du produit des valeurs propres de K_0W_0, donc il en est de même pour le minimum absolu (4.24b). Le produit KW, des matrices K_0 et W_0 après application de la transformation d'état T, donne:

$$KW = (T^{-1}K_0T^{-T})\ (T^TW_0T)$$
$$= T^{-1}\ K_0W_0\ T$$

Ce qui montre que les valeurs propres sont invariantes sous une transformation d'état, et elles sont déterminées uniquement par la fonction de transfert du filtre. Ce sont des constantes universelles pour toutes les réalisations d'espace d'état d'un filtre donné. On note que, cette borne inférieure n'est atteinte que pour certains filtres très particuliers comme le filtre pass-tout [1].

Gain de bruit minimum

Une première application de la transformation T_0 aura pour but de réduire la matrice K_0 à une matrice identité, i.e.,

$$K_1 = T_0^{-1}\ K_0\ T_0^{-T} = I$$

$$\Rightarrow\ K_0 = T_0T_0^T \qquad (4.41)$$

donc on peut obtenir la matrice T_0 en effectuant la factorisation de Cholesky pour la matrice K_0 (cf. Annexe A). Par suite, on a:

$$W_1 = T_0^T\ W_0\ T_0$$

Sans perte de généralité, il est plus simple de considérer la minimisation la minimisation du gain de bruit dans le système K_1 et W_1, respectivement, ainsi le problème devient:

Minimiser

$$Tr(T^T\ W_1\ T) = Tr(P^2\ R_1^T\ W_1\ R_1)$$
$$= \sum_{i=1}^{N} \lambda_i^2\ \mu_i^2 \qquad (4.42a)$$

soumis à (4.38), qui devient

$$K_1 = R_0\, P^{-2}\, R_0^T = \begin{bmatrix} 1 & & X \\ & \ddots & \\ X & & 1 \end{bmatrix} \qquad (4.42b)$$

On considère respectivement, la trace et le déterminant de (4.42b), il est remarquable que les éléments λ_i de la matrice P sont contrariés par:

$$\sum_{i=1}^{N} 1/\lambda_i^2 = N \qquad (4.43a)$$

et

$$\prod_{i=1}^{N} \lambda_i^2 \geq 1 \qquad (4.43b)$$

Cependant, (4.43a) $\Rightarrow$ (4.43b) par l'inégalité de la moyenne arithmétique-géommetrique, donc (4.43a) sera une contrainte nécessaire et suffisante. Ainsi, Pour minimiser (4.42a), où les λ_i sont soumis à (4.43a), on applique la méthode de Lagrange, définie par la fonction suivante:

$$F(\lambda,\alpha) = \sum_{i=1}^{N} \lambda_i^2 \mu_i^2 + \alpha \left[\sum_{i=1}^{N} 1/\lambda_i^2 - N \right] \qquad (4.44)$$

où α est le multiplicateur lagrangien. Par conséquent on a:

$$\frac{\partial F}{\partial \lambda_i} = 0 \Rightarrow \lambda_{ioptm}^2 = \frac{\sqrt{\alpha}}{\mu_i} \qquad \mu_i > 0 \text{ pour tout } i \qquad (4.45a)$$

$$\frac{\partial F}{\partial \alpha_i} = 0 \Rightarrow \sqrt{\alpha_{ioptm}} = \frac{1}{N} \sum_{i=1}^{N} \mu_i \qquad (4.45b)$$

en substituant (4.45a) dans (4.44), on obtient:

$$F(\lambda_{ioptm},\alpha) = 2\sqrt{\alpha} \sum_{i=1}^{N} \mu_i - \alpha N$$

d'où

$$\Delta F = F(\lambda,\alpha) - F(\lambda_{ioptm},\alpha)$$

$$= \sum_{i=1}^{N} \left(\lambda_i \mu_i - \frac{\sqrt{\alpha}}{\lambda_i} \right)^2$$

$\Delta F \geq 0$ montre que (4.45a) est l'unique minimum global de (4.42a). Par suite, substituons (4.45a) et (4.45b) dans (4.42a), on obtient:

$$\text{Min } \mathrm{Tr}(T^T W_1 T) = \frac{1}{N} \left(\sum_{i=1}^{N} \mu_i \right)^2 \qquad (4.46)$$

où μ_i est la racine carrée du $i^{ème}$ élément diagonal de la matrice $(R_1^T W_1 R_1)$. Si on désigne par $\phi_1^2,\ldots,\phi_N^2$ les valeurs propres de $K_0 W_0$, par invariance, sont aussi ceux de W_1 ($K_1 = I$), alors en considérant la trace et le déterminant de la matrice définie positive $(R_1^T W_1 R_1)$, on obtient:

$$\sum_{i=1}^{N} \mu_i^2 = \sum_{i=1}^{N} \phi_i^2 = Tr(W_1) \qquad (4.47a)$$

et

$$\prod_{i=1}^{N} \mu_i^2 \geq \prod_{i=1}^{N} \phi_i^2 = Det(K_1 W_1) \qquad (4.47b)$$

La somme (4.46) est donc, soumis à ces contraintes (4.47a) et (4.47b). Par conséquent, si $\mu_1^2,\ldots,\mu_N^2$ et $\phi_1^2,\ldots,\phi_N^2$ sont les éléments diagonaux et les valeurs propres, respectivement, de la matrice définie positive symétrique $(R_1^T W_1 R_1)$, alors [18] :

$$\sum_{i=1}^{N} \mu_i \geq \sum_{i=1}^{N} \phi_i \qquad \mu_i > 0 \text{ et } \phi_i > 0 \text{ pour tout } i$$

et l'égalité aura lieu si et seulement si la matrice est diagonale. Ce qui implique, que la condition $\mu_i = \phi_i$ pour tout i, est la solution unique du minimum global de (4.46), ce dernier est atteint si et seulement si la matrice orthogonale R_1 est la matrice de diagonalisation de W_1. Par conséquent, l'expression du gain de bruit minimum, pour une réalisation d'espace d'état d'un filtre numérique d'ordre N, soumis à la contrainte de normalisation, est donné par:

$$G_{optm} = \frac{1}{N} \left(\sum_{i=1}^{N} \phi_i \right)^2 \qquad (4.48)$$

où les ϕ_i (i=1,...,N) sont les racines carrées positives des valeurs propres invariantes de la matrice KW. Elles sont appelées modes du second ordre, et sont invariantes aussi à une transformation fréquentielle [12].

Transformation d'état d'optimisation

la structure du filtre numérique à bruit minimum, peut être obtenue à partir d'une réalisation d'espace d'état arbitraire initiale (A,B,C,D), en utilisant une transformation d'état T non singulière donnée par:

$$T = T_0 \, R_1 \, P_{optm} \, R_0^T \qquad (4.49)$$

avec

$$P_{optm} = \text{Diag}(\lambda_{1optm} \cdots \lambda_{Noptm})$$

$$\text{où } \lambda_{ioptm} = \left[\sum_{m=1}^{N} \phi_m \; / \; N \; \phi_i \right]^{1/2}$$

R_1 matrice orthogonale représente la matrice des vecteurs propres de $W = T_0^T W_0 T_0$, telle que:

$$R_1^T \; T_0^T \; W_0 \; T_0 \; R_1 = \begin{bmatrix} \phi_1^2 & 0 \\ 0 & \phi_N^2 \end{bmatrix}$$

R_0 est une matrice orthogonale et la méthode de son calcul est donnée dans l'annexe B, et T_0 donnée par (4.41) dont le calcul est effectué par le procédé de Cholesky [18] (Annexe A).

Exemple numérique

Dans le but de clarifier les étapes de calcul de la réalisation d'une structure d'état optimale, on considère l'exemple de la fonction de transfert d'un filtre passe-bas de Butterworth, d'ordre 4 avec une fréquence de coupure de $\pi/20$. Soit:

$$H(Z) = \frac{0.00003 \; (Z+1)^4}{Z^4 - 3.58973 \; Z^3 + 4.85127 \; Z^2 - 2.92405 \; Z + 0.66301}$$

les éléments A, B, C et D de la structure canonique sont:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -.66301 & 2.92405 & -4.85127 & 3.58973 \end{bmatrix} \; ; \; B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$C = \begin{bmatrix} .00001 & .00021 & .00003 & .00023 \end{bmatrix} \qquad ; \; D = .00003$$

En utilisant les équations (4.13) et (4.28), on obtient la matrice de covariance d'état K_0 et celle de l'élément bruit W_0:

$$K_0 = \begin{bmatrix} 207396.55794 & 206327.35472 & 203146.30016 & 197931.49460 \\ 206327.35472 & 207396.55794 & 206327.35472 & 203146.30016 \\ 203146.30016 & 206327.35472 & 207396.55794 & 206327.35472 \\ 197931.49460 & 203146.30016 & 206327.35472 & 207396.55794 \end{bmatrix}$$

$$W_0 = \begin{bmatrix} .02254 & -.07699 & .08812 & -.03382 \\ -.07699 & .26318 & -.30151 & .11587 \\ .08812 & -.30151 & .34581 & -.13307 \\ -.03382 & .11587 & -.13307 & .05128 \end{bmatrix}$$

Pour une telle structure, le gain de bruit est donnée par:

$$G_{can} = 141615.93999$$

et la mesure de sensibilité correspondante par:

$$M_{can} = 708083.69995$$

avec un choix de facteur de normalisation $\delta=1$, l'application de la transformation d'état donnée par (4.11) aux matrices K_0 et W_0 donne:

$$K_{nor} = \begin{bmatrix} 1 & .99484 & .97950 & .95436 \\ .99484 & 1 & .99484 & .97950 \\ .97950 & .99484 & 1 & .99484 \\ .95436 & .97950 & .99484 & 1 \end{bmatrix}$$

$$W_{nor} = \begin{bmatrix} 4675.39231 & -15968.09196 & 18276.02542 & -7015.92461 \\ -15968.09196 & 54584.00536 & -62534.03902 & 24032.83031 \\ 18276.02542 & -62534.03902 & 71720.56853 & -27598.39380 \\ -7015.92461 & 24032.83031 & -27598.39380 & 10635.97378 \end{bmatrix}$$

Les racines carrées des valeurs propres de la matrice produit K_0W_0 sont celles de la matrice $K_{nor}W_{nor}$; elles sont données par:

$$\phi_1 = 0.86593 \qquad \phi_2 = 0.48296$$
$$\phi_3 = 0.12940 \qquad \phi_4 = 0.01238$$

Pour trouver la transformation d'état T qui permet de donner un gain minimum, on doit calculer les différentes matrices suivantes:

* Matrice T_0 donnée par la factorisation de Cholesky:

$$T_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ .99484 & .10141 & 0 & 0 \\ .97950 & .20104 & .01221 & 0 \\ .95436 & .29645 & .03598 & .00293 \end{bmatrix}$$

* Matrice R_1 donnée par les vecteurs propres de $T_0^T W_0 T_0$:

$$R_1 = \begin{bmatrix} .63149 & -.87204 & .42222 & -.05154 \\ .74564 & .13142 & -.74065 & .20187 \\ .50129 & .62624 & .22733 & -.47441 \\ .27149 & .51109 & .63232 & .44143 \end{bmatrix}$$

* Matrice R_0 donnée par application des rotations successives de la forme (A.8) à la matrice $\left(P_{optm}\right)^{-1}$:

$$R_0 = \begin{bmatrix} .57470 & 0 & .81836 & 0 \\ 0 & .87500 & 0 & .48411 \\ -.57866 & -.34232 & .40637 & .61872 \\ .57866 & -.34232 & -.40637 & .61872 \end{bmatrix}$$

avec $P_{optm} = \text{Diag}\left[.65602 \quad .87843 \quad 1.69699 \quad 5.48583 \right]$

Par conséquent, la matrice T est donnée par:

$$T = \begin{bmatrix} .82445 & -.80717 & .13871 & .03581 \\ .74440 & -.73840 & .12297 & .18161 \\ .66345 & -.67205 & .08370 & .30292 \\ .58627 & -.60488 & .02744 & .40038 \end{bmatrix}$$

A partir de l'équation (4.48), on calcule le gain optimal:

$$G_{optm} = 0.55554$$

aussi la mesure de sensibilité minimum correspondante sera:

$$M_{optm} = 6.77770$$

ainsi, la structure d'état optimale du filtre considéré est donnée par:

$$A_{optm} = \begin{bmatrix} .88788 & -.01006 & .01671 & .07666 \\ -.03961 & .88046 & .01119 & -.09579 \\ -.12509 & -.14477 & .89589 & .05720 \\ -.06254 & .01859 & -.16872 & .92549 \end{bmatrix} ; \quad B_{optm} = \begin{bmatrix} .33284 \\ .32924 \\ -.06763 \\ .02014 \end{bmatrix}$$

$$C_{optm} = \begin{bmatrix} .15142 & -.15290 & .01710 & .06624 \end{bmatrix} ; \quad D_{optm} = .00003$$

On note que le programme qui synthétise la structure optimale est donné dans l'annexe C.

4.6 Performances d'une structure optimale

Dans cette section, on développe les performances de la structure optimale vis-à-vis de la structure canonique d'un filtre numérique récursif au moyen de simulation sur ordinateur avec la représentation à virgule fixe.

Gain de bruit de calcul

Afin de mettre en évidence l'effet de la transformation fréquentielle sur les performances de chaque type de structure, on étudie la variation du gain de bruit de calcul en fonction de la largeur de la bande passante du filtre. Pour cela on synthétise un filtre passe-bas de Butterworth d'ordre 10 avec des fréquences de coupure différentes (normalisées par π).

La figure (Fig 4.3) montre l'invariance du gain de la structure optimale, par contre celui de la structure canonique varie sensiblement avec la fréquence de coupure et il est important pour les filtres à bande étroite.

Pour des filtres prototypes de Butterworth avec la même fréquence de coupure normalisée $\pi/20$, le tableau (Tab 4.1) donne le gain des deux types de structures considérées en fonction de l'ordre du filtre. On remarque pour un ordre élevé, le gain canonique est très important et le gain optimal demeure faible.

Tab 4.1: Gain de bruit de calcul d'un filtre de Butterworth avec une fréquence de coupure de $\pi/20$.

N	G_{can}	M_{can}	G_{optm}	M_{optm}
4	9.03093	49.15468	0.55554	6.77770
6	247.72205	1740.05439	0.71213	10.98492
8	8019.68367	72185.15306	0.85972	15.73755
10	276762.16968	3044393.86655	1.00251	21.02765
12	9932605.91334	129123888.87354	1.14226	26.84946

Qualité du filtrage avec un nombre de bits limité:

Afin de voir le comportement du filtre numérique réalisé par la structure minimisée vis-à-vis de la structure canonique, on simule un filtre passe-bas de Butterworth d'ordre 10 avec une fréquence de coupure normalisée de $\pi/4$. Pour cela, on utilisera comme type de signaux à l'entrée:

- Un signal bruit blanc de distribution gausienne, centrée et de

variance unité (Fig 4.4).

- Un signal périodique résultant de la somme de deux sinusoïdes (Fig 4.7), donné par:

$$U(k) = 0.3\ SIN\left(\frac{\pi}{8}k\right) + 0.2\ SIN\left(\frac{\pi}{2}k\right)$$

Aussi la représentation binaire en arithmétique à virgule fixe est effectuée par une fonction de quantification par arrondi avec caractéristique de saturation.

On constate que lorsque la représentation binaire est faite sur des mots dont la précision est celle de l'ordinateur (supposée infinie), la fonction de filtrage est correctement effectuée en éliminant les composantes de hautes fréquences pour le bruit blanc (Fig 4.4) et rejetant la frèquence $\pi/2$ pour le signal sinusoïdal (Fig 4.7). Cependant, en limitant la longueur des mots binaires représentant le signal de l'entrée, les coefficients du filtre et les registres de stockage, on remarque pour la structure minimisée les signaux de la sortie sont conservés à partir de 8 bits (Fig 4.5) et (Fig 4.8), alors que pour la structure canonique, le filtrage n'est satisfaisant qu'à partir de 12 bits (Fig 4.6 et 4.9).

Sensibilité du filtre avec un nombre de bits limité

Les figures (Fig 4.10 et 4.11) présentent l'effet de la variation des coefficients de la fonction de transfert du filtre simulé, dû à la longueur finie des registres utilisés. On remarque une nette amélioration de la réponse fréquentielle pour la structure minimisée par rapport à la structure canonique, en particulier le cas des registres à faible nombre de bits.

Variance de l'erreur de calcul à la sortie du filtre

La figure (Fig 4.12) présente la variation de l'erreur de la sortie du filtre en fonction du nombre de bits représentant la longueur du mot du registre utilisé. Dans le cas d'un accumulateur double, le nombre est représenté par b bits après l'opération arithmétique, le résultat est accumulé dans un registre de 2b bits. Contrairement à un accumulateur simple, auquel après chaque opération l'arrondi est effectué et le résultat est sur b bits, ce

qui illustre la différence en grandeur des variances sur les figures (Fig 4.12a et 4.12b). A noter que la variance simple est superieur à la variance double pour un nombre de bits fini.

D'autre part, on remarque le rapprochement entre la variance théorique donnée par (4.19) et la variance pratique , calculée pour une entrée bruit blanc centrée de variance unité, donnée par:

$$\sigma^2_{prat} = \frac{1}{L}\sum_{k=1}^{L} E^2(k) - \left[\frac{1}{L}\sum_{k=1}^{L} E(k)\right]^2 \qquad (4.50)$$

où $E(k) = y(k) - y'(k)$ ($y'(k)$ donnée par (3.4)) représente l'erreur à la sortie du filtre et L le nombre d'échantillons du signal de sortie. Ce qui justifie l'application de l'approche utilisée pour la structure optimale.

Remarque

Dans le calcul de la variance du bruit d'arrondi, on a utilisé la quantification par arrondi avec saturation où le point décimal considéré comme mobile, du fait que la mobilité de la virgule dans cette quantification est destinée à élargir la dynamique afin de ne considérer que les erreurs de calcul car la probabilité de dépassement sera faible.

4.7 Conclusion

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 été convenablement choisi afin de rendre les dépassements peu probables et l'augmentation du bruit de calcul qui en découle négligeable, comme il a été confirmé par les résultas précédents. D'autre part par une simple transformation orthogonale, une infinité de transformation d'optimisation T est possible pour un même filtre numérique réalisé par la structure optimale.

Bien que, les structures optimales offrent des gains de bruit minimum, leurs réalisations pratiques s'avèrent complexes lorsqu'il s'agit des filtres d'ordre N assez grand. En effet, de telles structures requièrent $(N+1)^2$ multiplications pour calculer

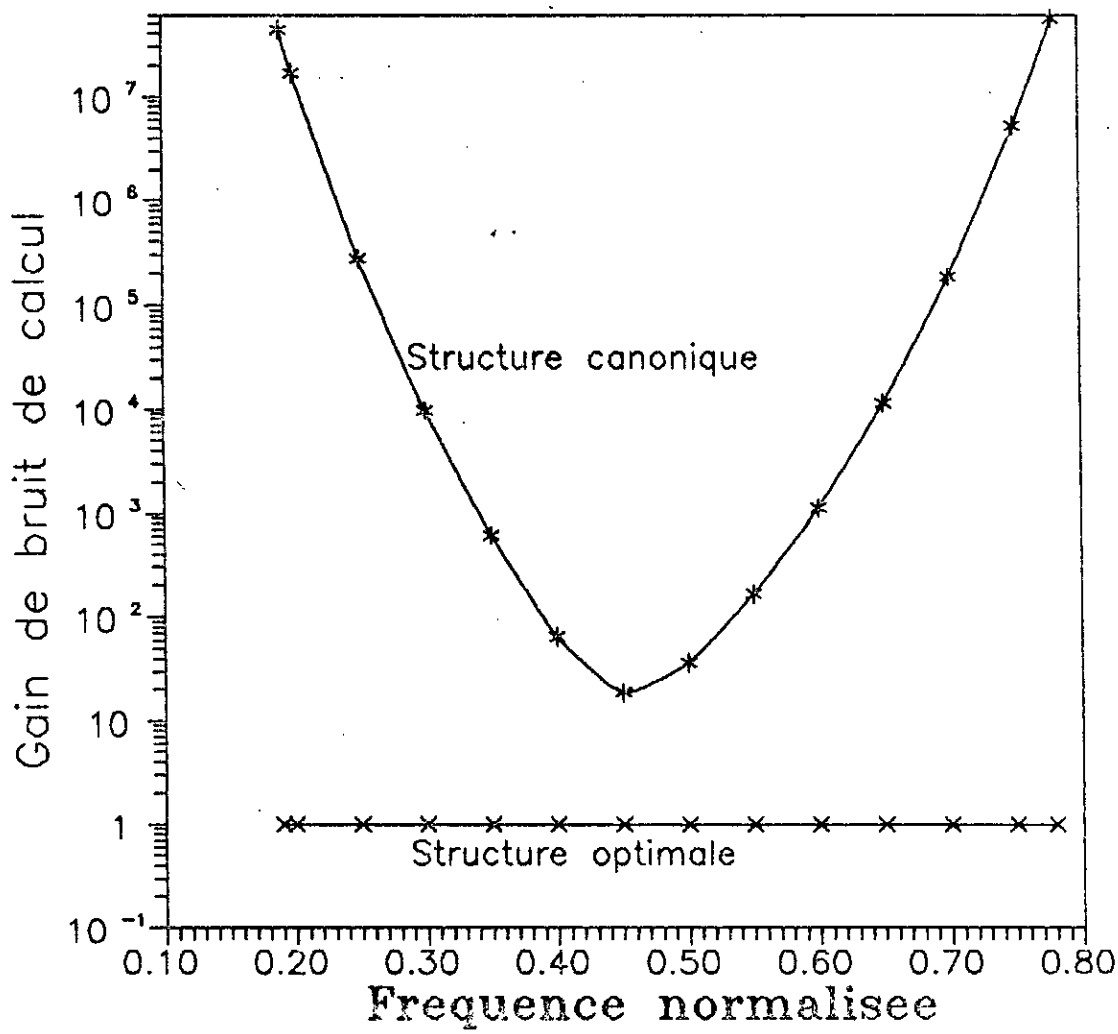


Fig.4.3: Gain de bruit de calcul d'un filtre pass-bas de Butterworth d'ordre 10 realise en structure optimale et canonique.

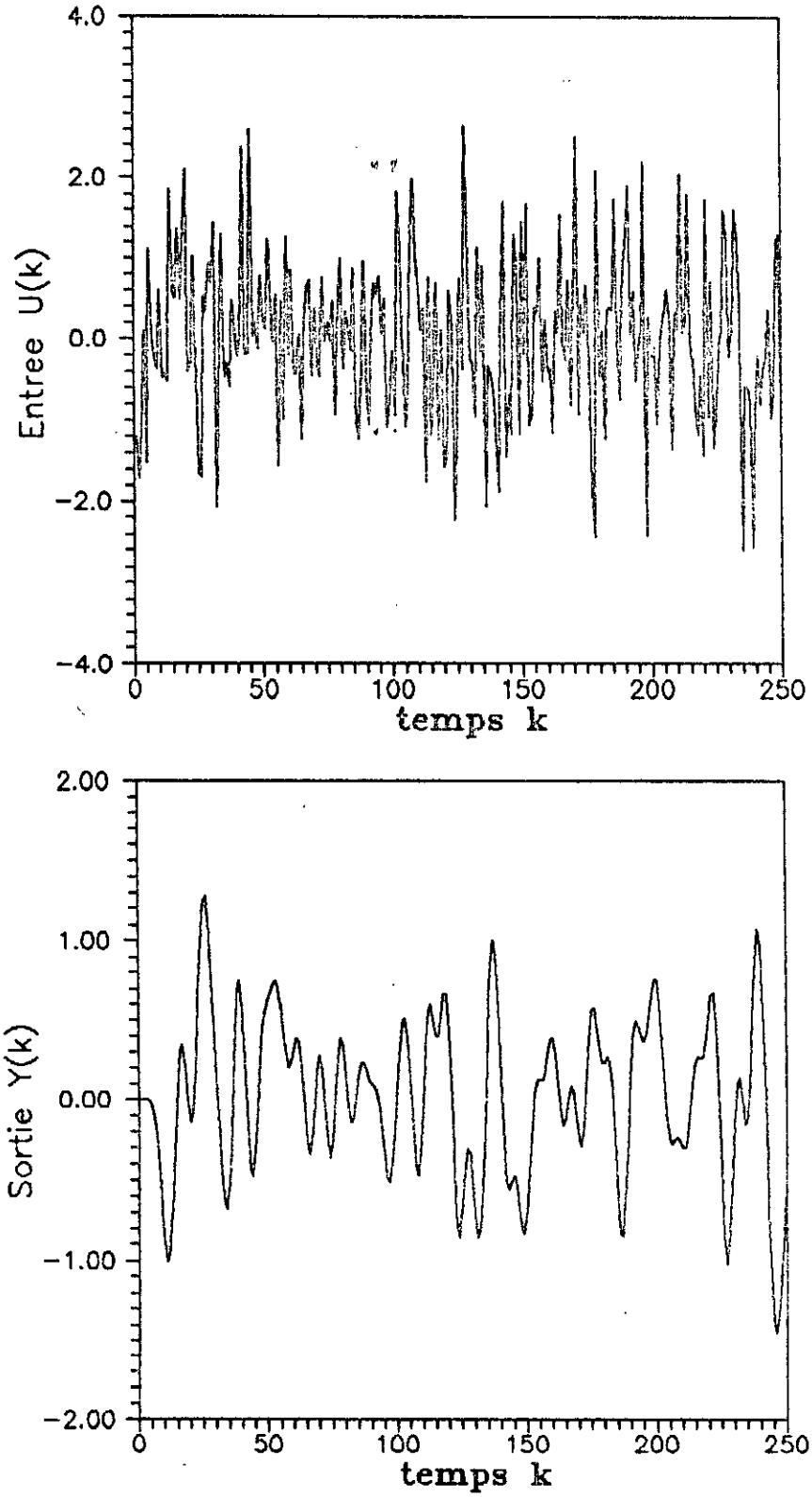


Fig.4.4: Signal bruit blanc centre de variance unite applique a un filtre pass-bas de Butterworth d'ordre 10.

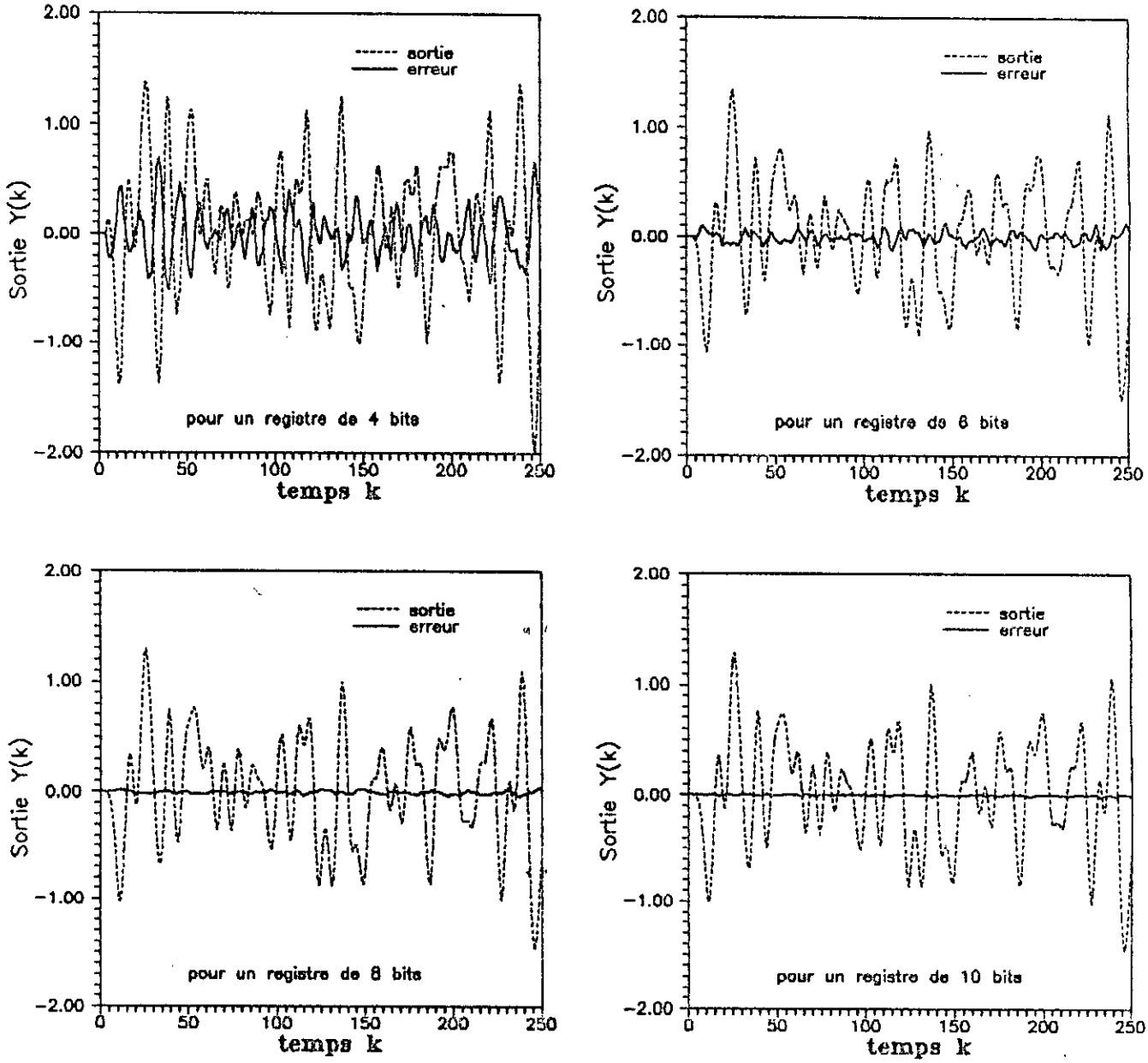


Fig 4.5: Signal bruit blanc centré de variance unité appliqué à un filtre passe-bas de Butterworth d'ordre 10 réalisé par la structure optimale, en utilisant des registres de stockages de longueurs finies de bits.

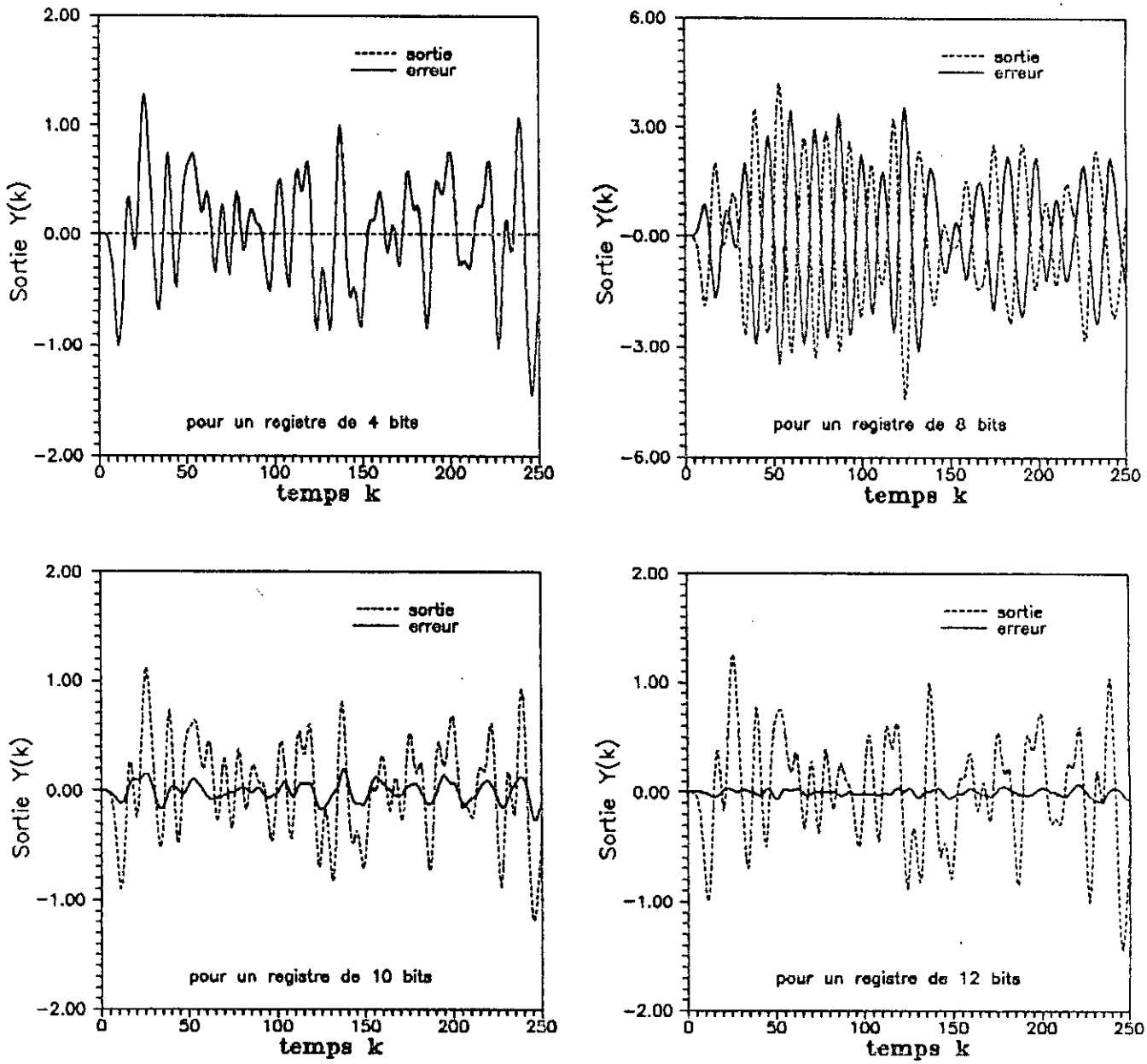


Fig 4.6: Signal bruit blanc centré de variance unité appliqué à un filtre passe-bas de Butterworth d'ordre 10 réalisé par la structure canonique en utilisant des registres de stockages de longueurs finies de bits.

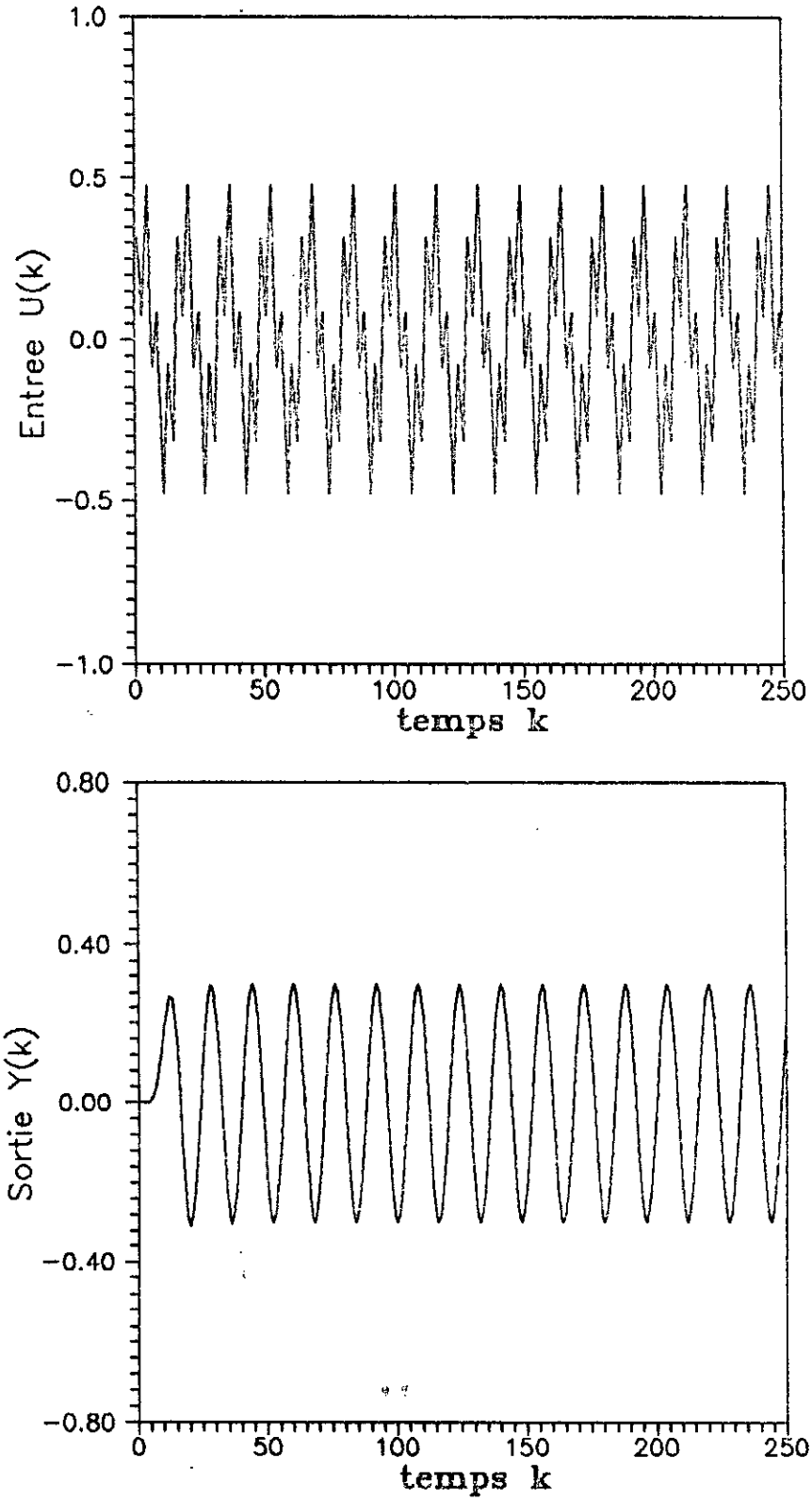


Fig.4.7: Signal sinusoidal applique a un filtre numerique de Butterworth d'ordre 10.

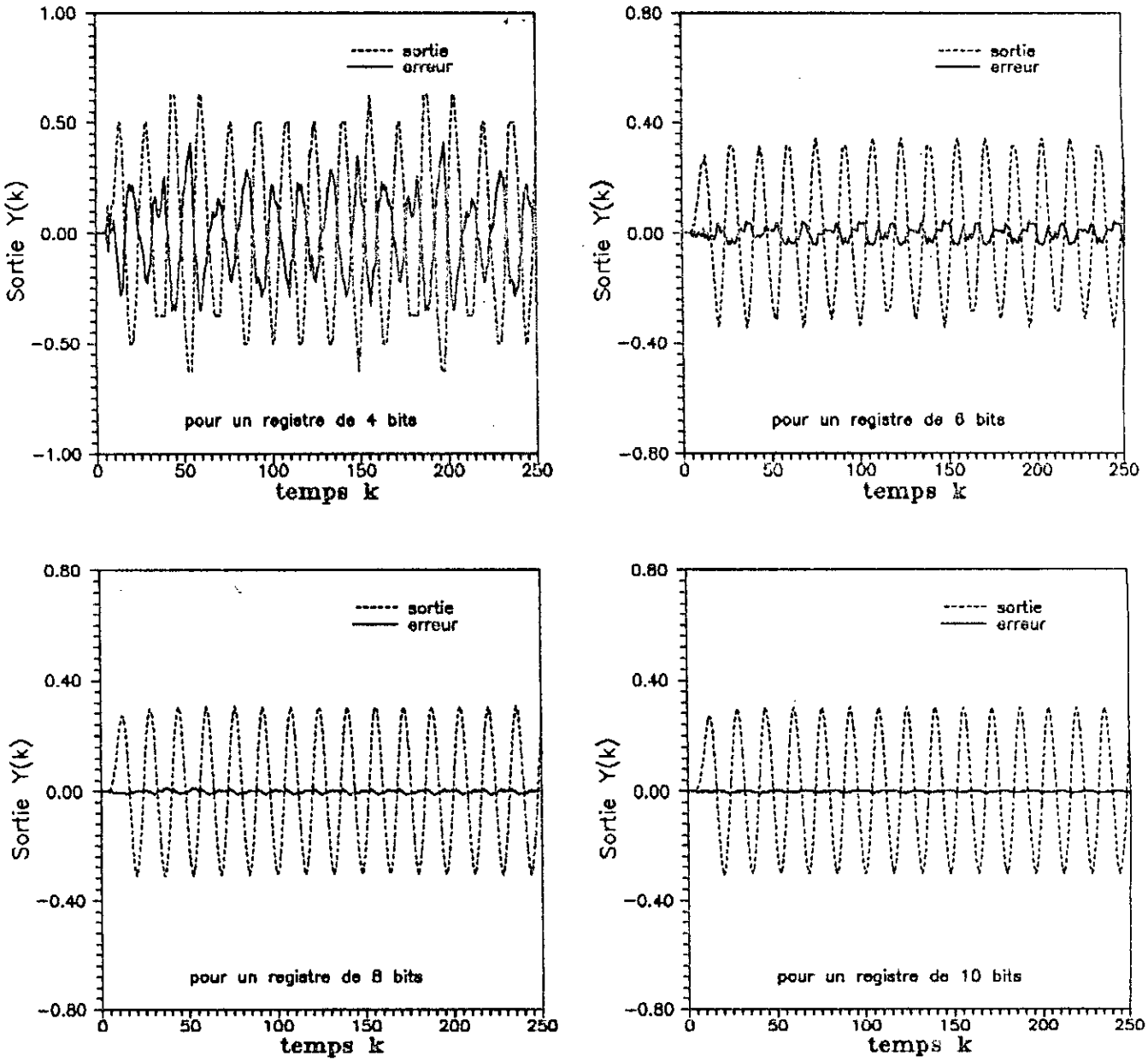


Fig 4.8: Signal sinusoïdal appliqué à un filtre passe-bas de Butterworth d'ordre 10 réalisé par la structure optimale en utilisant des registres de stockages de longueurs finies de bits.

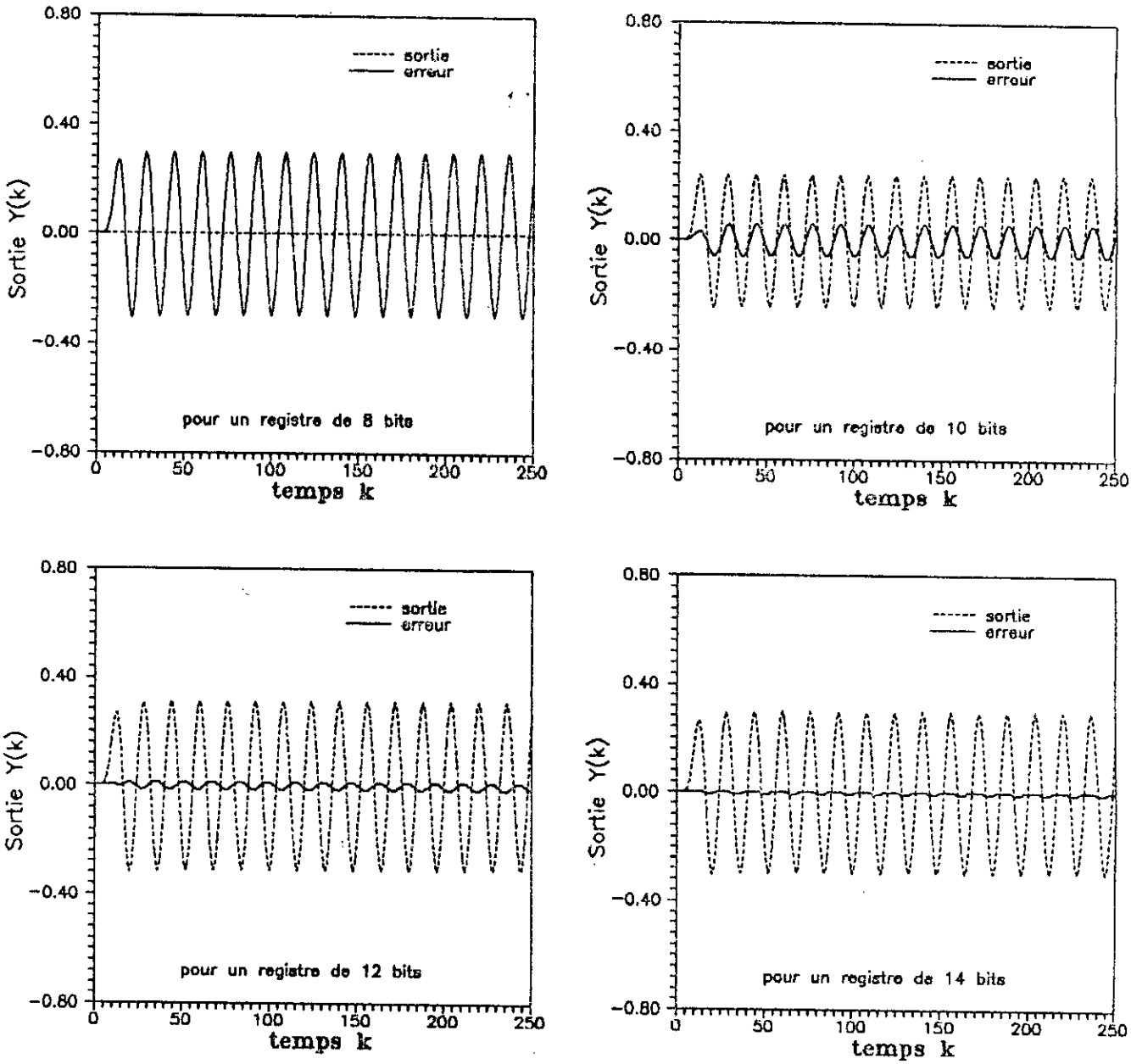


Fig 4.9: Signal sinusoïdal appliqué à un filtre passe-bas de Butterworth d'ordre 10 réalisé par la structure canonique en utilisant des registres de stockages de longueurs finies de bits.

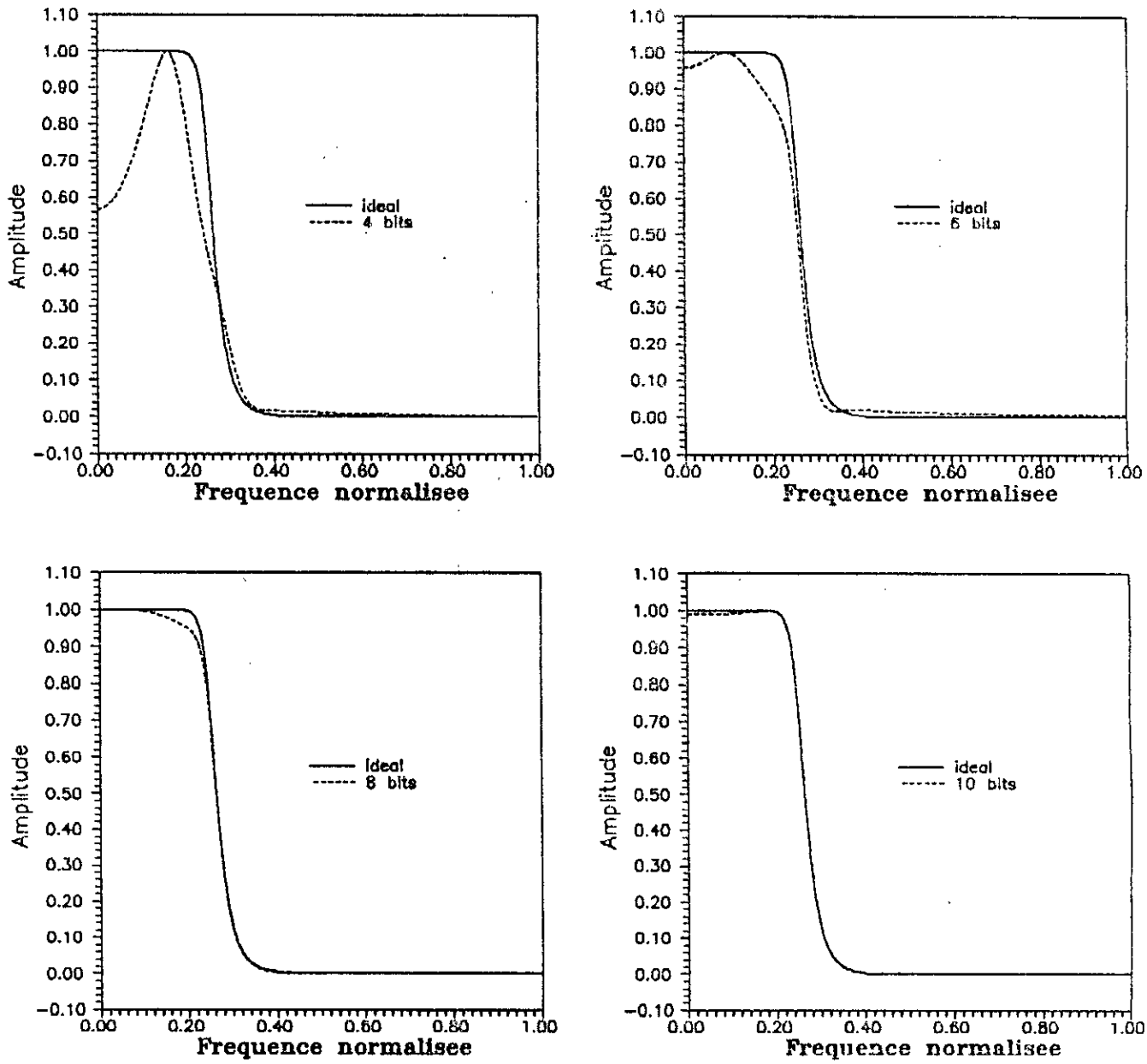


Fig 4.10: Spectre d'amplitude d'un filtre passe-bas de Butterworth d'ordre 10 réalisé par la structure optimale, en utilisant des registres de stockages de longueurs finies de bits.

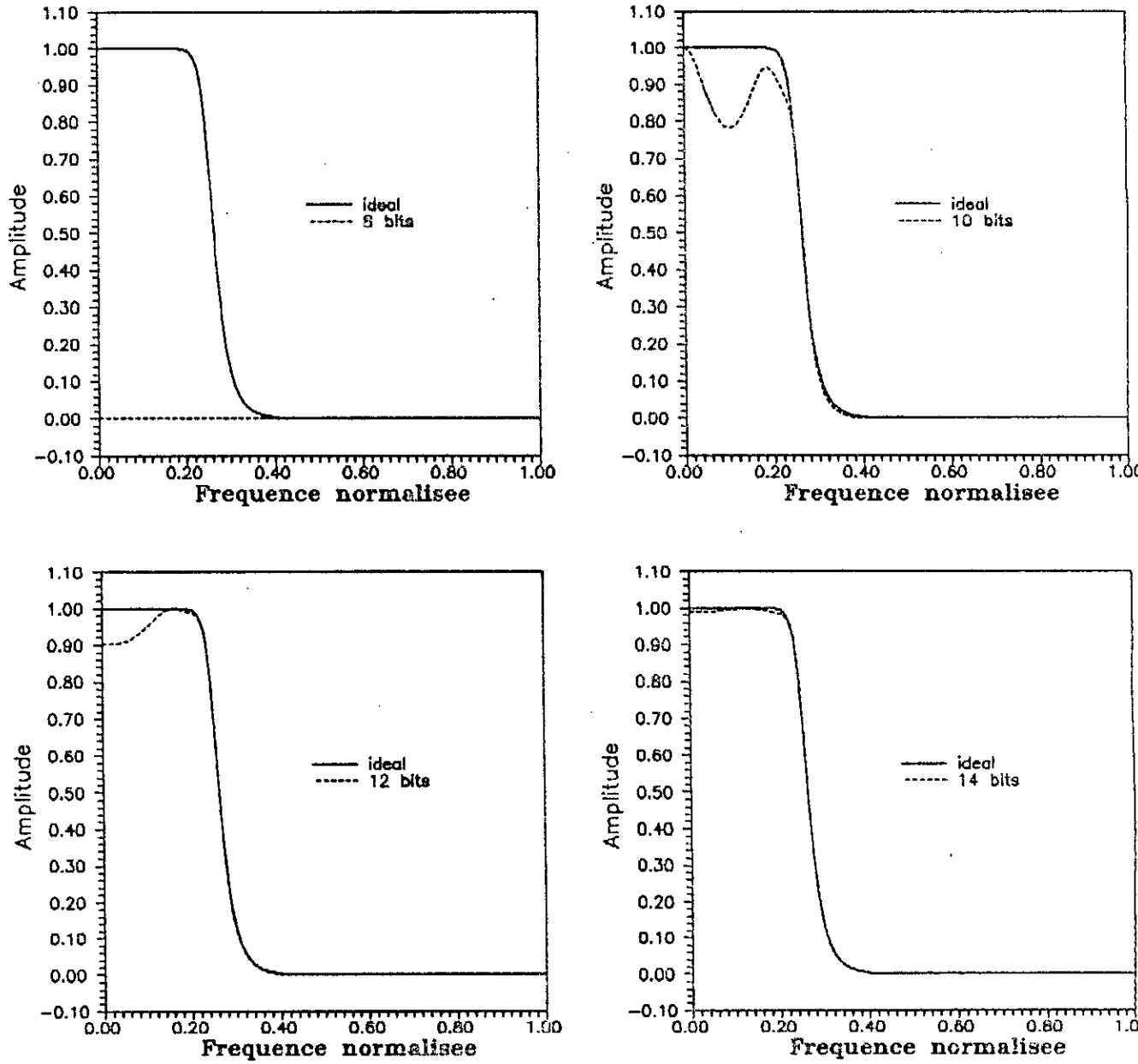
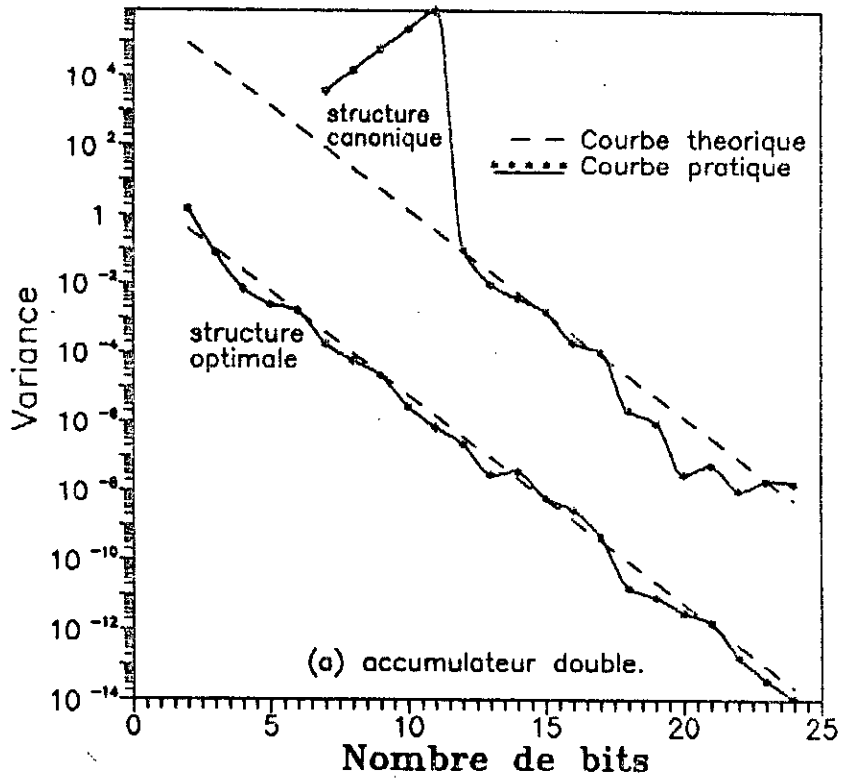


Fig 4.11: Spectre d'amplitude d'un filtre passe-bas de Butterworth d'ordre 10 réalisé par la structure canonique, en utilisant des registres de stockages de longueurs finies de bits.



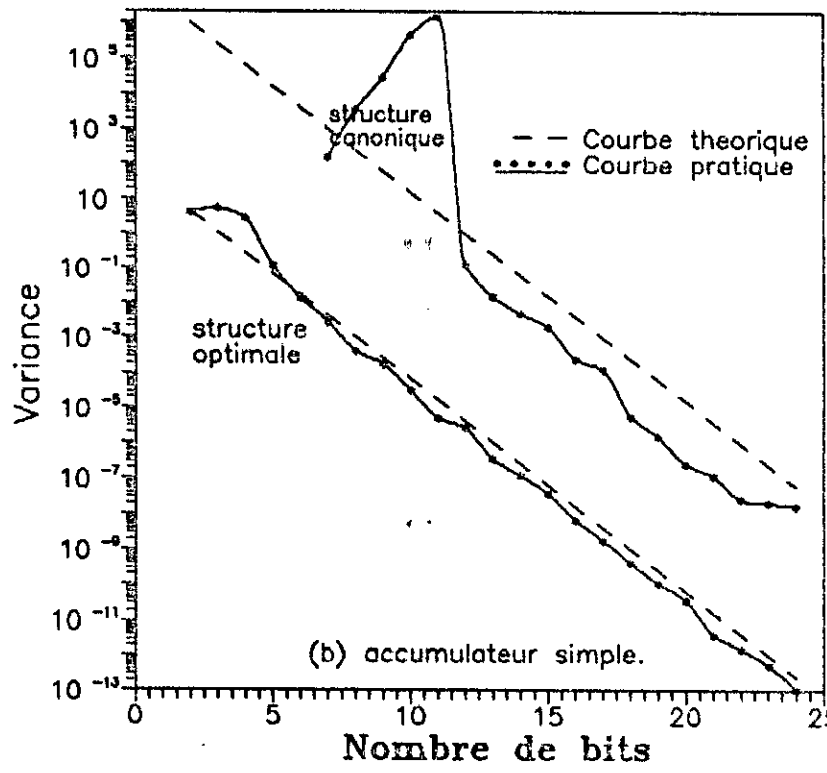


Fig.4.12: Variance de l'erreur a la sortie d'un filtre passe-bas de Butterworth d'ordre 10 pour une entree bruit blanc centree de variance unite.

CHAPITRE V

STRUCTURES DECOMPOSEES OPTIMALES

CHAPITRE V

STRUCTURES DÉCOMPOSÉES OPTIMALES

5.1 Introduction

Pour obtenir des structures de compromis entre le nombre de multiplications et le gain de bruit de calcul, on utilise une méthode traditionnelle qui permet de décomposer le filtre d'ordre N en des sections d'ordre 2, connectées soit en parallèle, soit en cascade, et éventuellement une cellule de premier ordre si N est impair. Plus souvent, les sections d'ordre 2 sont réalisées par la forme canonique ou directe, ce qui résulte en (2N+1) multiplications par sortie. Dans le cas des filtres à bruit minimum, la décomposition permet de réduire le nombre de multiplications de $(N+1)^2$ à (4N+1) avec des cellules de second ordre.

La réalisation décomposée augmente le gain minimal du filtre, mais simplifie énormément sa conception. De ce fait, si on décompose H(Z) en M sous-filtres $(A^{(i)}, B^{(i)}, C^{(i)}, D^{(i)})$ i = 1,...,M connectés en cascade ou en parallèle, on aura alors M problèmes de minimisation à résoudre. L'optimisation formulée dans ce cas par la connexion globale des M sous-filtres est différente de celle de la structure originale du filtre. Par conséquent, les modes d'ordre 2 du bloc, ne sont plus les mêmes que ceux du filtre d'ordre N, il en est de même pour le gain du bruit. Dans le cas d'une connexion cascade, on appellera les filtres à bruit minimum qui préservent une connexion globale des sous-filtres: les structures de bloc optimal et ceux qui préservent la connexion des sous-filtres avec une optimisation isolée de chaque section: les structures de sections optimales. En cascade les deux formes de structures citées ne présentent pas les mêmes caractéristiques.

Dans ce présent chapitre, on étudiera en premier lieu le procédé de minimisation de B. Bomar [3] et celui de S. Hwang [1] pour le cas de second ordre, ensuite on développera une théorie de

la représentation des structures de sections optimales et de bloc optimal pour le cas cascade [9], qui sera suivie des résultats interprétés par simulation.

5.2 Structure optimale du second ordre

Bruit de calcul

On considère un filtre numérique de second ordre ayant une fonction de transfert spécifiée par

$$H(Z) = \frac{Y(Z)}{U(Z)} = q_0 + \frac{q_1 Z^{-1} + q_2 Z^{-2}}{1 + p_1 Z^{-1} + p_2 Z^{-2}} \qquad (5.1)$$

où q_1, q_2, p_1 et p_2 sont des constantes réelles. Une telle fonction de transfert peut être réalisée par une structure d'espace d'état de la forme:

$$\begin{cases} x(k+1) = A\,x(k) + B\,u(k) \\ y(k) = C\,x(k) + D\,u(k) \end{cases} \qquad (5.2)$$

avec

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \; ; \; B = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \; ; \; C = \begin{bmatrix} c_1 & c_2 \end{bmatrix} \text{ et } D = q_0$$

où A est une matrice (2x2), B, C^T sont des vecteurs (2x1) et D le chemin direct entrée-sortie du filtre. En conséquence, on définit la matrice de covariance d'état K (2x2) et la matrice de l'élément bruit W (2x2) par:

$$K = A\,K\,A^T + B\,B^T$$
$$W = A^T\,W\,A + C\,C^T \qquad (5.3)$$

En utilisant la norme L_2 définie par (4.1b) (n=2), il est nécessaire (cf.section 4.2), que:

$$K_{11} = K_{22} = 1 \qquad (\delta=1) \qquad (5.4)$$

il vient par suite le gain du bruit de calcul pour une cellule d'ordre 2:

$$G' = \nu_0 + \nu_1 W_{11} + \nu_2 W_{22} \qquad (5.5)$$

où ν_0, ν_1 et ν_2 représentent le nombre d'arrondis utilisé pour le calcul de y(k), $x_1(k)$ et $x_2(k)$, respectivement. Si l'arrondi s'effectue après la somme (accumulateur à double précision), on a:

$$G' = 1 + W_{11} + W_{22} \qquad (\upsilon_0 = \upsilon_1 = \upsilon_2 = 1) \qquad (5.6)$$

Procédure de minimisation

La fonction de transfert de la structure d'espace d'état présentée par (5.1), peut s'exprimer sous la forme suivante:

$$H(Z) = q_0 + C^T (ZI - A)^{-1} B \qquad (5.7)$$

L'identification de (5.1) et (5.7), donne les relations entre les coefficients de la fonction de transfert et ceux de la structure (A,B,C), qui se résument dans le système suivant:

$$\begin{cases} q_1 = c_1 b_1 + c_2 b_2 \\ q_2 = c_1 b_2 a_{12} + c_2 b_1 a_{21} - c_1 b_1 a_{22} - c_2 b_2 a_{11} \\ p_1 = -(a_{11} + a_{22}) \\ p_2 = a_{11} a_{22} - a_{12} a_{21} \end{cases} \qquad (5.8)$$

Ces expressions forment ainsi, un système de 4 équations à 8 coefficients de (A,B,C); elles permettent d'avoir 4 réalisations équivalentes qui diffèrent seulement par les signes des coefficients (a_{11}, a_{12}, a_{21}, a_{22}, b_1, b_2, c_1, c_2) [3]. En plus du système (5.8), deux autres coefficients de contraintes s'ajoutent à cause des exigences de la normalisation (5.4). Ce qui permet d'écrire (5.3a) comme suit:

$$-A\ K\ A^T + K = B\ B^T \qquad (5.9)$$

Sachant que $K_{12} = K_{21}$ (K matrice symétrique) et $K_{11} = K_{22} = 1$, en substituant les éléments de A, K et B dans (5.9), on obtient:

$$\begin{cases} a_{11}^2 + a_{12}^2 + b_1^2 - 1 = -2\, a_{11} a_{12} \dfrac{a_{11}a_{21} + a_{12}a_{22} + b_1 b_2}{1 - a_{12}a_{21} - a_{11}a_{22}} \\ a_{11}^2 + a_{22}^2 + b_2^2 - 1 = -2\, a_{21} a_{12} \dfrac{a_{11}a_{21} + a_{12}a_{22} + b_1 b_2}{1 - a_{12}a_{21} - a_{11}a_{22}} \end{cases} \qquad (5.10)$$

Ainsi, (5.8) et (5.10) forment un système de 6 équations à 8 coefficients de (A,B,C). De ce fait, deux autres contraintes additives, caractérisant la structure à bruit minimum, sont données par les deux conditions nécessaires et suffisantes d'optimisation [8]:

$$\begin{cases} K = S\ W\ S & (S \text{ matrice diagonale}) \\ K_{ii} W_{ii} = K_{jj} W_{jj} & (K_{ii} = K_{jj} = 1\ ;\ i,j = 1,2) \end{cases} \qquad (5.11)$$

En termes d'éléments (A,B,C), les conditions (5.11) deviennent:

$$\begin{cases} a_{11} = a_{22} \\ b_1 c_1 = b_2 c_2 \end{cases} \qquad (5.12)$$

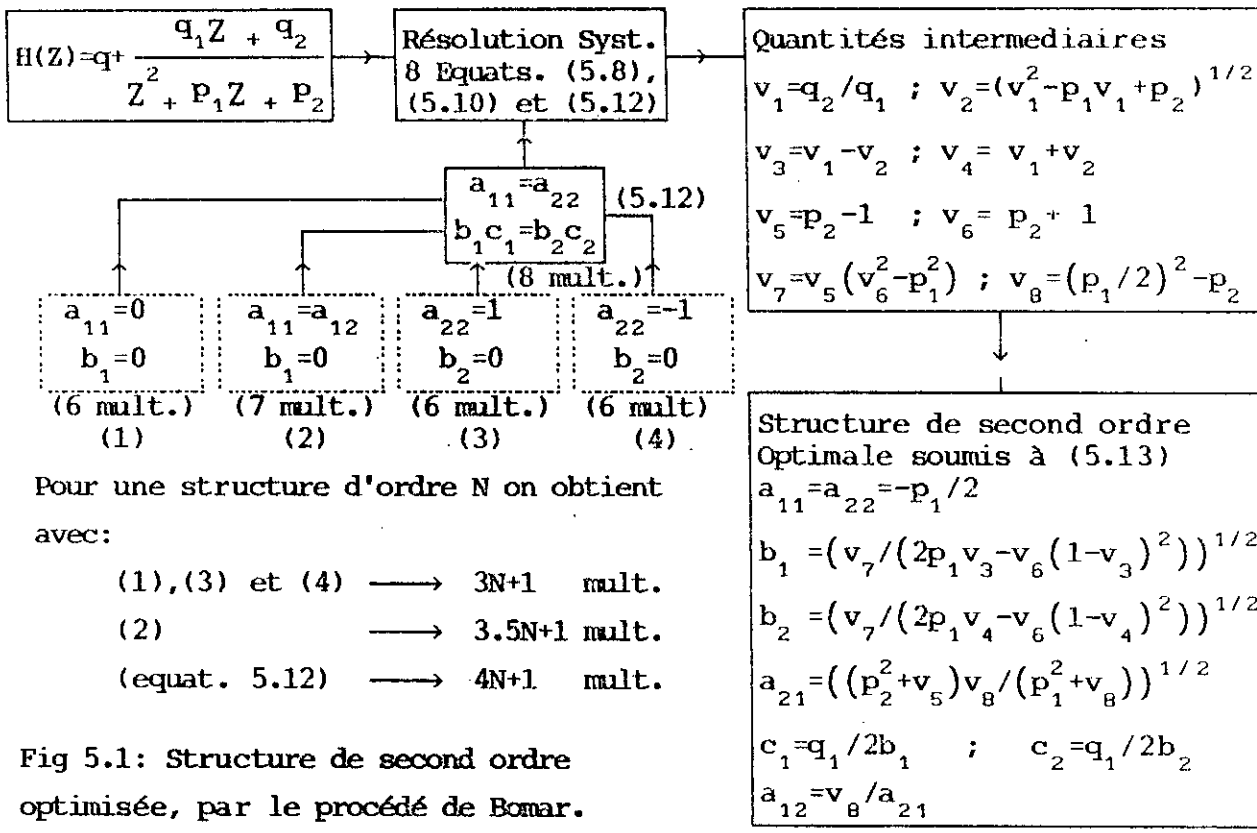
Par conséquent le système formé par (5.8), (5.10) et (5.12) permet de donner les coefficients de la structure optimale du second ordre (Fig 5.1). Ce processus purement algébrique et plus direct, a été proposé par B. Bomar [3]; il permet d'obtenir un certain nombre de structures optimales d'ordre 2, auxquelles parmi ces dernières on trouve celles qui réduisent davantage le nombre (4N) multiplications à (3N) en utilisant d'autres contraintes comme le résume la figure (5.1).

La procédure de S. Hwang [1], décrit dans le chapitre précedant dans le cas général d'ordre N, permet de donner dans le cas particulier (N=2) la structure optimale, par un processus de calcul assez direct et plus simple. Ce dernier consiste à établir une transformation de minimisation et de normalisation T (2x2) (cf. section 4.6), dont le calcul est résumé dans un organigramme présenté à la figure (5.2).

Remarques

1- On note la simplicité et la rapidité de calcul par rapport au cas général d'ordre N, pour l'obtention d'une structure minimisée. Par exemple, le calcul de K et W ou celui des valeurs et vecteurs propres d'une matrice, etc..., d'où l'avantage d'utiliser les structures décomposées.

2- D'autres procédures de minimisation des structures d'ordre 2, n'ont pas été citées dans ce travail, comme celle de C.W.Barnes [4] [14], concerne surtout les structures normales, et celle de Mullis Roberts [2], [8], obtenu du cas général d'ordre N. En fait, toutes ces structures citées, diffèrent du point de vue traitement mathématique, mais aboutissent à un même gain optimal.



Pour une structure d'ordre N on obtient avec:

(1),(3) et (4) ⟶ 3N+1 mult.

(2) ⟶ 3.5N+1 mult.

(equat. 5.12) ⟶ 4N+1 mult.

Fig 5.1: Structure de second ordre optimisée, par le procédé de Bomar.

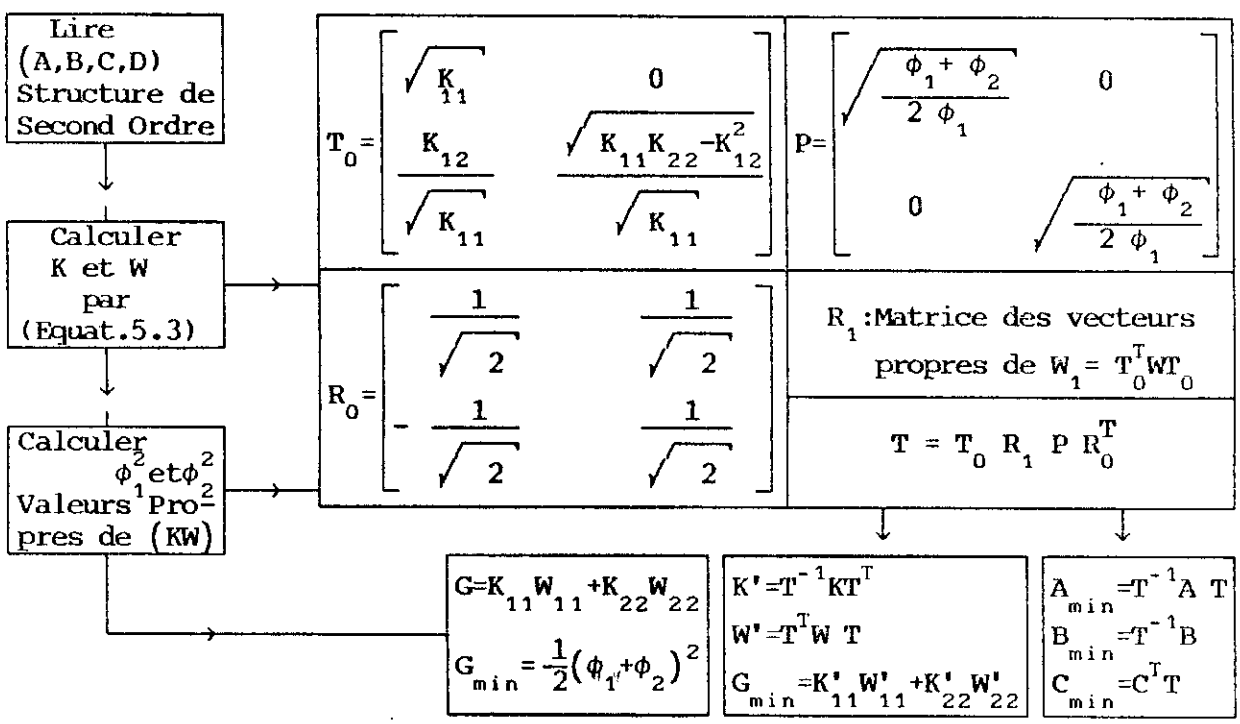


Fig 5.2: Structure de second ordre optimisée par le procédé de S. Hwang.

5.3 Structure parallèle optimale

Au lieu de réaliser H(Z) directement par (2.2), on peut effectuer une décomposition en somme de fractions rationnelles de second ordre (N pair), réalisées séparément (fig 5.3). De ce fait, on écrit H(Z) sous la forme:

$$H(Z) = \frac{\sum_{i=0}^{N} b_i Z^{-i}}{1 + \sum_{i=1}^{N} a_i Z^{-i}} = b_0 + \frac{\sum_{i=1}^{N} c_i Z^{N-i}}{Z^N + \sum_{i=1}^{N} a_i Z^{N-i}} \qquad (5.13)$$

avec

$$c_i = b_i - b_0 a_i \qquad i = 1,\ldots,N$$

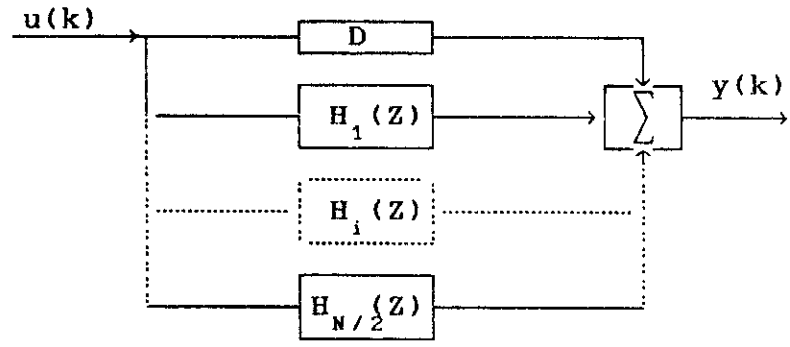


Fig 5.3: Structure parallèle

L'approche de Baistrow [16] permet de décomposer le dénominateur de (5.13) en un produit de polynôme d'ordre 2 (N pair). Ce qui permet de faire une décomposition en fraction rationnelle de second ordre. Il vient par cette décomposition:

$$H(Z) = b_0 + \sum_{i=1}^{N/2} \frac{q_{1i} Z + q_{2i}}{Z^2 + p_{1i} Z + p_{2i}} \qquad \text{(N pair)} \qquad (5.14)$$

C'est la forme représentée par la figure (5.3), à savoir:

$$D = b_0$$

et

$$H_i(Z) = \frac{q_{1i} Z^{-1} + q_{2i} Z^{-2}}{1 + p_{1i} Z^{-1} + p_{2i} Z^{-2}} \qquad (5.15)$$

Les pôles de la fonction de transfert $H_i(Z)$, sont en général des complexes conjugués donnés par:

$$\lambda_i = \alpha_i + j\,\beta_i \qquad \text{et} \qquad \lambda_i^* = \alpha_i - j\,\beta_i \tag{5.16}$$

avec

$$\alpha_i = -p_{1i}/\,2$$

$$\beta_i = -\frac{1}{2}\sqrt{4p_{2i}-p_{1i}^2}$$

En utilisant la construction (2.11), on peut avoir l'ensemble des structures canoniques du second ordre formant H(Z):

$$A_i = \begin{bmatrix} 0 & 1 \\ -p_{2i} & -p_{1i} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -(\alpha_i^2+\beta_i^2) & 2\alpha_i \end{bmatrix}$$

$$B_i = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad \text{et} \quad C_i = \begin{bmatrix} q_{2i} & q_{1i} \end{bmatrix} \tag{5.17}$$

A partir d'une telle réalisation (A_i, B_i, C_i) $i=1,\ldots,N/2$, on obtient la structure optimale d'ordre 2 $(A_{min}^{(i)}, B_{min}^{(i)}, C_{min}^{(i)})$, en utilisant le procédé de minimisation de la figure (5.1) ou (5.2). Par conséquent le gain total issu de toutes les cellules de structures optimales de second ordre, connectées en parallèle, est donné par:

$$G_{min} = \sum_{i=1}^{N/2} G_{min}^{(i)}$$

avec

$$G_{min}^{(i)} = \frac{1}{2}\,(\phi_{1i} + \phi_{2i})^2 \tag{5.18}$$

où ϕ_{1i}^2 et ϕ_{2i}^2 représentent les valeurs propres de la $i^{ème}$ cellule.

Remarque

Si N est impair, on aura une cellule additive du premier ordre, dont la fonction de transfert est:

$$H(Z) = \frac{\alpha}{Z - \beta}$$

et de gain optimal invariant, donné par:

$$G_1 = \frac{\alpha^2}{(1-\beta^2)} \qquad (\delta=1)$$

Exemple numérique

On reprend l'exemple cité dans le chapitre précédent (filtre de Butterworth d'ordre 4 ayant une frèquence de coupure de

$\pi/2$), en considérant cette fois la structure décomposée. Dans le cas d'une décomposition parallèle, la fonction de transfert du filtre considéré s'écrit:

$$H(Z) = H_1(Z) + H_2(Z) + D$$

avec

$$H_1(Z) = \frac{0.14318\ Z - 0.10801}{Z^2 - 1.72593\ Z + 0.74744}$$

$$H_2(Z) = \frac{-\ 0.14294\ Z + 0.12819}{Z^2 - 1.86380\ Z + 0.88703} \qquad \text{et} \quad D = .00003$$

Le tableau (Tab 5.1) présente les deux cellules du second ordre , où l'on note la grandeur assez faible du gain de bruit de calcul du filtre d'ordre 4 réalisé par la structure décomposée parallèle canonique par rapport à celle de la structure globale canonique (cf section 4.5).

5.4 Structure cascade optimale

La décomposition de H(Z) en produit correspond à la structure cascade où le filtre est réalisé par une suite de cellule du second ordre (Fig 5.4).

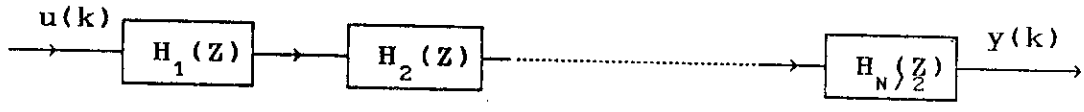


Fig 5.4 : Structure cascade

La réalisation obtenue par la connexion des sections du second ordre optimisée séparément diffère de celle de l'ensemble du bloc optimisé, du fait que les sections avales ne sont pas correctement normalisées. En isolation, la matrice de covariance de chaque section est utilisée avec une entrée bruit blanc commune pour chaque sous-filtre, alors qu'en cascade la matrice de covariance

Tab 5.1: Filtre passe-bas de Butterworth d'ordre 4 réalisé par deux sections de second ordre connectées en parallèle.

	Cellule 1	Cellule 2
A_{can}	$\begin{bmatrix} 0 & 1 \\ -.74744 & 1.72593 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \\ -.88703 & 1.86380 \end{bmatrix}$
B_{can}	$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$
C_{can}	$\begin{bmatrix} -.10801 & .14318 \end{bmatrix}$	$\begin{bmatrix} .12819 & -.14294 \end{bmatrix}$
$Gain_{can}$	22.69933	47.05897
Gain de la structure décomposée canonique :	69.75830	
T	$\begin{bmatrix} -7.69738 & -15.44265 \\ -4.87877 & -13.51087 \end{bmatrix}$	$\begin{bmatrix} -8.07361 & -13.74363 \\ -6.10383 & -14.26076 \end{bmatrix}$
modes d'ordre 2	.06783 ; .92138	.45213 ; .8056
A_{optm}	$\begin{bmatrix} .86296 & .02395 \\ -.11421 & .86296 \end{bmatrix}$	$\begin{bmatrix} .93190 & .17997 \\ -.10331 & .93190 \end{bmatrix}$
B_{optm}	$\begin{bmatrix} .53887 \\ -.26860 \end{bmatrix}$	$\begin{bmatrix} .43983 \\ -.25838 \end{bmatrix}$
C_{optm}	$\begin{bmatrix} .13285 & -.26653 \end{bmatrix}$	$\begin{bmatrix} -.16249 & .27661 \end{bmatrix}$
$Gain_{optm}$	0.48927	0.79105
Gain de la structure décomposée optimale :	1.28033	

pour chaque section avale est calculée non pas pour une entrée bruit blanc mais plutôt par une entrée correlée. En général, pour une normalisation correcte de la structure bloc, on doit calculer l'ensemble des matrices K et W pour l'ensemble du filtre.

La décomposition de H(Z) présentée par (2.2), en produit de sections second ordre donne:

$$H(Z) = \frac{\sum_{i=0}^{N} b_i Z^{-i}}{1 + \sum_{i=1}^{N} a_i Z^{-i}}$$

$$= b_0 \prod_{i=1}^{N/2} \frac{Z^2 + q_{1i}Z + q_{2i}}{Z^2 + p_{1i}Z + p_{2i}} = \prod_{i=1}^{N/2} H_i(Z) \qquad (5.19)$$

avec

$$H_i(Z) = D_i \frac{Z^2 + q_{1i}Z + q_{2i}}{Z^2 + p_{1i}Z + p_{2i}} = D_i \left(1 + \frac{q'_{1i}Z + q'_{2i}}{Z^2 + p_{1i}Z + p_{2i}}\right)$$

$$= D_i + \frac{(D_i q'_{1i})Z + (D_i q'_{2i})}{Z^2 + p_{1i}Z + p_{2i}} \qquad (5.20)$$

où

$$q'_{1i} = q_{1i} - p_{1i} \;,\quad q'_{2i} = q_{2i} - p_{2i} \quad \text{et} \quad D_i = (b_0)^{2/N}$$

En utilisant la construction (2.11) et l'équation (5.20), on peut construire l'ensemble des structures canoniques du second ordre formant H(Z), à savoir:

$$A_i = \begin{bmatrix} 0 & 1 \\ -p_{2i} & -p_{1i} \end{bmatrix} \quad ; \; B_i = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$C_i = \begin{bmatrix} D_i(q_{2i} - p_{2i}) & D_i(q_{1i} - p_{1i}) \end{bmatrix} \quad ; \; D_i = (b_0)^{2/N} \qquad (5.21)$$

Une fois la réalisation de chaque section (A_i, B_i, C_i, D_i) i=1,...,N/2, est obtenue, on utilisera l'algorithme de la figure (5.2), qui permet de minimiser ces sections séparément d'une part et d'autre part, l'étape à suivre est d'obtenir la structure de sections optimales $(A_{so}, B_{so}, C_{so}, D_{so})$ qui est résumée dans l'organigramme présenté par la figure (5.5). On note que les éléments diagonaux de la matrice K_{so} ne sont pas tous égaux à $1/\delta^2$.

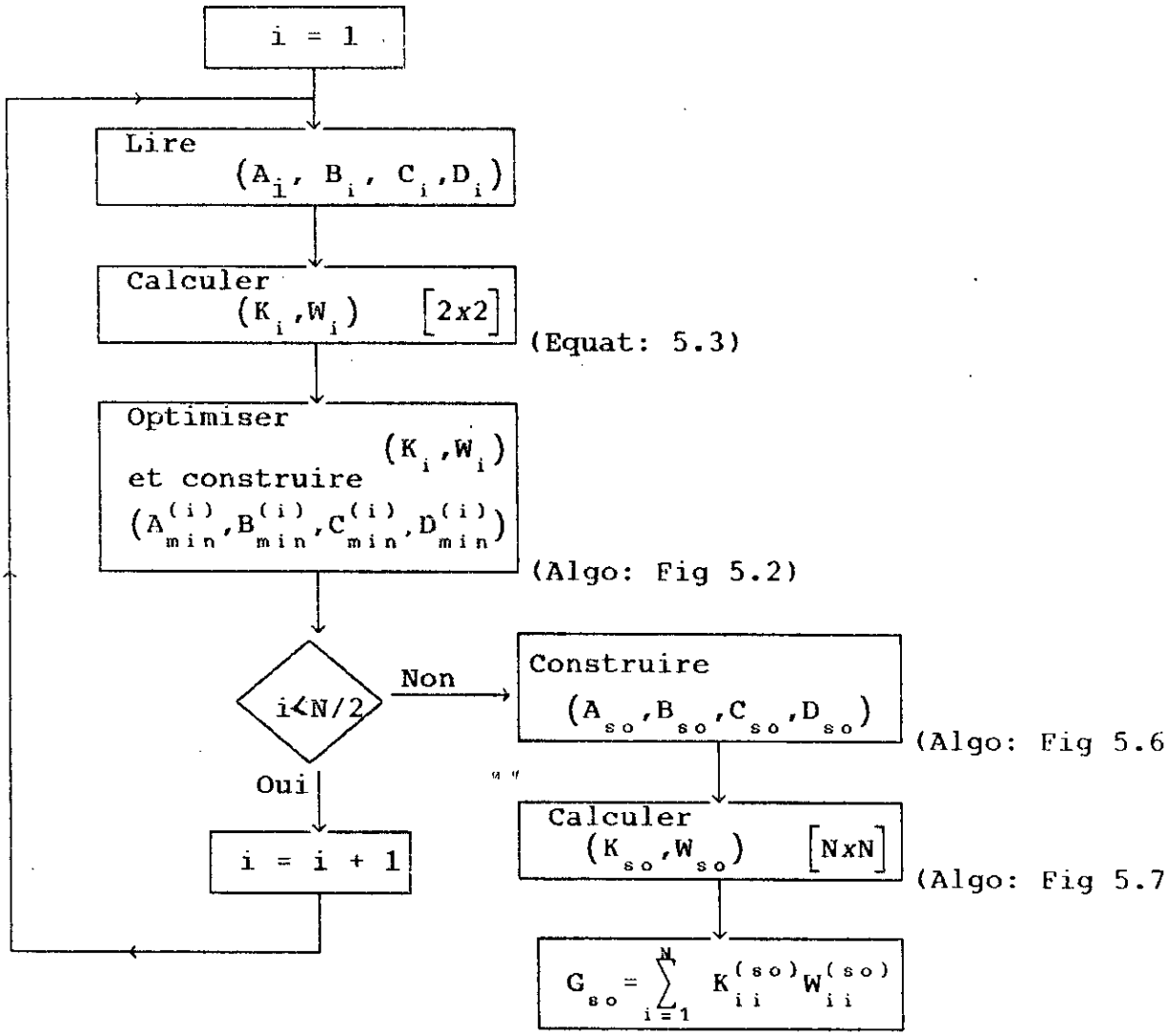


Fig 5.5 : Organigramme d'une structure cascade avec sections optimales (so)

Exemple numérique

La fonction transfert de l'exemple précédant, décomposée selon la représentation (5.19) en un produit de deux fonctions d'ordre 2, s'écrit comme suit:

$$H(Z) = H_1(Z)\ H_2(Z)$$

avec

$$H_1(Z) = 0.00558\ \frac{0.02081\ Z + 0.00141}{Z^2 - 1.72593\ Z + 0.74744}$$

et

$$H_2(Z) = 0.00558 \; \frac{0.02159\ Z + 0.00063}{Z^2 - 1.86380\ Z + 0.88703}$$

Le tableau (Tab 5.2) présente les deux cellules du second ordre.

Tab 5.2: Filtre passe-bas de Butterworth d'ordre 4 réalisé par deux sections de second ordre connectées en cascade.

	Cellule 1	Cellule 2
A_{can}	$\begin{bmatrix} 0 & 1 \\ -.74744 & 1.72593 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \\ -.88703 & 1.86380 \end{bmatrix}$
B_{can}	$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$
C_{can}	$\begin{bmatrix} .00141 & .02082 \end{bmatrix}$	$\begin{bmatrix} .00063 & .02159 \end{bmatrix}$
D_{can}	.00558	
$Gain_{can}$	6.59816	32.41557
T	$\begin{bmatrix} -1.02786 & -10.29038 \\ 1.02786 & -8.89536 \end{bmatrix}$	$\begin{bmatrix} -1.16311 & -14.21831 \\ 1.16311 & -13.38690 \end{bmatrix}$
modes d'ordre 2	.12304 ; .64263	.43253 ; .91369
A_{optm}	$\begin{bmatrix} .86296 & .01470 \\ -.18608 & .86296 \end{bmatrix}$	$\begin{bmatrix} .93190 & .11766 \\ -.15803 & .93190 \end{bmatrix}$
B_{optm}	$\begin{bmatrix} .52181 \\ -.05212 \end{bmatrix}$	$\begin{bmatrix} .44282 \\ -.03622 \end{bmatrix}$
C_{optm}	$\begin{bmatrix} .01995 & -.19977 \end{bmatrix}$	$\begin{bmatrix} -.16249 & .27661 \end{bmatrix}$
D_{optm}	.00558	.00558
$Gain_{optm}$	0.29312	0.90616

Par conséquent, les deux cellules réalisées en structures minimisées et connectées en cascade selon la structure présentée par la figure (fig5.6), on obtient la structure décomposée cascade de sections optimales, donnée par:

$$A_{so}=\begin{bmatrix} .86296 & .01470 & 0 & 0 \\ -.18608 & .86296 & 0 & 0 \\ .00883 & -.08846 & .93190 & .11766 \\ -.00072 & .00723 & -.15803 & .93190 \end{bmatrix} ; B_{so}=\begin{bmatrix} .52181 \\ -.05212 \\ .00247 \\ -.00020 \end{bmatrix}$$

$$C_{so}=\begin{bmatrix} .00011 & -.00111 & .02438 & -.29807 \end{bmatrix} ; D_{so}= .00003$$

Les matrices K_{so} et W_{so} correspondantes sont données par:

$$K_{so}=\begin{bmatrix} 1 & -.67859 & .20971 & -.13292 \\ -.67859 & 1 & -.40805 & .47291 \\ .20971 & -.40805 & .26405 & -.19297 \\ -.13292 & .47291 & -.19297 & .53993 \end{bmatrix}$$

$$W_{so}=\begin{bmatrix} .16383 & -.11117 & .11285 & .09737 \\ -.11117 & .11205 & -.18488 & -.02737 \\ .11285 & -.18488 & .45308 & -.16193 \\ .09737 & .02737 & -.16193 & .45308 \end{bmatrix}$$

Par conséquent, le gain du bruit de calcul est donné par:

$$Gain_{so}=\sum_{i=1}^{4} K_{ii}^{(so)} W_{ii}^{(so)}= 0.64016$$

On remarque bien, que la matrice de covariance K n'est pas représentée correctement avec la normalisation du filtre (δ=1), les éléments diagonaux ne sont pas tous èguaux à 1.

Structure de bloc optimal

Ayant la réalisation canonique de chaque section d'ordre 2, (A_i, B_i, C_i, D_i) i=1,...,N/2, donnée par l'équation (5.20). On doit trouver, dans une première étape, la structure bloc originale (A,B,C,D) d'ordre N de l'ensemble des sections mises en cascade. Il existe une technique mathématique récurrente, qui permet d'obtenir une telle structure à partir de la description suivante:

Pour la cellule 1, on a:

$$\begin{cases} x_1(k+1) = A_1 x_1(k) + B_1 u(k) \\ y_1(k) = C_1 x_1(k) + D_1 u(k) \end{cases}$$

Pour la cellule 2, on obtient:

$$\begin{cases} x_2(k+1) = A_2 x_2(k) + B_2 y_1(k) \\ y_2(k) = C_2 x_2(k) + D_2 y_1(k) \end{cases}$$

Par conséquent, la structure bloc d'ordre 4 formée à partir de ces deux premières cellules, est donnée sous la forme suivante:

$$x(k+1) = \begin{bmatrix} A_1 & 0 \\ B_2C_1 & A_2 \end{bmatrix} x(k) + \begin{bmatrix} B_1 \\ B_2D_1 \end{bmatrix} u(k)$$

$$y(k) = \begin{bmatrix} D_2C_1 & C_2 \end{bmatrix} x(k) + D_2D_1\, u(k)$$

Si on continue jusqu'à la $(N/2)^{ème}$ cellule d'ordre 2, on obtient ainsi la structure bloc d'ordre N, donnée par la figure (5.6). Ainsi, la structure bloc (A,B,C,D) obtenue, on calcule l'ensemble des matrices (K,W) de cette structure d'ordre N, par les équations (4.14) et (4.29) en utilisant l'algorithme de la figure (5.7).

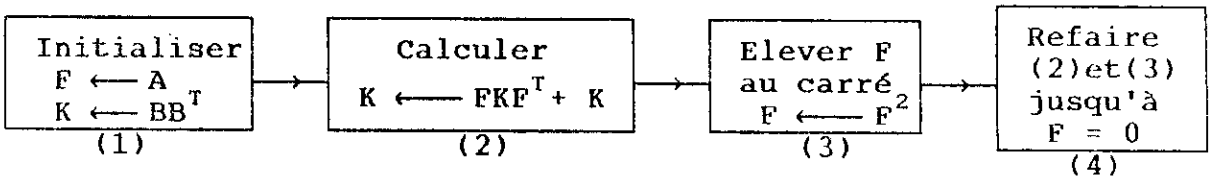


Fig 5.7: Algorithme de calcul des matrices K et W

A partir de (K,W) de l'ensemble du filtre, on extrait les blocs diagonaux (K_i,W_i) d'ordre 2, i=1,...,N/2, K_i et W_i seront la matrice de covariance et celle de l'élément bruit, respectivement, de chaque section individuelle. De ce fait, on optimise ces différentes sections individuelles en utilisant le procédé de la figure (5.1) ou (5.2), ainsi on trouve pour chaque pair (K_i,W_i), une transformation d'état d'optimisation T_i (2x2) i=1,...,N/2, qui satisfont par construction le choix par bloc diagonal et permettent de trouver les structures optimales de second ordre connectées en cascade. Par exemple, dans le cas de la connexion de deux cellules d'ordre 2, on opte pour le choix bloc diagonal de forme:

$$\left[\begin{array}{c:c} A & B \\ \hdashline C & D \end{array}\right] = \begin{bmatrix} I & 0 & 0 \\ 0 & A_2 & B_2 \\ 0 & C_2 & D_2 \end{bmatrix} \begin{bmatrix} A_1 & 0 & B_1 \\ 0 & I & 0 \\ C_1 & 0 & D_1 \end{bmatrix}$$

$$A = \begin{bmatrix} A_1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ B_2C_1 & A_2 & 0 & \cdots & 0 & 0 & 0 \\ B_3C_1D & B_3C_2 & A_3 & \cdots & 0 & 0 & 0 \\ B_4C_1D^2 & B_4C_2D & B_4C_3 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ B_{M-1}C_1D^{M-3} & B_{M-1}C_2D^{M-4} & B_{M-1}C_1D^{M-5} & \cdots & B_MC_{M-2} & A_{M-1} & 0 \\ B_MC_1D^{M-2} & B_MC_2D^{M-3} & B_MC_3D^{M-4} & \cdots & B_MC_{M-2}D & B_MC_{M-1} & A_M \end{bmatrix}$$

$$B = \begin{bmatrix} B_1 \\ B_2D \\ B_3D^{(2)} \\ \vdots \\ B_MD^{(M-1)} \end{bmatrix} \qquad C = \begin{bmatrix} C_1D^{M-1} & C_2D^{M-2} & \cdots & C_{M-2}D^2 & C_{M-1}D & C_M \end{bmatrix}$$

$$D = \prod_{i=1}^{M} D_i$$

Avec : $M = N / 2$ (N pair)

$D_i = (b_0)^{1/M}$ $i = 1,\ldots,M$

$D^j = D_1D_2\ldots D_j$ $j = 0,\ldots,M$

Fig 5.6: Structure cascade bloc (A,B,C,D)

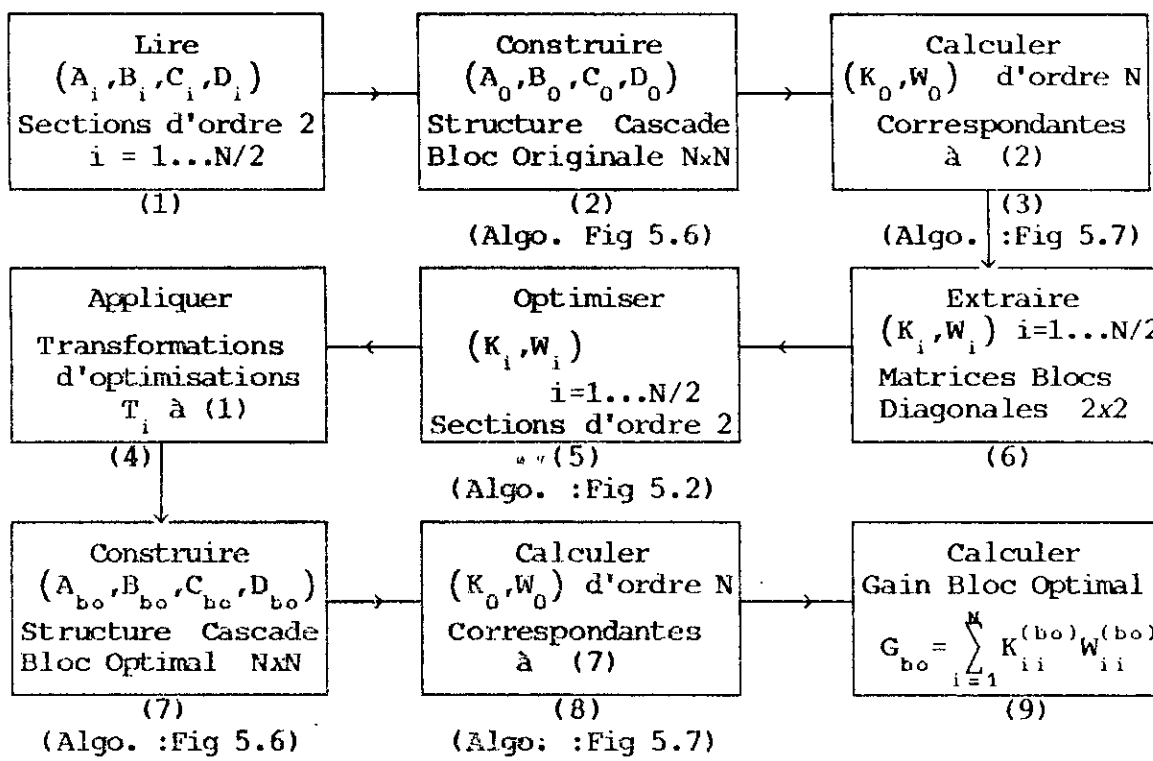


Fig 5.8: Organigramme d' une structure cascade de bloc optimal

le résultat de la structure bloc optimale est le suivant:

$$\left[\begin{array}{c|c} A_{min} & B_{min} \\ \hline C_{min} & D_{min} \end{array}\right] = \begin{bmatrix} T_1^{-1} & 0 & 0 \\ 0 & T_2^{-1} & 0 \\ 0 & C & 1 \end{bmatrix} \begin{bmatrix} A_1 & 0 & B_1 \\ B_2C_1 & A_2 & B_2D_1 \\ D_2C_1 & C_2 & D_2D_1 \end{bmatrix} \begin{bmatrix} T_1 & 0 & 0 \\ 0 & T_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (5.22a)$$

$$= \begin{bmatrix} I & 0 & 0 \\ 0 & T_2^{-1}A_2T_2 & T_2^{-1}B_2 \\ 0 & C_2T_2 & D_2 \end{bmatrix} \begin{bmatrix} T_1^{-1}A_1T_1 & 0 & T_1^{-1}B_1 \\ 0 & I & 0 \\ C_1T_1 & 0 & D_1 \end{bmatrix} \qquad (5.22b)$$

Dans l'équation (5.22a), les transformations individuelles d'optimisation sont appliquées en forme de bloc diagonale afin de préserver la structure cascade en bloc. Ce résultat est équivalent aux transformations d'optimisation, qui donnent les structures $(T_i^{-1}A_iT_i,\ T_i^{-1}B_i,\ C_iT_i,\ D_i)$, $i=1,\ldots,N/2$, séparément comme le montre l'équation (5.22b). Par ces différentes structures optimales, on forme la structure bloc optimal $(A_{bo},B_{bo},C_{bo},D_{bo})$ d'ordre N, en utilisant la structure de la figure (5.6). Enfin, on résume les différentes étapes de la réalisation de la structure cascade de bloc optimal dans un organigramme présenté par la figure (5.8).

Exemple numérique

Avec le même exemple , la même décomposition de la fonction de transfert et à partir des deux sections canoniques de second ordre (Tab 5.2), on déduit la structure bloc original d'ordre 4 (Fig 5.6), à savoir:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -.74744 & 1.72595 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ .00141 & .02082 & -.88703 & 1.86380 \end{bmatrix} ; \quad B = \begin{bmatrix} 0 \\ 1 \\ 0 \\ .00558 \end{bmatrix}$$

$$C = \begin{bmatrix} .00001 & .00012 & .00063 & .02159 \end{bmatrix} ; \quad D = .00003$$

Les matrices originales K et W correspondantes à cette structure sont données par:

$$K = \begin{bmatrix} 92.59321 & 91.45323 & 62.61736 & 67.95149 \\ 91.45323 & 92.59321 & 57.28323 & 62.61736 \\ 62.61736 & 57.28323 & 103.12894 & 102.60112 \\ 67.95149 & 62.61736 & 102.60112 & 103.12894 \end{bmatrix}$$

$$W = \begin{bmatrix} .02841 & -.03784 & .01874 & -.01673 \\ -.03784 & .05096 & -.03046 & .02845 \\ .01874 & -.03046 & .07446 & -.08290 \\ -.01673 & .02845 & -.08290 & .09463 \end{bmatrix}$$

Ainsi, le gain du bruit de calcul de la structure cascade bloc original est donné par:

$$G_{original} = \sum_{i=1}^{4} K_{ii} W_{ii} = 24.78883$$

En extrant les blocs diagonaux (2x2) des matrices K et W, on obtient des cellules d'ordre 2 qu'on minimise séparément (Tab 5.3).

Tab 5.3: Minimisation de deux çellules de second ordre extraites de la structure cascade bloc original

	Çellule 1	Çellule 2
K	$\begin{bmatrix} 92.59321 & 91.45323 \\ 91.45323 & 92.59321 \end{bmatrix}$	$\begin{bmatrix} 103.12894 & 102.60112 \\ 102.60112 & 103.12894 \end{bmatrix}$
W	$\begin{bmatrix} .02841 & -.03784 \\ -.03784 & .05096 \end{bmatrix}$	$\begin{bmatrix} .07446 & -.08290 \\ -.08290 & .09463 \end{bmatrix}$
Gain	7.34948	17.43934
T	$\begin{bmatrix} -2.71789 & -11.52776 \\ -.40362 & -9.92611 \end{bmatrix}$	$\begin{bmatrix} 1.63542 & -9.29001 \\ 2.77474 & -8.56179 \end{bmatrix}$
modes d'ordre 2	.08787 ; .64734	.22375 ; .61376
A_{optm}	$\begin{bmatrix} .86871 & .01630 \\ -.16980 & .85721 \end{bmatrix}$	$\begin{bmatrix} .91805 & .13706 \\ -.13706 & .94574 \end{bmatrix}$
B_{optm}	$\begin{bmatrix} .51635 \\ -.12174 \end{bmatrix}$	$\begin{bmatrix} .78894 \\ .13886 \end{bmatrix}$
C_{optm}	$\begin{bmatrix} -.01224 & -.22298 \end{bmatrix}$	$\begin{bmatrix} .06095 & -.19076 \end{bmatrix}$
D_{optm}	.00558	.00558
$Gain_{optm}$	0.27027	0.35072

Par conséquent, la structure bloc optimal est donnée par:

$$A_{bo}=\begin{bmatrix} .86871 & .01630 & 0 & 0 \\ -.16980 & .85721 & 0 & 0 \\ -.00965 & -.17592 & .91805 & .13706 \\ -.00170 & -.03096 & -.13706 & -.94574 \end{bmatrix} ; \quad B_{bo}=\begin{bmatrix} .51635 \\ -.12174 \\ .00440 \\ -.00077 \end{bmatrix}$$

$$C_{bo}=\begin{bmatrix} -.00007 & -.00124 & .06095 & -.19076 \end{bmatrix} ; \quad D_{bo}= .00003$$

Les matrices K_{bo} et W_{bo} correspondantes sont données par:

$$K_{bo}=\begin{bmatrix} 1 & -.76095 & .40939 & -.11500 \\ -.76095 & 1 & -.79751 & .48841 \\ .40939 & -.79751 & 1 & -.46566 \\ -.11500 & .48841 & -.46566 & 1 \end{bmatrix}$$

$$W_{bo}=\begin{bmatrix} .13513 & -.10283 & .03112 & .06798 \\ -.10283 & .13513 & -.10723 & -.03526 \\ .03112 & -.10723 & .17536 & -.08166 \\ .06798 & -.03526 & -.08166 & .17536 \end{bmatrix}$$

d'où le gain du bruit de calcul de la structure bloc optimal:

$$G_{bo} = \sum_{i=1}^{4} K_{ii}^{(bo)} \; W_{ii}^{(bo)} = 0.62099$$

On note bien, que tous les éléments diagonaux de la matrice K sont égaux à 1, aussi la différence de gains entre cette structure et celle de sections optimales est faible.

5.5 Performances d'une structure décomposée optimale

D'une manière similaire, on développe dans cette section les mêmes critères de performances vus pour la structure optimale.

Gain du bruit de calcul

Avec le même filtre passe-bas de Butterworth d'ordre 10 utilisé au chapitre IV, on obtient la figure (Fig 5.9) qui montre l'invariance du gain de bruit de la structure décomposée parallèle et cascade respectivement, par contre celui de la structure décomposée canonique (parallèle et cascade) varie avec la fréquence de coupure, et il est assez important pour les fréquences faibles par

rapport à la structure canonique globale (Fig 4.3).

Le tableau (Tab 5.4) présente la variation du gain de bruit du filtre considéré des différentes structures décomposées, en fonction de l'ordre du filtre. A titre comparatif avec le tableau (Tab 4.1), on en déduit la série des gains de bruit de calcul d'un filtre prototype, classés par ordre décroissant:

$$\left\{ G_{\text{can global}};\ G_{\text{dec can paral}};\ G_{\text{dec optm paral}};\ G_{\text{dec can casca}};\ G_{\text{dec optm casca}};\ G_{\text{optm global}} \right\} \qquad (5.23)$$

Tab 5.4: Gain de bruit de calcul d'un filtre passe-bas de Butterworth pour différentes structures d'état décomposées.

(fréquence de coupure $\pi/4$)

N / Gain	4	6	8	10	12
Can. Paral.	3.41424	14.47701	83.02936	564.51865	4233.52343
Optm. Paral.	1.28033	4.93440	25.77941	166.59017	1211.31653
Can. Casca.	1.34721	2.20685	3.34356	4.95160	7.34363
Bloc optm Casca.	0.62094	0.93202	1.33910	1.90665	2.73214
Sect optm Casca.	0.64011	0.96655	1.39867	2.01184	2.91910

Qualité du filtrage avec un nombre de bits limité

L'opération du filtrage sur le signal sinusoïdal utilisé dans le chapitre précédent (cf. section 4.6), est refaite dans cette partie en utilisant les structures décomposées parallèle et cascade . Aussi, on fait varier la longueur des mots binaires pour chaque type de structure. Par conséquent, On obtient les figures (Fig 5.10 et 5.11) correspondantes à la structure décomposée optimale parallèle et cascade respectivement, sur lesquelles on constate que la sinusoïde de sortie est conservée bien avant 10 bits, et une nette amélioration de cette dernière par rapport au cas

global optimal (Fig 4.8). D'autre part, sur les figures (Fig 5.12 et 5.13), à partir de 8 bits le signal n'est plus distordu relativement au cas global canonique (Fig 4.9).

Sensibilité du filtre avec un nombre de bits limité

Pour différentes valeurs de longueur du registre, on trace la réponse fréquentielle du filtre considéré pour les structures décomposées. On remarque pour les cas décomposés optimal parallèle et optimal cascade (Fig 5.14 et 5.15), respectivement, la distorsion de la courbe disparaît à partir de 6 bits donc, une amélioration par rapport au cas global optimal (Fig 4.10). Pour le cas décomposé canonique (Fig 5.16 et 5.17), l'amélioration apparait à partir 8 bits alors que le cas global canonique (4.11) c'est à partir de 14 bits.

5.6 Conclusion

La théorie de la décomposition du filtre en cellules de second ordre [4] connectées en parallèle ou en cascade a permis d'étudier le compromis entre le gain de bruit de calcul et la complexité de réalisation. Les procédures d'optimisation utilisées par Hwang [1], Bomar [3] et Barnes [4] aboutissent à des sections de second ordre à gain de bruit minimum, ce qui réduit le nombre de multiplications pour un filtre d'ordre N à (4N+1). Cependant, le gain de bruit d'une telle structure est supérieur à celui de la structure globale optimale. D'autre part, le nombre de multiplications est de (2N+1) pour la structure globale canonique, il en est de même pour la décomposée canonique mais de gain en bruit de calcul beaucoup plus faible, ce qui rend ce type de structure, ainsi que la décomposée optimale beaucoup plus intéressant, à savoir avec les structures décomposées, la complexité du filtre est énormément simplifiée en revanche d'une légère augmentation du gain de bruit de calcul. Toutefois ce dernier peut être amélioré de 6 db en ajoutant un bit à la longueur du mot, donc l'augmentation du gain de bruit peut être compensée par un prix très faible d'où l'intérêt d'utilisation de ces structures.

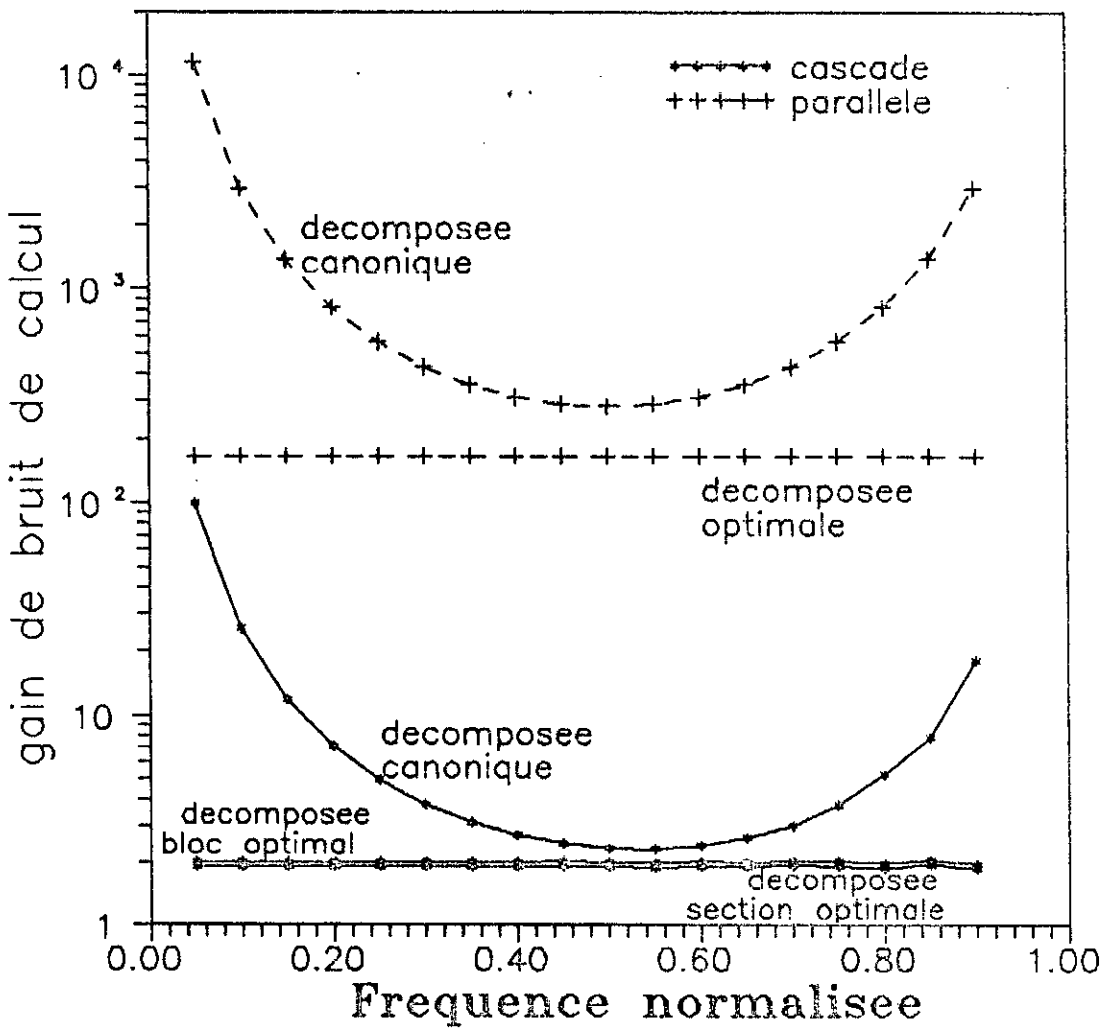


Fig.5.9: Gain de bruit de calcul d'un filtre pass-bas de Butterworth d'ordre 10 realise en structure decomposee optimale et canonique pour les cas parallele et cascade.

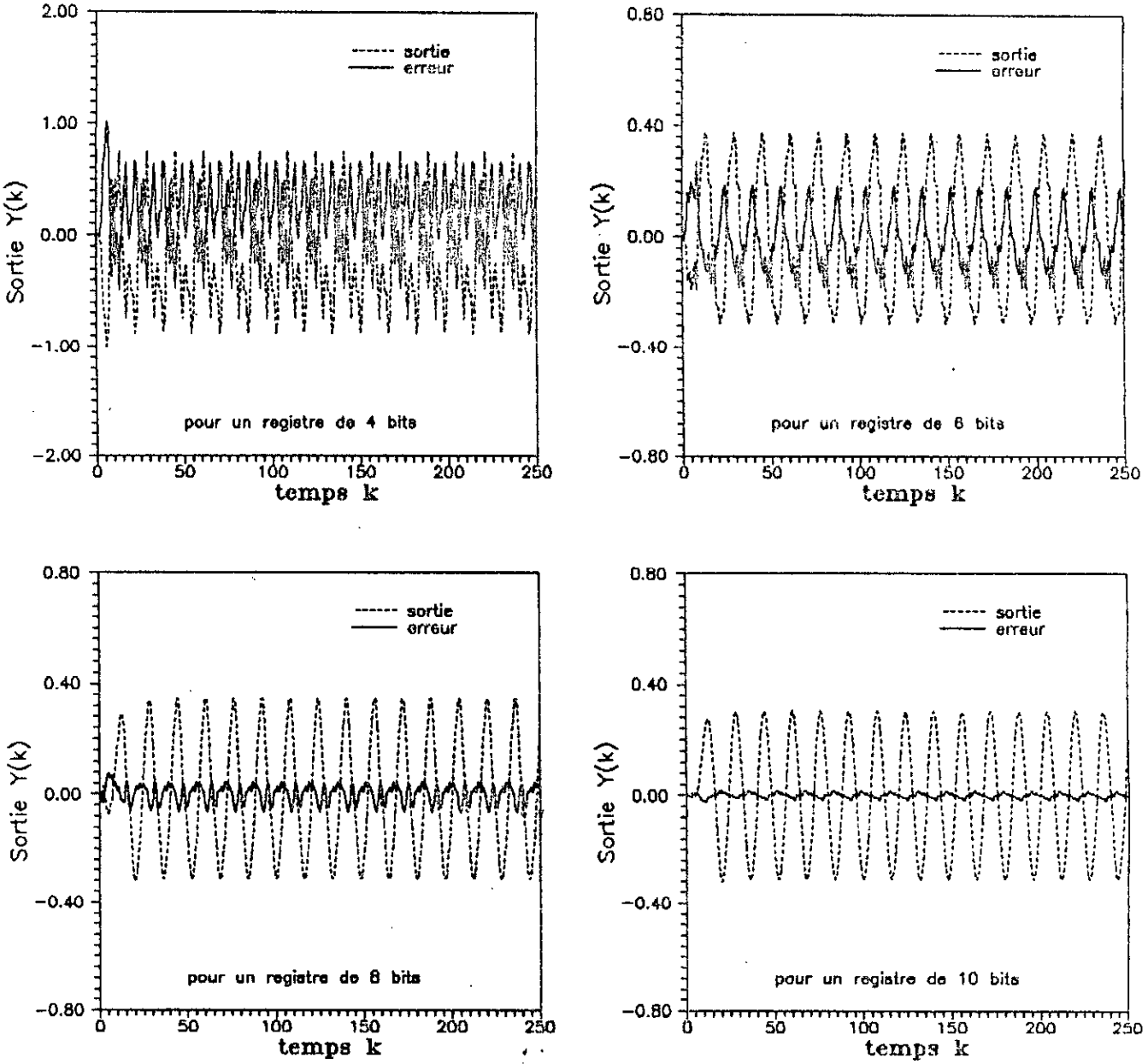


Fig 5.10: Signal sinusoïdal appliqué à un filtre passe-bas de Butterworth d'ordre 10 réalisé par la structure décomposée parallèle optimale en utilisant des registres de stockages de longueurs finies de bits.

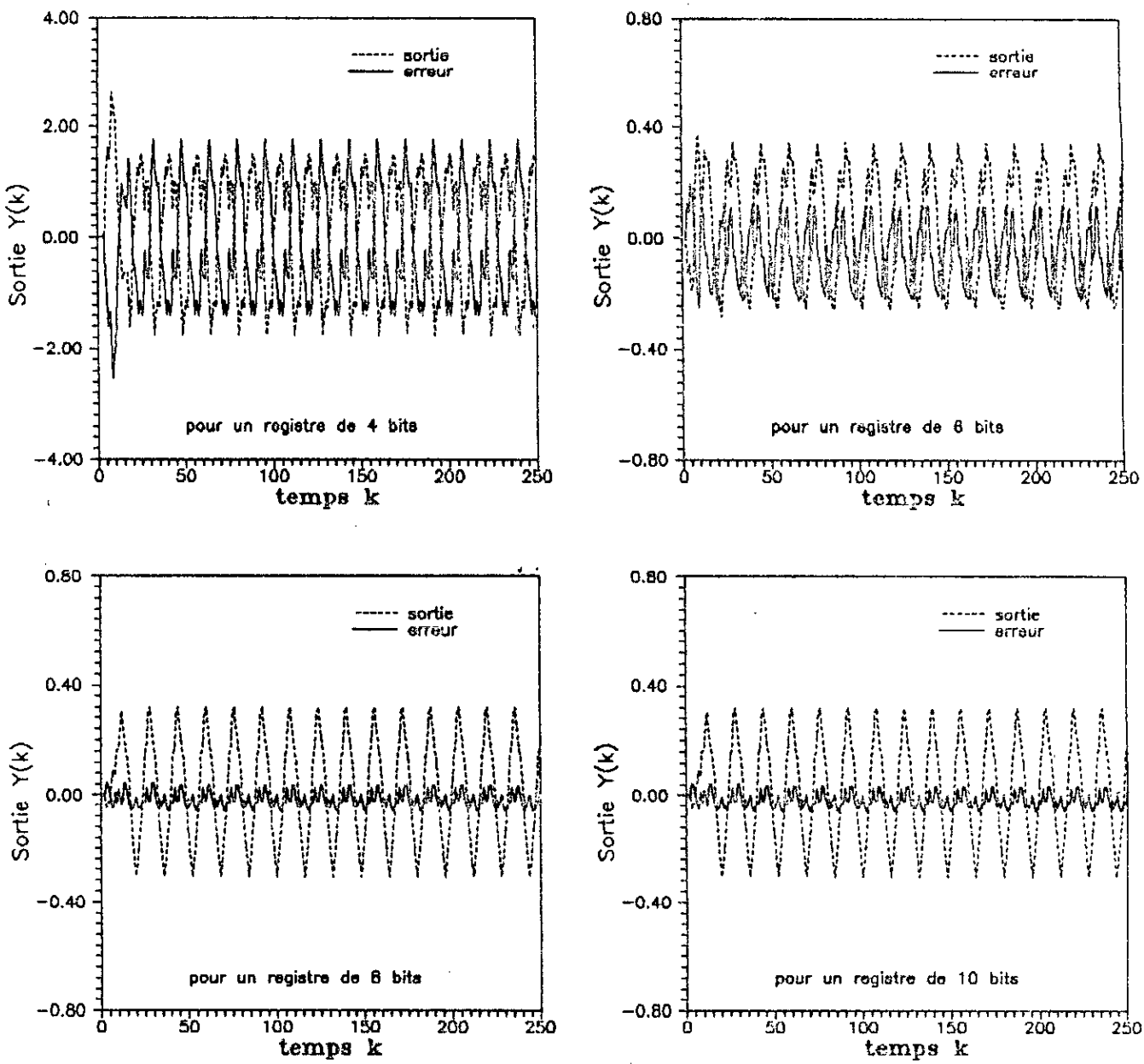


Fig 5.11: Signal sinusoïdal appliqué à un filtre passe-bas de Butterworth d'ordre 10 réalisé par la structure décomposée parallèle canonique en utilisant des registres de stockages de longueurs finies de bits.

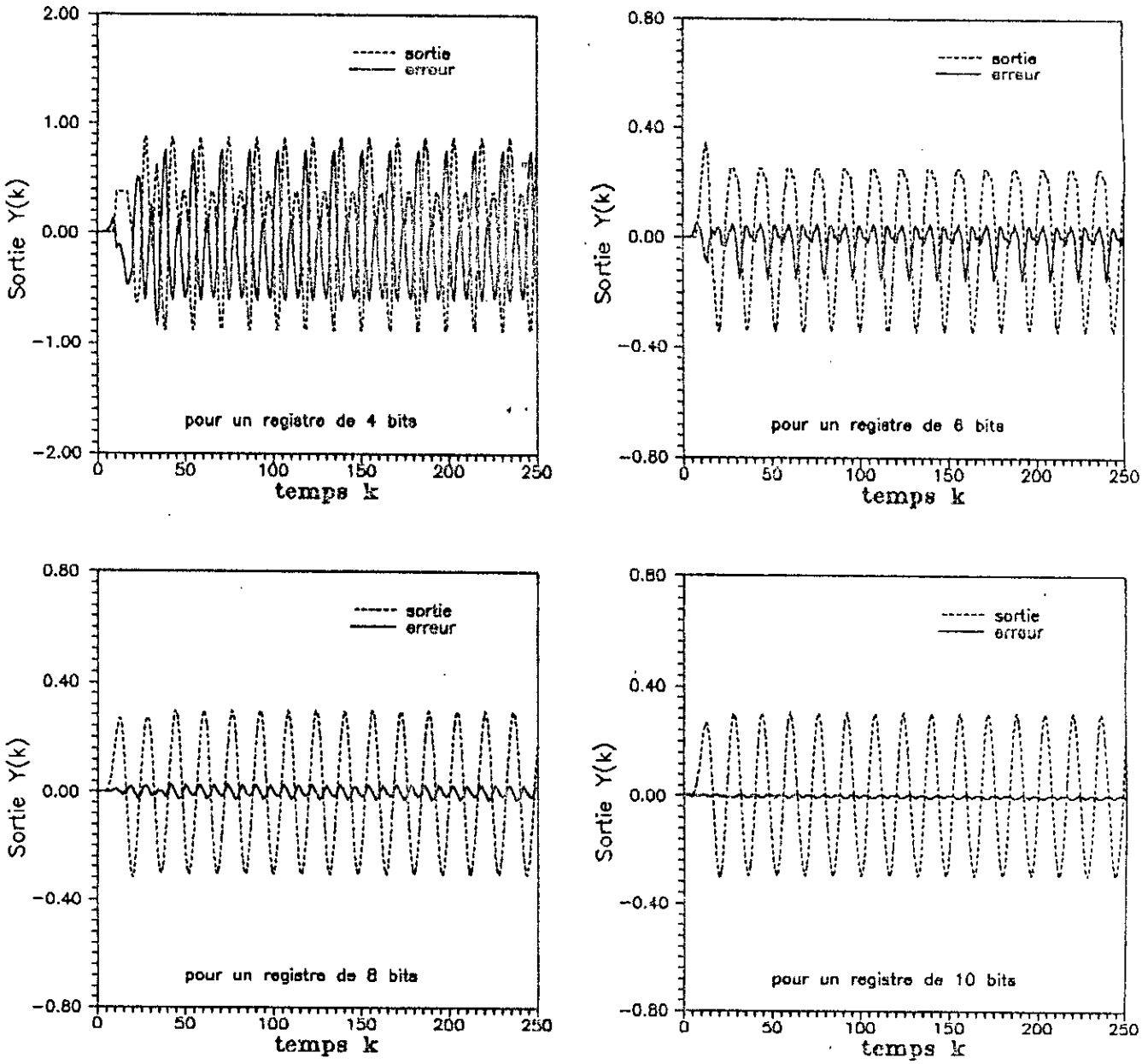


Fig 5.12: Signal sinusoïdal appliqué à un filtre passe-bas de Butterworth d'ordre 10 réalisé par la structure décomposée cascade optimale en utilisant des registres de stockages de longueurs finies de bits.

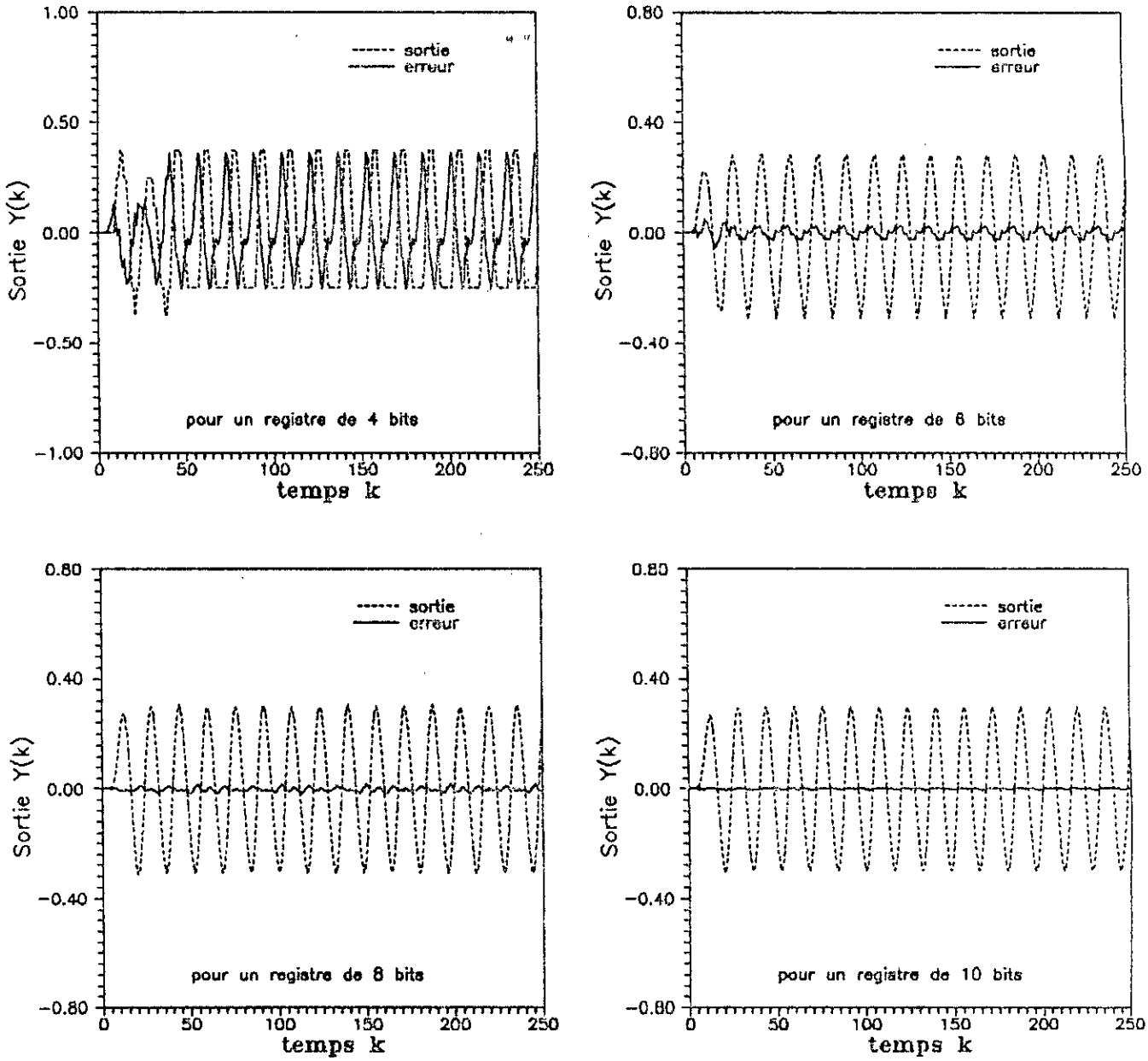


Fig 5.13: Signal sinusoïdal appliqué à un filtre passe-bas de Butterworth d'ordre 10 réalisé par la structure décomposée cascade canonique en utilisant des registres de stockages de longueurs finies de bits.

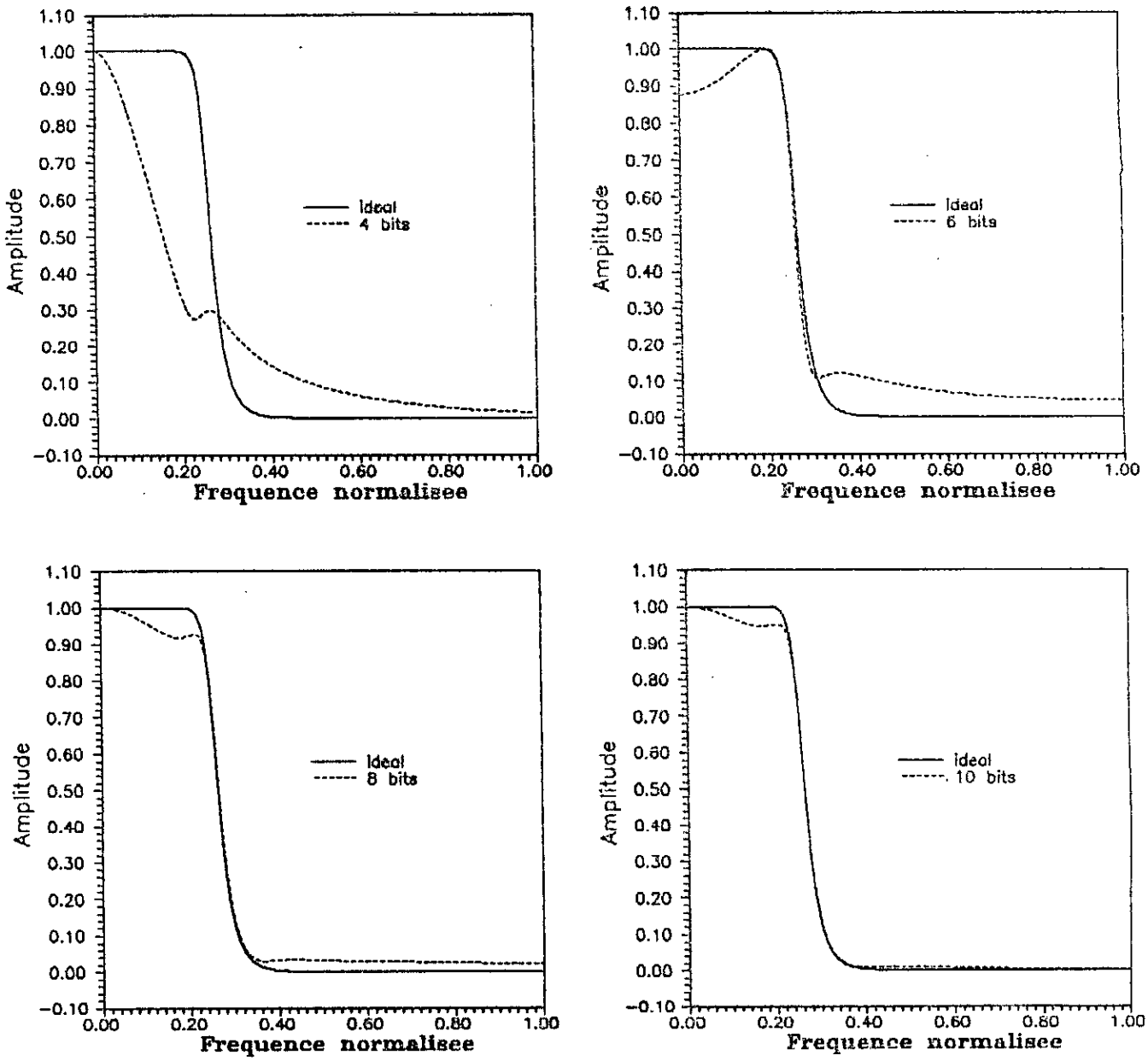


Fig 5.14: Spectre d'amplitude d'un filtre passe-bas de Butterworth d'ordre 10 réalisé par la structure décomposée parallèle optimale en utilisant des registres de stockages de longueurs finies de bits.

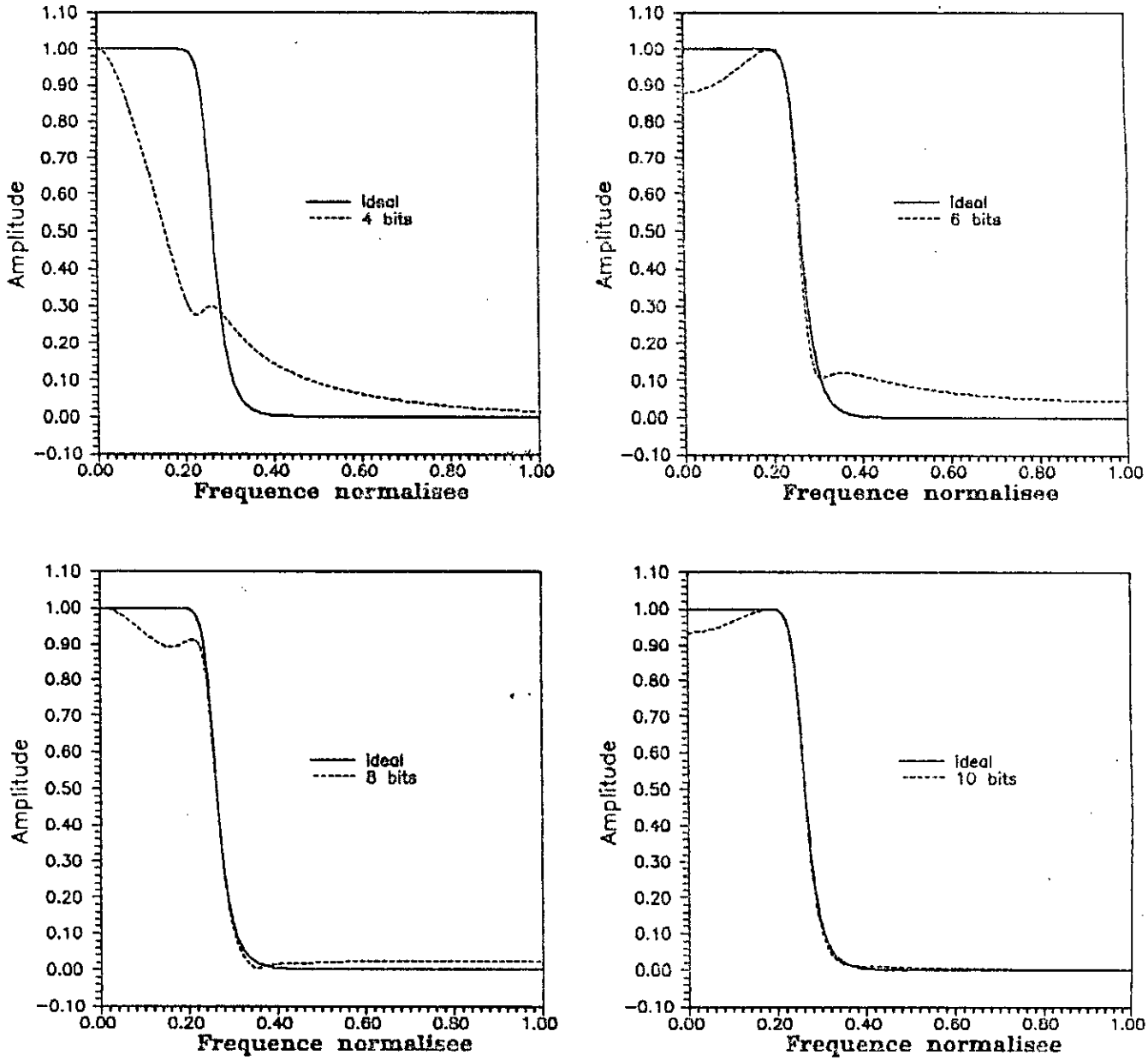


Fig 5.15: Spectre d'amplitude d'un filtre passe-bas de Butterworth d'ordre 10 réalisé par la structure décomposée parallèle canonique en utilisant des registres de stockages de longueurs finies de bits.

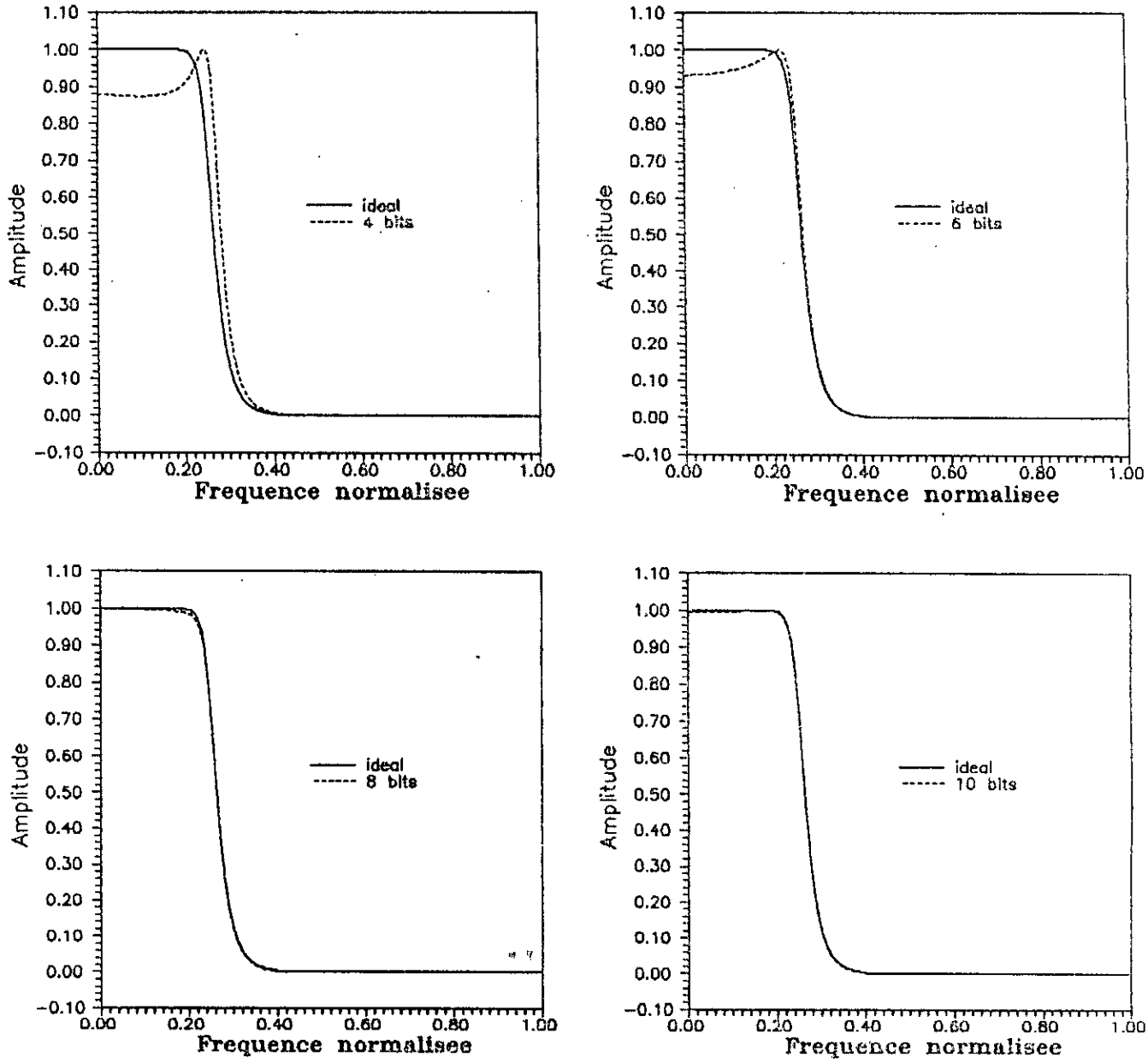


Fig 5.16: Spectre d'amplitude d'un filtre passe-bas de Butterworth d'ordre 10 réalisé par la structure décomposée cascade optimale en utilisant des registres de stockages de longueurs finies de bits.

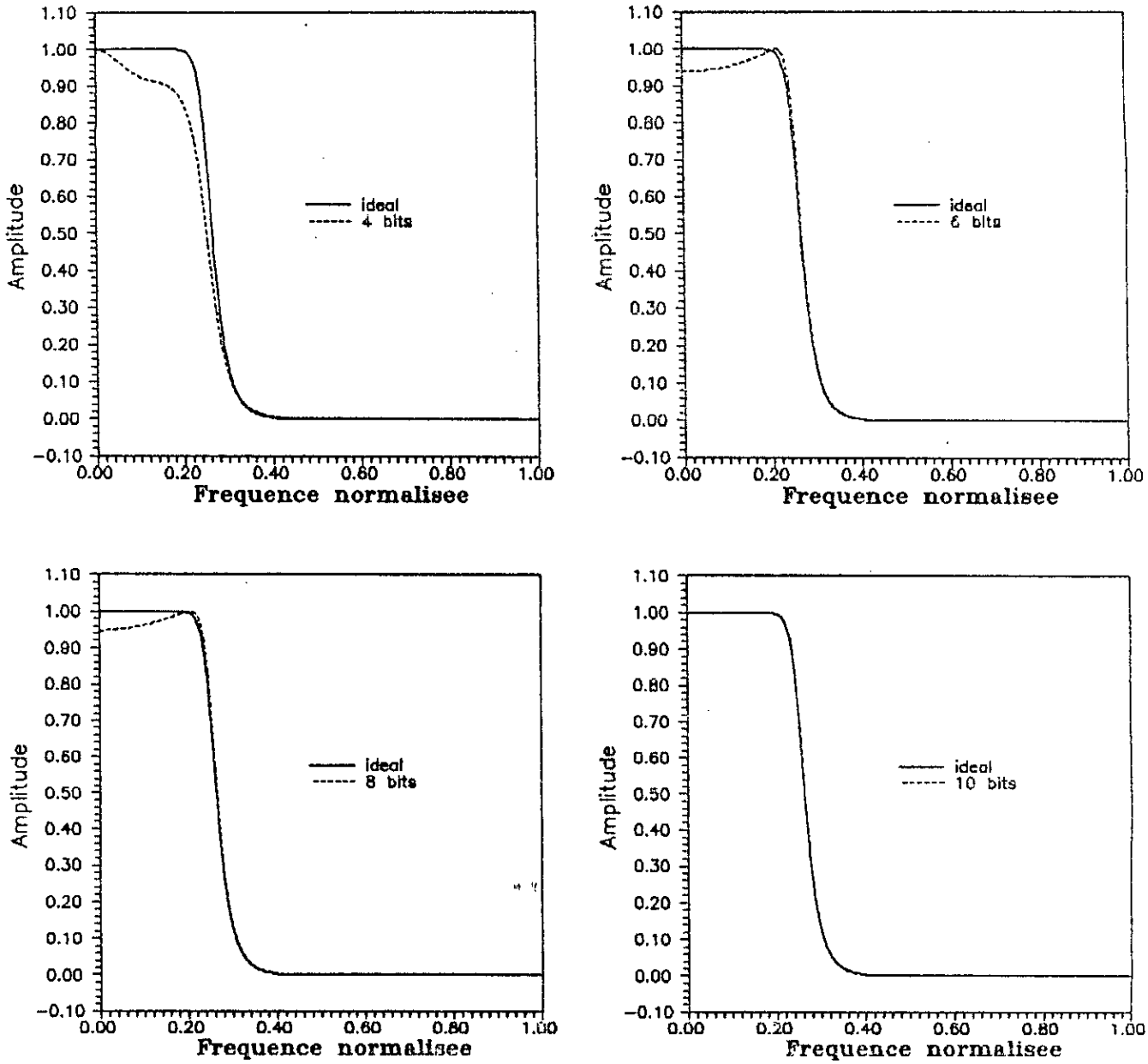


Fig 5.17: Spectre d'amplitude d'un filtre passe-bas de Butterworth d'ordre 10 réalisé par la structure décomposée cascade canonique en utilisant des registres de stockages de longueurs finies de bits.

CHAPITRE VI

CONCLUSION GENERALE

CHAPITRE VI

CONCLUSION GENERALE

Une étude sur les filtres numérique récursifs (R.I.I) représentés par la structure d'espace d'état est établie dans ce travail afin de déduire précisément l'opération interne du filtrage, du fait que ces filtres présentent des réalisations efficaces et leur représentation en espace d'état permet de décrire leur comportement interne (cf. Chap.II). De plus, grâce à la propriété de changement de coordonnées dans une telle représentation, par une simple transformation non singulière on peut choisir la meilleure représentation vis-à-vis de multiple erreurs de mise en oeuvre d'un filtre numérique (cf. Chap.III).

Le choix et la simulation des structures des filtres numériques synthétisés avant leur implantation permettent de résoudre les problèmes que peut poser leur réalisation matérielle; en plus des problèmes classiques: échantillonnage, facteur de qualité, binarisation, représentation arithmétique, ... il y'a le problème de la précision avec la quelle doit s'opérer le filtrage à cause de l'utilisation de registres de tailles réelles pour représenter les valeurs des paramètres des filtres et des signaux qui les parcourent ainsi que les valeurs résultant des diverses opérations arithmétiques, d'où l'obligation de tenir compte des effets néfastes des erreurs de calcul, ainsi est donc l'intérêt primordial de notre étude: Minimiser davantage ses erreurs de calculs.

Dans ce travail, la simulation des filtres numériques est faite sur la base d'une représentation des nombres binaires en arithmétique à virgule fixe. Pour une telle réalisation, on est amené à optimiser les effets non linéaires introduits par la quantification des coefficients et des résultats des opérations arithmétiques en utilisant des registres de stockage à longueur de mot de bit bien finie. Par conséquent, ces effets affectent la performance du filtre, d'où la nécessitée de trouver une structure

dont l'ensemble de ses erreurs de calculs est minimisé (cf. Chap. IV).

La méthode d'approche utilisée dans notre étude, pour l'obtention d'une structure à gain de bruit minimal, est celle de S. Hwang [1]. Sous une contrainte de normalisation convenablement choisie, cette approche permet de rendre les dépassements peu probables et l'augmentation du bruit de calcul qui en découle négligeable vis-à-vis de la structure canonique (forme directe). L'inconvénient majeur des structures à faible bruit de calcul réside dans la complexité de leurs réalisations matérielles due au grand nombre de multiplications qu'elles nécessitent; soit $(N+1)^2$ pour un filtre d'ordre N, alors qu'il n'est que de (2N+1) pour les structures canoniques. Cependant, on a attaché une grande importance au choix définitif d'un ensemble de compromis entre coefficients et erreurs guidées par la structure optimale. En particulier, pour obtenir une rapidité de calcul satisfaisante, on fait réduire le nombre d'opérations, tout en tenant compte de la capacité des registres utilisés et aux sources de bruits internes et externes (c-à-d en maintenant un niveau de bruit faible), d'où le recours aux structures décomposées en sections d'ordre 2 (cf. Chap.V), du fait que:

* Le problème de minimisation d'une section de second ordre présente une facilité de procédure du point de vue traitement mathématique par rapport au cas général d'ordre N.

* Le nombre d'opérations arithmétiques est réduit à (4N+1) par rapport à la structure optimale.

* La faible sensibilité aux arrondis des coefficients et du bruit de calcul, malgré que ce dernier augmente légèrement par rapport à celui d'une structure optimale globale.

En utilisant une procédure d'optimisation des filtres de second ordre, soit celle de Hwang [1] où celle de Bomar [3], on peut obtenir une structure décomposée optimale, soit en parallèle ou en cascade (bloc optimal ou sections optimales) tel qu'il est décrit dans le chapitre V, avec un gain de bruit de calcul minimal, mais pas assez supérieur à celui de la structure globale optimale.

La simulation d'un filtre numérique de Butterworth nous a

permis de vérifier les résultats élaborés en théorie ainsi que la validité des hypothèses faites sur les erreurs de calcul pour l'ensemble des différentes structures utilisées dans cette étude. Par conséquent, on peut conclure que les structures décomposées optimales et canoniques présentent un certain équilibre de performances du point de vue erreurs de calcul et nombre d'opérations, donc un bon compromis qui se situe entre les deux structures extrêmes à savoir la canonique et l'optimale globale. D'autre part avec les structures décomposées, l'augmentation légère du gain de bruit de calcul par rapport aux structures optimales peut être compensée en ajoutant quelques bits à la longueur du mot, du fait que l'élargissement de cette dernière de 1 bit correspond à une amélioration du gain de 6 db, donc c'est un effet indésiré qui peut être compensé par un prix très faible.

Aussi, la réalisation d'un filtre numérique récursif en virgule fixe n'est pas unique, elle dépend de la structure d'état qui doit être choisie de façon à répondre au cahier de charges exigé pour une application donnée, à savoir le coût de réalisations vu l'aspect économique (nombre de multiplieurs nécessaires), et la qualité du filtrage vu l'aspect technique (tolérances en bruit de calcul et rapport signal/bruit désiré).

BIBLIOGRAPHIE

BIBLIOGRAPHIE

[1] S. Y. Hwang, "Minimum uncorrelated unit noise in state-space digital filtering", IEEE Trans. Acoust. Speech, Signal Processing, vol. ASSP-25, pp. 273-281, Aug. 1977.

[2] C. T. Mullis and R. A. Roberts, "Synthesis of minimum roundoff noise fixed point digital filters", IEEE Trans. Circuits Syst., vol. CAS-23, pp. 551-562, Sept. 1976.

[3] B. W. Bomar, "New second-order state-space structures for realizing low roundoff noise digital filters", IEEE Trans. Acoust., Speech., Signal Processing, vol. ASSP-33, pp. 106-110 Feb. 1985.

[4] C. W. Barnes, "On the design of optimal state-space realizations of second-order digital filters", IEEE Trans. Circuits Syst., vol. CAS-31, pp.602-608, July 1984.

[5] M. Kunt, Traitement Numérique des Signaux, Edition Dunod, 1981.

[6] M. Bellanger, Traitement Numérique du Signal, Edition Masson, 1981.

[7] A. Antoniou, Digital Filters: Analysis and Design, McGraw-Hill, New York, 1979.

[8] C. T. Mullis and R. A. Roberts, Digital Signal Processing, Addison Wesley, Readings MA, 1987.

[9] L. R. Rabiner and B. Gold, Theory and Applications Of Digital Signal Processing, Prentice Hall, Englewood Cliffs, New Jersey. 1975.

[10] A. C. M. Claasen et J. B. H. Peek and F. G. Meckllenbrauker, "Effects of quantization and overflow in recursive digital filters", IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-24, pp. 517-529, Dec. 1976.

[11] V. Tavsanaglu and L. Thiele, "Optimal design of state-space digital filters by simultaneous minimization of sensitivity and roundoff noise", IEEE Trans. Circuits Syst., vol. CAS-31, pp. 884-888, Oct. 1984.

[12] C. T. Mullis and R. A. Roberts, "Roundoff noise in digital filters: Frequency transformations and invariants", IEEE Trans. Acoust., Speech., Signal Processing, vol. ASSP-24, pp. 538-550, Dec. 1976.

[13] L. B. Jackson, A. G. Lindgren and Y. Kim, "Optimal synthesis of second-order state-space structures for digital filters", IEE Trans. Circuit Syst., vol. CAS-26, pp. 149-153, March 1979.

[14] C. W. Barnes, "Roundoff noise and overflow in normal digital filters", IEEE Trans. Circuits Syst., vol. CAS-26, pp. 154-159, March 1979.

[15] C. W. Barnes, "Computationally efficient second-order digital filters sections with low roundoff noise gain", IEEE Trans. Circuits Syst., vol. CAS-31, pp. 841-847, Oct. 1984.

[16] A. Reverchon et M. Ducamp, Mathématique sur micro-ordinateur, Eyrolles, Paris 1986.

[17] A. V. Oppenheim and R. W. Schafer, Digital Signal Processing, Englewood Cliffs, Prentice Hall, 1975.

[18] R. Bellman, Introduction to Matrix Analysis, McGraw-Hill, New york, 1970.

[19] B. Derras, "New efficient state-space structures for the realization of recursive digital filters", Master Thesis, Department of Electrical and Computer Engineering, University of Colorado 1985.

[20] M. Arjmand and R. A. Roberts, "Reduced multiplier, low roundoff noise digital filters", IEEE Proc. of ICASSP, 1979.

[21] M. Kawamata and T. Higuichi, "A unifed approach to the optimal synthesis of fixed-point state-space digital filters", IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-33, pp.911-920, Aug. 1985.

[22] C. T. Mullis , R. A. Roberts and W. L. Mills, "Normal realizations of IIR digital filters", IEEE Proc. of ICASSP, 1979.

[23] B. W. Bomar and J. C. Hung, "Minimum roundoff noise digital filters with some power-of-two coefficients", IEEE Trans. Circuits Syst., vol. CAS-34, pp. 833-840, Oct. 1984.

[24] M. Labarrere, J. P. Krief et B. Gimonet, Le Filtrage et ses applications, Edition Cepadues, 1982.

ANNEXES

ANNEXE A

METHODE DE CHOLESKY

Pour une matrice A définie positive symétrique, on peut toujours la décomposer sous la forme suivante [18]:

$$A = S\,S^{T} \tag{A.1}$$

où S est une matrice triangulaire inférieure.

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & & & \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{bmatrix} = \begin{bmatrix} s_{11} & 0 & \cdots & 0 \\ s_{21} & s_{22} & \cdots & 0 \\ \vdots & & & \\ s_{N1} & s_{N2} & \cdots & s_{NN} \end{bmatrix} \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1N} \\ 0 & s_{22} & \cdots & s_{2N} \\ \vdots & & & \\ 0 & 0 & \cdots & s_{NN} \end{bmatrix}$$

où les coefficients s_{ij} sont donnés par:

$$s_{ii} = \left[a_{ii} - \sum_{j=1}^{i-1} s_{ij}^{2} \right]^{1/2} \qquad i = 1\ldots,N \tag{A.2}$$

$$s_{ji} = \begin{cases} 0 & i > j \\ \dfrac{1}{s_{ii}} \left[a_{ij} - \sum_{k=1}^{i-1} s_{jk}\, s_{ik} \right] & j = i+1, i+2, \ldots, N \end{cases} \tag{A.3}$$

Cette méthode permet de calculer la matrice de transformation T_0 de normalisation, donnée par l'équation (4.41).

ANNEXE B

METHODE DE CALCUL D'UNE MATRICE ORTHOGONALE DE TRANSFORMATION D'ETAT OPTIMALE

Pour une structure d'état optimale, la matrice de transformation d'optimalisée (4.49) est donnée par:

$$T = T_0 \; R_1 \; P_{optm} \; R_0^T \qquad (B.1)$$

Afin de calculer la matrice orthogonale R_0, on considère une matrice M diagonale d'ordre N, telle que les éléments diagonaux positifs μ_i^2 vérifient:

$$\sum_{i=1}^{N} \mu_i^2 = N \qquad \text{pour tout } i \qquad (B.2)$$

alors, il existe une matrice orthogonale R_0 telle que, tous les éléments diagonaux de la matrice définie positive

$$K = R_0 \; M \; R_0^T \qquad (B.3)$$

sont égaux à l'unité, avec R_0 donnée par le produit suivant 1 :

$$R_0 = R_N \; R_{N-1} \cdot \cdot \cdot R_2 \qquad (B.4)$$

où les éléments de la matrice orthogonale R_i, i = 2,...,N ont la forme suivante:

$$R_i = \begin{bmatrix} I & 0 & 0 \\ \cdots\cdots & \cos(\psi_i) \cdots\cdots & \sin(\psi_i) \cdots\cdots \\ 0 & I & 0 \\ \cdots\cdots & -\sin(\psi_i) \cdots\cdots & \cos(\psi_i) \cdots\cdots \\ 0 & 0 & I \end{bmatrix} \qquad i = 2,...N \qquad (B.5)$$

Le calcul des $\cos(\psi_i)$ et $\sin(\psi_i)$ est donné par un processus récurrent, qui permet de calculer en premier lieu la matrice R_0, en l'explicitant de la manière suivante:

$$M_1 = R_2\, M\, R_2^T =$$

$$= \left[\begin{array}{cc:c} \cos(\psi_2) & \sin(\psi_2) & 0 \\ -\sin(\psi_2) & \cos(\psi_2) & \\ \hdashline 0 & & I \end{array}\right] \left[\begin{array}{cc:c} \mu_1^2 & 0 & 0 \\ 0 & \mu_2^2 & \\ \hdashline 0 & & \mu_N^2 \end{array}\right] \left[\begin{array}{cc:c} \cos(\psi_2) & -\sin(\psi_2) & 0 \\ \sin(\psi_2) & \cos(\psi_2) & \\ \hdashline 0 & & I \end{array}\right]$$

Par suite de calcul, on trouve:

$$M_1 = \left[\begin{array}{cc:c} \mu_1^2\cos^2(\psi_2) + \mu_2^2\sin^2(\psi_2) & (\mu_2^2-\mu_1^2)\cos(\psi_2)\sin(\psi_2) & \\ (\mu_2^2-\mu_1^2)\cos(\psi_2)\sin(\psi_2) & \mu_2^2\cos^2(\psi_2) + \mu_1^2\sin^2(\psi_2) & 0 \\ \hdashline 0 & & \begin{array}{cc} \mu_3^2 & \\ & \mu_N^2 \end{array} \end{array}\right]$$

Par un choix convenable, les deux premiers éléments diagonaux de la matrice M_1 sont égaux à l'unité, par conséquent ce choix donne:

$$\cos(\psi_2) = \left[\frac{\mu_2^2 - 1}{\mu_2^2 - \mu_1^2}\right]^{1/2} \qquad (B.6)$$

et

$$\sin(\psi_2) = \left[\frac{1 - \mu_1^2}{\mu_2^2 - \mu_1^2}\right]^{1/2} \qquad (B.7)$$

Cependant, avec la condition (B.2) on aura:

$$\mu_1^2 < 1 \quad \text{et} \quad \mu_2^2 > 1$$

le cas où $\mu_i^2 = 1$, n'est pas considéré. Par suite en substituant (B.6) et (B.7) dans la matrice M_1, on obtient:

$$M_1 = R_2 M R_2^T = \left[\begin{array}{c:ccc} 1 & \sqrt{(1-\mu_1^2)(\mu_2^2-1)} & 0 \ \cdots & 0 \\ \hdashline \sqrt{(1-\mu_1^2)(\mu_2^2-1)} & \mu_2'^2 & 0 \ \cdots & 0 \\ 0 & 0 & \mu_3^2 & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & 0 \ \cdots & \mu_N^2 \end{array}\right] \qquad (B.8)$$

où $\mu_2'^2 = \mu_1^2 + \mu_2^2 - 1 > 0$. Des considérations similaires et des transformations peuvent être appliquées aux sous matrices diagonales $\left[(N-1)x(N-1)\right]$ de (B.8), aussi on continue le processus jusqu'à la dernière sous matrice [2x2] et obtenir enfin la matrice donnée par (B.3), sous la forme:

$$R_0 \, M \, R_0^T = \begin{bmatrix} 1 & & X \\ & \ddots & \\ X & & 1 \end{bmatrix} \qquad (B.9)$$

d'où la matrice de transformation d'optimalisée est:

$$T = T_0 \, R_1 \, P_{optm} \, R_N^T \, R_{N-1}^T \, . \, . \, . \, R_2^T \qquad (B.10)$$

Programme de synthèse des structures optimales

```
      PROGRAMME DE MINIMISATION
C     Ce programme permet d'obtenir une structure d'état d'un filtre
C     numérique prototype avec un gain de bruit de calcul minimum, a
C     partir d'une structure canonique.

C     Paramètres d'entrée:
C     N : Ordre du filtre numérique choisie.
C     Q : Vecteur des coefficients du numérateur.
C     P : Vecteur des coefficients du dénominateur.
C     Del : Facteur de normalisation ($).

C     Paramètres intermédiaires:
C     (A,B,C,D): Structure canonique; (K,W): Matrices correspondantes.
C      T : Matrice de transformation d'optimisation.
C     (Am,Bm,Cm,D): Structure optimale; (Km,Wm): Matrices correspondantes.

C     Paramètres de sortie:
C     Gc : Gain de la structure canonique.
C     Go : Gain de la structure optimale.

      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(50,50),B(50),C(50),W(50,50),P(51),Q(51)
      DIMENSION AM(50,50),BM(50),CM(50),WM(50,50),T(50,50)
      REAL*8 K(50,50),KM(50,50)
C
      OPEN (UNIT=17,FILE='COEF.DAT',STATUS='OLD')
C
      WRITE(*,*)' '
      WRITE(*,*)'Donner facteur de normalisation'
      READ(*,*)DEL
      WRITE(*,*)' '
      WRITE(*,*)'Donner ordre N'
      READ(*,*)N
      DO I=1,N+1
         READ(17,*)Q(I)
      END DO
      DO I=1,N+1
         READ(17,*)P(I)
      END DO
C
      Call ConstC (Q,P,A,B,C,N)
      D=Q(N+1)
      Iflag=0
      Call Calkw(A,B,K,N,Iflag)
      Iflag=1
      Call Calkw(A,C,W,N,Iflag)
      Call VARI (K,W,N,Gc)
      CALL MINO  (K,W,T,N,Go,Del)
      CALL CONSTM (T,A,B,C,AM,BM,CM,N)
      CALL WK (K,W,T,KM,WM,N)
      CALL VARI(KM,WM,N,Gm)

C
      WRITE(*,*)' '
      WRITE(*,*)' Gain structure canonique =',Gc
      WRITE(*,*)' Gain structure canonique =',Go
C
      STOP
      END
```

```
C*************************************************************************
      Subroutine Constc (Q,P,A,B,C,N)
C     Cette subroutine permet de construire la structure d'état canonique

C     Paramètres d'entrée:
C     Q,P : Vecteurs des coefficients du numérateur et dénominateur,
C           respectivement du filtre choisi.
C     N   : l'ordre du filtre.

C     Paramètres de sortie:
C     A,B,C : Matrice A (NxN) ; Vecteur B (Nx1) ; Vecteur C (1xN)

      Real*8 A(50,50),B(50),C(50),P(51),Q(51)
      Do 10 J=1,N
      A(N,J)=-1.D0*P(N+2-J)
 10   Continue
      Do 20 I=1,N-1
         A(I,I+1)=1.D0
 20   Continue
      Do 30 J=1,N
         C(J)=Q(J)-P(N+2-J)*Q(N+1)
 30   Continue
      B(N)=1.D0
      Return
      End
C*************************************************************************
      Subroutine Calkw (A,B,Y,N,Iflag)
C     Cette subroutine permet de construire la matrice de covariance K,
C     et la matrice de l'élément bruit W.

C     Paramètres d'entrée:
C     Eléments A(NxN) et B(Nx1) pour le calcul de K
C     ou
C     Eléments A(NxN) et C(1xN) pour le calcul de W.
C     Iflag : Iflag=0 pour le calcul de K (A --> A et B --- B ).
C             Iflag=1 pour le calcul de W (A --> A^T et B --> C^T).
C     N     : ordre du filtre.

C     Paramètres de sortie:
C     Y     : Matrice (NxN) représentant soit K ou soit W.

      REAL*8 A(50,50),B(50),Y(50,50),F(50,50),E(50,50),D(50,50)
      REAL*8 T,Epsilon
      Epsilon=1.D-10
      Do 20 i=1,n
                  Do 10 j=1,n
                        Y(i,j)=B(i)*B(j)
 10               Continue
 20   Continue
      IF (Iflag.EQ.0) Then
      Do 30 i=1,n
                  Do 25 j=1,n
                     F(i,j)=A(i,j)
 25               Continue
 30   Continue
                     Else
      Do 40 i=1,n
                  Do 35 j=1,n
                     F(i,j)=A(j,i)
```

```
35              Continue
40    Continue
                    end if
45    fx=0
      Call PM (F,Y,E,N,FX)
      fx=-1
      Call PM (E,F,D,N,FX)
      Do 60 i=1,n
                Do 50 j=1,n
                      Y(i,j)=Y(i,j)+D(i,j)
50              Continue
60    Continue
      Call Puiss(F,N)
      T=0.D0
      Do 80 i=1,n
                Do 70 j=1,n
                      T=T+DABS(F(i,j))
70               Continue
80    Continue
      If(T.GT.Epsilon) goto 45
      Return
      End
C*************************************************************************
      Subroutine Puiss (F,N)
C     Cette subroutine permet de calculer le carré d'une matrice.

C     Paramètres d'entrée:
C     N : l'ordre de la matrice.
C     F : la matrice F(NxN)

C     Paramètres de sortie:
C     F : La matrice résultat;exemple F= FxF.

      REAL*8 F(50,50),X(50,50)
      Do 30 I=1,N
         Do 20 J=1,N
            X(I,J)=0.D0
            Do 10 K=1,N
               X(I,J)=X(I,J)+F(I,K)*F(K,J)
   10       Continue
   20    Continue
   30 Continue
      Do 50 I=1,N
         Do 40 J=1,N
            F(I,J)=X(I,J)
   40    Continue
   50 Continue
      Return
      End
C*************************************************************************
      Subroutine Pm (A,B,C,N,FX)
C     cette subroutine permet de faire le produit de deux matrices.

C     Paramètres d'entrée:
C     N   : l'ordre des matrices.
C     A,B : Matrices NxN.
C     Fx  : Fx=0  produit  A x B
            Fx=-1 produit  A x B^T
            Fx=+1 produit  A^T x B
```

```
C     Paramètres de sortie:
C     C     : Matrice résultat NxN (C=AxB).

      REAL*8  A(50,50),B(50,50),C(50,50)
      DO 4 I=1,N
      DO 4 K=1,N
            C(I,K)=0.D0
         DO 4 J=1,N
             If (FX) 1,2,3
  1      C(I,K)=C(I,K)+A(I,J)*B(K,J)
         GOTO 4
  2      C(I,K)=C(I,K)+A(I,J)*B(J,K)
         GOTO 4
  3      C(I,K)=C(I,K)+A(J,I)*B(J,K)
  4   CONTINUE
      Return
      End
C*******************************************************************************
      Subroutine Mult (A,B,C,N,FM)
C     Cette subroutine permet de faire le produit d'une matrice (NxN) par
C     un vecteur (Nx1) et vice versa.

C     Paramètres d'entrée:
C     N  : L'ordre de la matrice et du vecteur, respectivement.
C     A  : Matrice (NxN).
C     B  : Vecteur (Nx1).
C     Fm : Fm=0 produit A x B
           Fm=1 produit B x A^T

C     Paramètres de sortie:
C     C  : Vecteur résultat Nx1 (C=AxB).

      Real*8 A(50,50),B(50),C(50)
      DO 20 I=1,N
        C(I)=0
        DO 10 J=1,N
          IF (FM.EQ.0) THEN
             C(I)=C(I)+A(I,J)*B(J)
          ELSE
             C(I)=C(I)+B(J)*A(J,I)
          END IF
  10    Continue
  20  Continue
      Return
      End
C*******************************************************************************
      Subroutine Vari (K,W,N,G)
C     Cette subroutine permet de calculer le gain du bruit de calcul
C     d'une structure d'état du filtre numérique choisi.

C     Paramètres d'entrée:
C     N    : l'ordre du filtre.
C     K,W : Matrices de covariance (K) et de l'élément bruit (W).

C     Paramètres de sortie:
C     G    : Gain de bruit du calcul.

      REAL*8 K(50,50),W(50,50),G
```

```
      G=0.D0
      DO I=1,N
          G=G+K(I,I)*W(I,I)
      END DO
      RETURN
      END
C*************************************************************************
      Subroutine Mino (Ks,Ws,T,N,Gmin,Del)
C     Cette subroutine permet d'obtenir la matrice de transformation
C     d'optimisation T en utilisant une normalisation appropriée.

C     Paramètres d'entrée:
C     N      : L'ordre du filtre.
C     Ks,Ws  : Matrices K et W de la structure canonique.
C     Del    : Paramètre de normalisation.

C     Paramètres intermédiaires:
C     Wr     : Vecteur d'ordre N donnant les valeurs propres de la matrice
C              produit V=KxW, dont leur racines carrées représentent les
C              modes de second ordre.

C     Paramètres de sortie:
C     T      : Matrice (NxN) d'état d'optimisation.
C     Gmin   : Gain du bruit de calcul en fonction des modes de second
C              ordre.

      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION Z(50,50),WR(50),WI(50),T0(50,50),V(50,50),Ws(50,50)
      DIMENSION W1(50,50),U(50,50),R0(50,50),W(50,50),R(50,50)
      DIMENSION AA(50,50),BB(50,50),T1(50,50),UN(50,50),T(50,50),X(50)
      REAL*8 K(50,50),M(50,50),ID(50,50),Ks(50,50)
      DO I=1,N
         UN(I,I)=DEL*DSQRT(Ks(I,I))
      END DO
      Call WK (Ks,Ws,UN,K,W,N)
      FX=0
      Call PM (K,W,V,N,FX)
      Call Eigen (V,SCALE,Z,WR,WI,N,IERR)
      Gmin=0.D0
      DO 20 I=1,N
           Gmin=Gmin+DSQRT(WR(I))
   20 CONTINUE
      Gmin=(Gmin**2)/N
      Call TransT0 (k,T0,N)
      FX=1
      Call PM (T0,W,U,N,FX)
      FX=0
      Call PM (U,T0,W1,N,FX)
      Call Eigen (W1,SCALE,Z,WR,WI,N,IERR)
      Call MatriceM (M,X,AA,WR,N)
      Call MatriceR0 (M,X,R0,N)
      FX=0
      Call PM (Z,AA,BB,N,FX)
      FX=-1
      Call PM(BB,R0,T1,N,FX)
      FX=0
      Call PM (T0,T1,T,N,FX)
      RETURN
      END
```

```
C*************************************************************************
      Subroutine WK (K,W,T,BB,PP,N)
C     Cette subroutine permet de calculer les nouvelles matrices après
C     avoir appliquer une matrice d'état de transformation.
C     Paramètres d'entrée:
C     N      : L'ordre du filtre.
C     T      : Matrice d'état de transformation.
C     K,W    : Matrices K et W originales.

C     Paramètres de sortie:
C     BB,PP : Matrices K et W transformées (BB=T'KT^T et PP=T^TWT).

      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION W(50,50),W1(50,50),U(50,50)
      DIMENSION  TT(50,50),V(50,50),PP(50,50)
      DIMENSION BB(50,50),T(50,50)
      REAL*8 K(50,50)
      DO I=1,N
        DO J=1,N
          TT(I,J)=T(I,J)
        END DO
      END DO
      CALL INVERSE (TT,N,IER)
      FX=1
      CALL PM (T,W,U,N,FX)
      FX=0
      CALL PM (U,T,PP,N,FX)
      CALL PM (TT,K,V,N,FX)
      FX=-1
      CALL PM (V,TT,BB,N,FX)
      RETURN
      END
C*************************************************************************
      Subroutine TransT0 (K,T0,N)
C     Cette subroutine permet de décomposer une matrice symétrique
C     (NxN) en produit de matrices triangulaires inférieure et supérieure,
C     respectivement, en utilisant la méthode de Chelosky. A ce but
C     On définit la matrice de transformation T0 telle que K=T0xT0^T.

C     Paramètres d'entrée:
C     N  : L'ordre du filtre.
C     K  : Matrice de covariance K (NxN).

C     Paramètres de sortie:
C     T0 : Matrice de transformation (NxN) qui transforme la matrice K
C          originale en une matrice identité.

      REAL*8 K(50,50),T0(50,50),S1,S2
      T0(1,1)=DSQRT(K(1,1))
      DO 10 J=2,N
           T0(J,1)=K(1,J)/T0(1,1)
  10  CONTINUE
      DO 50 I=2,N
           S1=0.D0
           DO 20 J1=1,I-1
                S1=S1+T0(I,J1)**2
  20       CONTINUE
           T0(I,I)=DSQRT(K(I,I)-S1)
           DO 40 J2=I+1,N
```

```
                S2=0.D0
                DO 30 L=1,I-1
                     S2=S2+T0(J2,L)*T0(I,L)
  30            CONTINUE
                T0(J2,I)=(K(I,J2)-S2)/T0(I,I)
  40        CONTINUE
  50    CONTINUE
        Return
        End
C*******************************************************************************
        Subroutine MatriceM (M,X,AA,WR,N)
C       Cette subroutine permet de calculer la matrice de transformation
C       d'état P*, qui est en fonction des modes de second ordre.

C       Paramètres d'entrée:
C       N  : L'ordre du filtre.
C       Wr : Vecteur des valeurs propres de la matrice produit KxW.

C       Paramètres intermédiaires:
C       Lambda : Vecteur contenant des éléments optimums en fonction
C                des modes de second ordre (Méthode de S. Ewang).

C       Paramètres de sortie:
C       M  : Matrice diagonale " P*" (NxN) contenant les éléments inverses
C            du vecteur Lambda.
C       X  : Vecteur contenant les éléments diagonaux de la matrice M.
C       AA ; Matrice inverse "(P*)-1 " (NxN) de M.

        REAL*8 M(50,50),WR(50),LAMBDA(50),X(50),AA(50,50),TEMP
        TEMP=0.D0
        DO 10 J=1,N
             TEMP=TEMP+DSQRT(WR(J))
  10    CONTINUE
        DO 20 I=1,N
             LAMBDA(I)=TEMP/(N*DSQRT(WR(I)))
  20    CONTINUE
        DO 30 I= 1,N
             M(I,I)=1/LAMBDA(I)
             X(I)=M(I,I)
             AA(I,I)=DSQRT(LAMBDA(I))
  30    CONTINUE
        RETURN
        END
C*******************************************************************************
        Subroutine MatriceR0 (M,X,R0,N)
C       Cette subroutine permet de calculer la matrice orthogonale R0 de
C       transformation d'état.

C       Paramètres d'entrée:
C       N  : L'ordre du filtre.
C       M  : Matrice diagonale P .
C       X  : Vecteur contenant les éléments de M.

C       Paramètres intermédiaires:
C       R  : Matrice orthogonale (NxN) R0=RN xRN-1 ... xR2 .

C       Paramètres de sortie:
C       R0 : Matrice de transformation orthogonale (NxN).
```

```
      Real*8 R0(50,50),R(50,50),X(50),CPSI,SPSI,H(50,50),S(50,50)
      Real*8 T(50,50)
      K=1
      L=1
      JJ=1
      DO I=1,N
        R0(I,I)=1.D0
      END DO
    1 IF((X(K).GT.1.AND.X(L+1).LT.1).OR.(X(K).LT.1.AND.X(L+1).GT.1))
     xGOTO 2
      L=L+1
      R(L,L)=1.D0
      GOTO 1
    2 CPSI=DSQRT((X(L+1)-1.D0)/(X(L+1)-X(K)))
      SPSI=DSQRT((1.D0-X(K))/(X(L+1)-X(K)))
      R(K,K)=CPSI
      R(L+1,L+1)=CPSI
      R(K,L+1)=SPSI
      R(L+1,K)=-1.D0*SPSI
      DO 20 I=L,N
         R(I+2,I+2)=1.D0
   20 Continue
      FX=0
      Call PM (R,R0,T,N,FX)
      DO I=1,N
        DO J=1,N
          R0(I,J)= T(I,J)
        END DO
      END DO
      Call PM (R0,M,S,N,FX)
      FX=-1
      Call PM (S,R0,H,N,FX)
      DO 40 I=1,L
         DO 30 J=1,L
            R(I,J+1)=0.D0
            R(J+1,I)=0.D0
   30    Continue
         R(I,I)=1.D0
   40 Continue
      K=K+1
      JJ=JJ+1
      X(L+1)=H(L+1,L+1)
      L=JJ
      IF (JJ.NE.N) GOTO 1
      Return
      End
C*********************************************************************
      Subroutine Inverse (A,N,Ier)
C     Cette subroutine permet de calculer l'inverse d'une matrice.

C     Paramètres d'entrée:
C     N  : L'ordre de la matrice.
C     A  : Matrice carrée NxN.

C     Paramètres de sortie:
C     A  : Matrice inverse NxN (A)^-1.

      Real*8 A(50,50),E(50),Ainv(50,50)
      Integer N,Ier,I,J,P(50)
```

```
      Do 10 I=1,N
        E(I)=0.D0
10    Continue
      Call Lufact(A,P,N,Ier)
      Do 20 I=1,N
         E(I)=1.D0
         Call Subst(A,P,E,Ainv,N,I)
         E(I)=0.D0
20    Continue
      Do 40 I=1,N
         Do 30 J=1,N
            A(I,J)=Ainv(I,J)
30       Continue
40    Continue
      Return
      End
      Subroutine Lufact(A,P,N,Ier)
      Real*8 A(50,50),Pivot,Temp
      Integer N,I,J,K,P(50),IP,L
      Ier=0
      Do 5 I=1,N
         P(I)=I
 5    Continue
      Do 50 K=1,N-1
         Pivot=Dabs(A(K,K))
         IP=K
         Do 10 I=K+1,N
            If (Pivot.Lt.Dabs(A(I,K))) Then
               Pivot=Dabs(A(I,K))
               IP=I
            Endif
10       Continue
         If (Pivot.Eq.0.D0) Goto 60
         If (IP.Ne.K) Then
            Do 20 I=1,N
               Temp=A(K,I)
               A(K,I)=A(IP,I)
               A(IP,I)=Temp
20          Continue
            L=P(K)
            P(K)=P(IP)
            P(IP)=L
         Endif
         Do 40 I=K+1,N
            A(I,K)=A(I,K)/A(K,K)
            Do 30 J=K+1,N
               A(I,J)=A(I,J)-A(I,K)*A(K,J)
30          Continue
40       Continue
50    Continue
      If (A(N,N).Eq.0.D0) Goto 60
      Return
60    Ier=1
      Return
      End
      Subroutine Subst(A,P,E,B,N,K)
      Real*8 A(50,50),B(50,50),E(50),Y(50),Temp
      Integer N,K,I,J,L,P(50)
      Y(1)=E(P(1))
```

```
      Do 20 I=2,N
         Temp=0.D0
         Do 10 J=1,I-1
            Temp=Temp+A(I,J)*Y(J)
   10    Continue
         Y(I)=E(P(I))-Temp
   20 Continue
       Y(N)=Y(N)/A(N,N)
       Do 40 I=1,N-1
          L=N-I
          Temp=0.D0
          Do 30 J=L+1,N
             Temp=Temp+A(L,J)*Y(J)
   30    Continue
         Y(L)=(Y(L)-Temp)/A(L,L)
   40 Continue
      Do 50 I=1,N
         B(I,K)=Y(I)
   50 Continue
      Return
      End
C*********************************************************************
      Subroutine ConstM (T,A,B,C,A1,B1,C1,N)
C     Cette subroutine permet d'obtenir la nouvelle structure d'etat
C     optimale du filtre numerique choisie.

C     Parametres d'entree:
C     N      : L'ordre du filtre.
C     A,B,C  : Matrices A, B et C de la structure canonique.
C     T      : Matrice de transformation d'optimisation.

C     Parametres de sortie:
C     A1,B1,C1 : Matrices Aoptm ,Boptm et Coptm de la structure optimale.

      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(50,50),B(50),C(50),A1(50,50),B1(50),C1(50),TT(50,50)
      DIMENSION xD(50,50),T(50,50)
      REAL*8 K(50,50)
      DO I=1,N
        DO J=1,N
          TT(I,J)=T(I,J)
        END DO
      END DO
      CALL INVERSE (TT,N,IER)
      FX=0
      CALL PM (TT,A,xD,N,FX)
      CALL PM (xD,T,A1,N,FX)
      FM=0
      CALL Mult (TT,B,B1,N,FM)
      FM=1
      CALL Mult (T,C,C1,N,FM)
      RETURN
      END
C*********************************************************************
C     Le sous programme EIGEN faisant partie du Logiciel EISPACK
C     a ete  transcrit selon la procedure exposée dans:
C     NUM. MATH. 13, 293-304(1969) BY PARLETT AND REINSCH.
C     HANDBOOK FOR AUTO. COMP., VOL.II-LINEAR ALGEBRA, 315-326(1971).
C*********************************************************************
```

Programme de synthèse des structures décomposées

```
      Programme de minimisation décomposée
C     Ce programme permet d'obtenir les structures décomposée parallèle
C     et cascade respectivement, avec un gain de bruit de calcul minimum
C     a partir d'une structure décomposée canonique

C     Paramètres d'entrée:
C     Q,P: Vecteurs des coefficients du numérateur et dénominateur,
C          respectivement du filtre prototype simule.
C     N  : l'ordre du filtre.#
C     IN : Paramètre de normalisation.

C     Paramètres de sortie
C     Gc,Gm  : Gain canonique et minimisé respectivement, de la structure
C              parallèle.
C     Gcc,Go : Gain canonique et minimisé respectivement, de la structure
C              cascade.

      IMPLICIT REAL*8 (A-H,O-Z)
      Real*8   P(21),Q(21),PB(21),IN

      OPEN (UNIT=15,FILE='COEF.DAT',STATUS='OLD')

      WRITE(*,*)'     '
      WRITE(*,*)'           STRUCTURES  DECOMPOSEES          '
      WRITE(*,*)' Voulez vous la structure parallèle ou cascade?'
      WRITE(*,*)' Appuyez sur 1 si parallèle ou 2 si cascade:'
      READ(*,*)PC
      IF (PC.EQ.1) THEN
       WRITE(*,*)'     '
       WRITE(*,*)'Donner facteur de normalisation &:'
       READ(*,*)IN
             ELSE
       WRITE(*,*)'     '
       WRITE(*,*)'Bloc Optimal ou Sections Optimales?'
       WRITE(*,*)'Appuyez sur 0 si Bloc optimal 1 si Sections Optimales'
       READ(*,*)CODE
       WRITE(*,*)'     '
       WRITE(*,*)'Donner facteur de normalisation &:'
       READ(*,*)IN
      END IF
      WRITE(*,*)'Donner ordre N du filtre:'
      READ(*,*)N
      DO I=1,N+1
         READ(15,*)Q(I)
      END DO
      DO I=1,N+1
          READ(15,*)P(I)
          PB(I)=P(I)
      END DO
C
      IF (PC.EQ.2) GOTO 10
      CALL PARA (P,Q,N,IN,Gm,Gc)
C
      write(*,*)'     '
      WRITE(*,*)'Pour la structure décomposée parallèle on trouve:'
      WRITE(*,*)'Gain canonique=',Gc
      WRITE(*,*)'Gain minimum=',Gm
      write(*,*)'     '
C
```

```
      WRITE(*,*)'Si vous voulez pour le même filtre la décomposition
      xcascade; Faites votre choix; '
      WRITE(*,*)'Bloc Optimal ou Sections Optimales?'
      WRITE(*,*)'Appuyez sur 0 si Bloc optimal 1 si Sections Optimales'
      WRITE(*,*)'Si non tapez (3) FIN:'
      READ(*,*)CODE
      WRITE(*,*)'      '
C
      IF (CODE.EQ.3) GOTO 20
 10   CALL CASCA (PB,Q,N,IN,Gcc,Go,Code)
C
      write(*,*)'      '
      WRITE(*,*)'Pour la structure décomposée cascade on trouve:'
      IF (CODE.EQ.1) Then
           WRITE(*,*)'Gain minimum (SO)=',Go
      ELSE
           WRITE(*,*)'Gain canonique=',Gcc
           WRITE(*,*)'Gain minimum (BO)=',Go
      END IF
      write(*,*)'      '
C
20    STOP
      END
C*************************************************************************
      Subroutine Para (P,Q,N,IN,Ga,Gb)
C     Cette subroutine calcule le gain du bruit de calcul de la
C     structure décomposée parallèle canonique et optimale, respectivement.

C     Paramètres d'entrée:
C     N  : L'ordre du filtre.
C     Q  : Vecteur contenant les coefficients du numérateur.
C     P  : Vecteur contenant les coefficients du dénominateur.
C     IN : Paramètre de normalisation.

C     Paramètres de sortie:
C     Gb : Gain canonique de la structure parallèle.
C     Ga : Gain optimale de la structure parallèle.

      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION P(21),Q(21),W(20,20),W2(2,2),R(20),PP(10),QQ(10)
      DIMENSION PN(10),QN(10),PD(10),QD(10),ZZ(10),WW(10),CT(20)
      DIMENSION BB(10,2),CC(10,2),AA(10,2,2),AT(20,20),BT(20),PB(21)
      COMPLEX*16 RN(10),RD(10),SN(10),SD(10),R1(10),R2(10)
      REAL*8 K(20,20),K2(2,2),AM(10,2,2),CM(10,2),BM(10,2),IN
      DO I=2,N+1
         R(I)=Q(I)-Q(1)*P(I)
      END DO
      K1=N
      CALL BAISTROW (P,PP,QQ,R1,R2,K1)
      CALL CONSTR1 (R,PP,QQ,R1,R2,AM,BM,CM,N)
      FN=0
      Ga=0.D0
      Gb=0.D0
      DO L=1,N/2
         CALL GAIN (AM,BM,CM,K2,W2,AA,BB,CC,G1,G2,L,FN,IN)
         Ga=Ga+G2
         Gb=Gb+G1
      END DO
      Return
```

```
      End
C*************************************************************************
      Subroutine Casca (PB,Q,N,IN,Gcc,Go,Code)
C     Cette subroutine calcule le gain du bruit de calcul de la structure
C     décomposée cascade canonique et optimale, respectivement.

C     Paramètres d'entrée:
C     N     : L'ordre du filtre.
C     Q     : Vecteur contenant les coefficients du numérateur.
C     Pb    : Vecteur contenant les coefficients du dénominateur.
C     In    : Paramètre de normalisation.

C     Paramètres de sortie:
C     Gcc   : Gain de la structure cascade canonique.
C     Go    : Gain de la structure cascade optimale.

      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION Q(21),W(20,20),W2(2,2),R(20),PP(10),QQ(10)
      DIMENSION PN(10),QN(10),PD(10),QD(10),ZZ(10),WW(10),CT(20)
      DIMENSION BB(10,2),CC(10,2),AA(10,2,2),AT(20,20),BT(20),PB(21)
      COMPLEX*16 RN(10),RD(10),SN(10),SD(10),R1(10),R2(10)
      REAL*8 K(20,20),K2(2,2),AM(10,2,2),CM(10,2),EM(10,2),IN
      D=(Q(1))**(2./N)
      K1=N
      L=N
      CALL BAISTROW (Q,PN,QN,RN,SN,K1)
      CALL BAISTROW (PB,PD,QD,RD,SD,L)
      CALL CONSTR2 (PN,QN,PD,QD,D,AM,EM,CM,N)
      IF (CODE.EQ.1) THEN
           FN=0
           DO M=1,N/2
              CALL GAIN (AM,EM,CM,K2,W2,AA,BB,CC,G1,G2,M,FN,IN)
           END DO
      ELSE
          CALL ABCD (N,AM,EM,CM,D,AT,BT,CT)
          IFLAG=0
          CALL CALKW1(AT,BT,K,N,IFLAG)
          IFLAG=1
          CALL CALKW1(AT,CT,W,N,IFLAG)
          GCC=0.D0
          DO I=1,N
            GCC=GCC+W(I,I)*K(I,I)
          END DO
          FN=1
          MM=0
          DO M=1,N-1,2
             DO I=1,2
                DO J=1,2
                   K2(I,J)=K(I+M-1,J+M-1)
                   W2(I,J)=W(I+M-1,J+M-1)
                END DO
             END DO
             MM=MM+1
             CALL GAIN (AM,EM,CM,K2,W2,AA,BB,CC,G1,G2,MM,FN,IN)
          END DO
      END IF
      CALL ABCD (N,AA,BB,CC,D,AT,BT,CT)
      IFLAG=0
      CALL CALKW1(AT,BT,K,N,IFLAG)
```

```
      IFLAG=1
      CALL CALKW1(AT,CT,W,N,IFLAG)
      GO=0.D0
      DO I=1,N
        GO=GO+W(I,I)*K(I,I)
      END DO
      Return
      End
C*************************************************************************
      Subroutine Baistrow (fA,PP,QQ,R1,R2,N)
C     Cette subroutine permet de décomposer un polynôme d'ordre N (pair)
C     en un produit de polynôme de second ordre par la méthode de Baistrow.

C     Paramètres d'entrée:
C     N  : L'ordre du polynôme.
C     Fa : polynôme d'ordre N.

C     Paramètres de sortie:
C     PP,QQ : Coefficients du polynôme de second ordre (X^2+ppX+qq).
C     R1,R2 : Les racines complexes du polynôme de second ordre.

      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION fA(20),B(20),C(20),PP(20),QQ(20),DET(20)
      COMPLEX*16 R1(20),R2(20)
      M=N
      S1=FA(1)
      DO I=1,N+1
         fA(I)=fA(I)/S1
      END DO
      E=1.D-10
      PRINT*,'Donner le couple (P,Q) initial :'
    1 READ*,P,Q
      L=1
   80 J=0
   60 B(1)=fA(1)
      B(2)=fA(2)-(P*B(1))
      DO K=3,N+1
         B(K)=fA(K)-P*B(K-1)-Q*B(K-2)
      END DO
      C(1)=B(1)
      C(2)=B(2)-P*C(1)
      IF (N.EQ.3) GOTO 40
      DO K=3,N-1
         C(K)=B(K)-P*C(K-1)-Q*C(K-2)
      END DO
   40 C(N)=-P*C(N-1)-Q*C(N-2)
      D=C(N-1)*C(N-1)-C(N)*C(N-2)
      U=B(N)*C(N-1)-B(N+1)*C(N-2)
      V=B(N+1)*C(N-1)-B(N)*C(N)
      IF (D.EQ.0) PRINT*,'PAS DE SOLUTIONS'
      Y1=U/D
      Z1=V/D
      P=P+Y1
      Q=Q+Z1
      IF (DABS(Y1)+DABS(Z1).LT.E) GOTO 70
      J=J+1
      IF (J.LE.9000) GOTO 60
      PRINT*,'Changer le couple (P,Q)" TROP D'ITERATIONS"'
      GOTO 1
```

```
 70   PP(L)=P
      QQ(L)=Q
      N=N-2
      DO I=1,N+1
         fA(I)=B(I)
      END DO
      IF (N.GT.2) THEN
           L=L+1
           GOTO 80
        ELSE
        IF (N.EQ.2) THEN
             L=L+1
             PP(L)=B(2)
             QQ(L)=B(3)
        ELSE
           PRINT*,'N EST IMPAIR'
        ENDIF
      ENDIF
      DO L=1,M/2
            DET(L)=(PP(L)**2)-4*QQ(L)
          IF (DET(L).GE.0) THEN
            R1(L)=0.5*(DSQRT(DET(L))-PP(L))
            R2(L)=-0.5*(DSQRT(DET(L))+PP(L))
          ELSE
            ALPHA=-0.5*PP(L)
            BETA=0.5*DSQRT(-DET(L))
            R1(L)=DCMPLX(ALPHA,BETA)
            R2(L)=DCMPLX(ALPHA,-BETA)
          END IF
      END DO
      RETURN
      END
C*************************************************************************
      Subroutine Constr1 (DN,PP,QQ,R1,R2,AM,BM,CM,N)
C     Cette subroutine permet de décomposer une fraction rationnelle en
C     une somme de fractions rationnelle d'ordre 2, ainsi d'obtenir les
C     sections d'ordre deux de la structure parallèle canonique.

C     Paramètres d'entrée:
C     N      : L'ordre du filtre.
C     M      : Le numéro de la section d'ordre 2 (M=1,...,N/2).
C     Dn     : polynôme d'ordre (N-1) du numérateur.
C     PP,QQ  : coefficients du polynôme de second ordre (X²+ppX+qq)
C     R1,R2  : Les racines complexes du polynôme de second ordre.

C     Paramètres de sortie:
C     Am,Bm,Cm : Structure de second ordre canonique.

      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION DN(21),PP(10),QQ(10),ZZ(10),WW(10)
      DIMENSION AM(10,2,2),BM(10,2),CM(10,2)
      COMPLEX*16 R1(20),R2(20),AN(20),BN(20)
      COMPLEX*16 PD1,PD2,PN1,PN2,AD(20),BD(20),X(20),Y(20)
      DO L=1,N/2
         PN1=DN(N+1)
         PN2=DN(N+1)
         DO I=1,N-1
            PN1=PN1+DN(N+1-I)*(R1(L)**I)
            PN2=PN2+DN(N+1-I)*(R2(L)**I)
```

```
            END DO
            AN(L)=PN1
            BN(L)=PN2
            PD1=(1,0)
            PD2=(1,0)
            DO 50 J=1,N/2
               IF(J.EQ.L) GOTO 50
               PD1=PD1*((R1(L)**2)+(PP(J)*R1(L))+QQ(J))
               PD2=PD2*((R2(L)**2)+(PP(J)*R2(L))+QQ(J))
 50         CONTINUE
            AD(L)=PD1
            BD(L)=PD2
            X(L)=AN(L)/AD(L)
            Y(L)=BN(L)/BD(L)
            ZZ(L)=(X(L)-Y(L))/(R1(L)-R2(L))
            WW(L)=X(L)-ZZ(L)*R1(L)
         END DO

C        Construction des matrices A, B et C d'ordre 2.

         DO M=1,N/2
            BM(M,1)=0.D0
            BM(M,2)=1.D0
            AM(M,1,1)=0.D0
            AM(M,1,2)=1.D0
            AM(M,2,1)=-QQ(M)
            AM(M,2,2)=-PP(M)
            CM(M,1)=WW(M)
            CM(M,2)=ZZ(M)
         END DO
         RETURN
         END
C*************************************************************************
         Subroutine Constr2 (PN,QN,PD,QD,D,AM,BM,CM,N)
C        Cette subroutine permet de construire les cellules de second ordre
C        canoniques issue d'une décomposition produit de la fonction de
C        transfert (structure cascade).

C        Paramètres d'entrée:
C        N      : L'ordre du filtre.
C        D      : Coefficient du degre 3 du polynôme du numérateur.
C        Pn,Qn  : Coefficients du polynôme du numérateur de second ordre.
C        Pd,Qd  : Coefficients du polynôme du dénominateur de second ordre.

C        Paramètres de sortie:
C        Am,Bm,CM : Structure de second ordre canonique.

         REAL*8 PN(10),PD(10),QN(10),QD(10),AM(10,2,2),BM(10,2),CM(10,2),D
         DO L=1,N/2
            AM(L,1,1)=0.D0
            AM(L,1,2)=1.D0
            AM(L,2,1)=-QD(L)
            AM(L,2,2)=-PD(L)
            BM(L,1)=0.D0
            BM(L,2)=1.D0
            CM(L,1)=(QN(L)-QD(L))*D
            CM(L,2)=(PN(L)-PD(L))*D
         END DO
         RETURN
```

```
      END
C*********************************************************************
      Subroutine ABCD (L,AA,BB,CC,D,AT,BT,CT)
C     Cette subroutine permet d'obtenir la structure bloc cascade
C     à partir des cellules de second ordre.
C     Paramètres d'entrée:
C     L          : L'ordre du filtre.
C     AA,BB,CC,D : Structure A, B, C et D de second ordre.

C     Paramètres de sortie:
C     AT,BT,CT   : Structure bloc A, B et C d'ordre N.

      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION AT(20,20),BB(10,2),CC(10,2),AA(10,2,2)
      DIMENSION BC(10,10,2,2),PP(10,10,2,2),BT(20),CT(20)
      NN=L/2-1
      DO LB=1,NN
         DO MM=1,NN
            DO I=1,2
               SS=0
               DO J=1,2
                  BC(LB,MM,I,J)=BB(MM+LB,I)*CC(MM,J)*D**(LB-1)
               END DO
            END DO
         END DO
         NN=NN-1
      END DO
        DO K=1,L/2
           DO I=1,2
              DO J=1,2
                 PP(1,K,I,J)=AA(K,I,J)
              END DO
           END DO
        END DO
        NN=L/2+1
        LLB=1
        DO M=3,L-1,2
           NN=NN-1
           DO K=1,NN-1
              DO I=1,2
                 DO J=1,2
                    PP(M,K,I,J)=BC(LLB,K,I,J)
                 END DO
              END DO
           END DO
           LLB=LLB+1
        END DO
        NN=L/2+1
        DO M=1,L-1,2
           NN=NN-1
           DO K=1,NN
              DO I=1,2
                 DO J=1,2
                    AT(I+M+2*K-3,J+2*K-2)=PP(M,K,I,J)
                 END DO
              END DO
            END DO
         END DO
        IC=0
```

```
      DO M=1,L/2
         DO I=1,2
            IC=IC+1
            BT(IC)=BB(M,I)*D**(M-1)
            CT(IC)=CC(M,I)*D**(L/2-M)
         END DO
      END DO
      RETURN
      END
C*************************************************************************
      Subroutine GAIN (AM,BM,CM,K,W,AA,BB,CC,G,GMIN,M,FN,IN)
C     Cette subroutine permet de minimiser une structure d'ordre 2, en
C     utilisant la méthode de S. Hwang.

C     Paramètres d'entrée:
C     Am,Bm,Cm : Structure canonique (A,B,C) de second ordre.
C     K,W      : Matrices d'ordre 2 extraites des matrices blocs K et W
C                d'ordre N (uniquement pour le cas cascade).
C     In       : Paramètre de normalisation.
C     M        : Numéro de la structure (M=1,...,N/2).
C     Fn       : Fn=0 La minimisation s'effectue à partir de (A,B,C).
C              : Fn=1 La minimisation s'effectue à partir de (K,W).

C     Paramètres de sortie:
C     AA,BB,CC : Structure optimale (A,B,C) de second ordre .
C     G        : Gain de la structure canonique de second ordre.
C     Gmin     : Gain de la structure optimale de second ordre.

      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(2,2),B(2),C(2),W(2,2),V0(2,2),V1(2,2),V2(2,2)
      DIMENSION A2(2,2),E(2),DP(2),VP(2,2),R(2,2),U(2,2),W1(2,2)
      DIMENSION TI(2,2),DI(2,2),T(2,2),A1(2,2),B1(2),C1(2),AM(10,2,2)
      DIMENSION AA(10,2,2),BB(10,2),CC(10,2),UN(2,2),TN(2,2),BM(10,2)
      REAL*8 K(2,2),K1(2,2),CM(10,2),IN
      DO I=1,2
         B(I)=BM(M,I)
         C(I)=CM(M,I)
            DO J=1,2
               A(I,J)=AM(M,I,J)
            END DO
         END DO
      DO I=1,2
         DO J=1,2
            A1(I,J)=A(I,J)
         END DO
      END DO
      IF (FN.NE.0) GOTO 14
      iflag=0
      CALL calKW (A,B,K,iflag)
      iflag=1
      CALL calKW (A,C,W,iflag)
  14  FX=0
      CALL PM (K,W,V0,FX,2)
      CALL EIGEN2 (V0,DP,VP)
      G=K(1,1)*W(1,1)+K(2,2)*W(2,2)
      GMIN=((DSQRT(DP(1))+DSQRT(DP(2)))**2)/2.D0
      V0(1,1)=DSQRT(K(1,1))
      V0(1,2)=0.D0
      V0(2,1)=K(1,2)/V0(1,1)
```

```
      V0(2,2)=(DSQRT((K(1,1)*K(2,2))-K(1,2)**2))/V0(1,1)
      S=DSQRT(DP(1))+DSQRT(DP(2))
      V1(1,1)=DSQRT(S/(2.D0*DSQRT(DP(2))))
      V1(2,2)=DSQRT(S/(2.D0*DSQRT(DP(1))))
      Q=1.D0/DSQRT(2.D0)
      DO I=1,2
        DO J=1,2
         V2(I,J)=Q
        END DO
      END DO
      V2(2,1)=-1.D0*Q
      FX=1
      CALL PM (V0,W,A2,FX,2)
      FX=0
      CALL PM (A2,V0,R,FX,2)
      CALL EIGEN2 (R,RP,VP)
      CALL PM (VP,V1,A2,FX,2)
      FX=-1
      CALL PM (A2,V2,U,FX,2)
      FX=0
      CALL PM (V0,U,T,FX,2)
      CALL INVERSE (T,TI)
      FX=1
      CALL PM (T,W,A2,FX,2)
      FX=0
      CALL PM (A2,T,W1,FX,2)
      CALL PM (TI,K,DI,FX,2)
      FX=-1
      CALL PM (DI,TI,K1,FX,2)
      UN(1,1)=IN*DSQRT(K1(1,1))
      UN(2,2)=IN*DSQRT(K1(2,2))
      CALL INVERSE (UN,TI)
      FX=1
      CALL PM (UN,W1,A2,FX,2)
      FX=0
      CALL PM (A2,UN,W1,FX,2)
      CALL PM (TI,K1,DI,FX,2)
      FX=-1
      CALL PM (DI,TI,K1,FX,2)
      FX=0
      CALL PM (T,UN,TN,FX,2)
      CALL INVERSE (TN,TI)
      CALL PM (TI,A1,DI,FX,2)
      CALL PM (DI,TN,A1,FX,2)
      FM=0
      CALL MULT (TI,B,B1,FM)
      FM=1
      CALL MULT (TN,C,C1,FM)
      G1=K1(1,1)*W1(1,1)+K1(2,2)*W1(2,2)
         DO I=1,2
            BB(M,I)=B1(I)
            CC(M,I)=C1(I)
            DO J=1,2
               AA(M,I,J)=A1(I,J)
            END DO
         END DO
      RETURN
      END
C*****************************************************************************
```

```
      Subroutine Calkw (F,D,V,IFLAG)
C     Cette subroutine permet de calculer les matrices K (2x2) et W(2x2)
C     à partir d'une résolution de système de 3 équations à 3 inconnues.
C     Paramètres d'entrée:
C     F,D   : Matrices A (2x2) (respect. A) et B (2x1) (respect. C).
C     Iflag : Iflag=0   F=A  et D=B (calcul de K(2x2)).
C             Iflag=1   F=A' et D=C' (calcul de W(2x2)).

C     Paramètres de sortie:
C     V     : Matrice K (respect. W).

      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION F(2,2),D(2),V(2,2)
      IF (IFLAG.EQ.0) GOTO 10
      U=F(1,2)
      F(1,2)=F(2,1)
      F(2,1)=U
10    AL1=(F(1,1)**2)-1
      AL2=2*F(1,1)*F(1,2)
      AL3=F(1,2)**2
      BE1=F(1,1)*F(2,1)
      BE2=(F(1,2)*F(2,1))+(F(1,1)*F(2,2))-1
      BE3=F(1,2)*F(2,2)
      GA1=F(2,1)**2
      GA2=2*F(2,1)*F(2,2)
      GA3=(F(2,2)**2)-1
      P1=(AL2*GA1)-(GA2*AL1)
      PP1=(AL2*BE1)-(AL1*BE2)
      P2=(AL3*GA1)-(GA3*AL1)
      PP2=(AL3*BE1)-(AL1*BE3)
      Q1=((D(2)**2)*AL1)-((D(1)**2)*GA1)
      Q2=(D(1)*D(2)*AL1)-((D(1)**2)*BE1)
      V(2,2)=((PP1*Q1)-(P1*Q2))/((PP1*P2)-(PP2*P1))
      V(1,2)=(Q1-(P2*V(2,2)))/P1
      V(2,1)=V(1,2)
      V(1,1)=((D(1)**2)+(AL2*V(1,2))+(AL3*V(2,2)))/(-1.D0*AL1)
      RETURN
      END
C*********************************************************************
      Subroutine Mult (A,B,C,FM)
C     Cette subroutine calcule le produit d'une matrice (2x2) par un
C     vecteur (2x1) et vice versa.

      REAL*8 A(2,2),B(2),C(2)
      DO I=1,2
         C(I)=0.D0
         DO J=1,2
            IF (FM.EQ.0) THEN
                C(I)=C(I)+A(I,J)*B(J)
            ELSE
                C(I)=C(I)+B(J)*A(J,I)
            END IF
         END DO
      END DO
      RETURN
      END
C*********************************************************************
      Subroutine Eigen2 (A,DP,VP)
C     Cette subroutine calcule les valeurs propres (Dp) et les vecteurs
```

```
c     propres (Vp) d'une matrice A(2x2).

      REAL*8 A(2,2),DP(2),VP(2,2),D,E,F,G
      D=((A(1,1)+A(2,2))**2)-4*((A(1,1)*A(2,2))-(A(1,2)*A(2,1)))
      E=DSQRT(D)
      DP(1)=0.5*(A(1,1)+A(2,2)-E)
      DP(2)=0.5*(A(1,1)+A(2,2)+E)
      VP(1,1)=-2.D0*A(2,1)/(A(1,1)-A(2,2)-E)
      VP(1,2)=-2.D0*A(2,1)/(A(1,1)-A(2,2)+E)
      F=DSQRT(1+VP(1,1)**2)
      G=DSQRT(1+VP(1,2)**2)
      VP(2,1)=1.D0/F
      VP(2,2)=1.D0/G
      VP(1,1)=VP(1,1)/F
      VP(1,2)=VP(1,2)/G
      RETURN
      END
C*************************************************************************
      Subroutine Inverse (A,AI)
c     Cette subroutine calcule la matrice inverse Ai(2x2) d'une matrice
c     A(2x2).

      REAL*8 A(2,2),AI(2,2),DETA
      DETA=(A(1,1)*A(2,2))-(A(1,2)*A(2,1))
      AI(1,1)=A(2,2)/DETA
      AI(2,2)=A(1,1)/DETA
      AI(1,2)=(-1.D0*A(1,2))/DETA
      AI(2,1)=(-1.D0*A(2,1))/DETA
      RETURN
      END
C*************************************************************************
c      Note: Subroutine Calkw1 est la même utilisée dans le programme
c            précédent de minimisation.
C*************************************************************************
```