N° d'ordre : 298

50376
1988
195

THESE

présentée à

L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE FLANDRES ARTOIS

pour obtenir le

DOCTORAT en ELECTRONIQUE

par

Vincent DEVLAMINCK

ETUDE DES BRUITS NUMERIQUES DANS LES STRUCTURES DE FILTRES A COEFFICIENTS FIXES. APPLICATIONS A L'ARITHMETIQUE CLASSIQUE ET A L'ARITHMETIQUE DISTRIBUEE.

Soutenue le 21 Décembre.1988 devant la commission d'examen

Membres du Jury :	L. Raczy	Président
	M. Bellanger	Rapporteur
	C. Gimenes	Rapporteur
	J.P. Dubus	Directeur de thèse
	H. Vu Thien	Examinateur
	A. Lebrun	Examinateur

A mes parents

A Christine

A mes enfants

Ce travail a été effectué au sein du Laboratoire de Mesures Automatiques de l'Université des Sciences et Technique de Lille Flandres Artois.

Je tient à exprimer tout d'abord ma profonde reconnaissance à Monsieur le Professeur J.P.DUBUS qui a assuré la direction de cette étude, pour son aide constante et efficace à la réalisation de ce travail. Je le remercie de la confiance qu'il a investie en moi, du soutien qu'il m'a apporté.

Je suis particulièrement reconnaissant envers Monsieur le Professeur L.RACZY pour l'honneur qu'il me fait de présider le jury.

Je remercie Monsieur le Professeur M.BELLANGER, adjoint au Directeur scientifique de la société T.R.T., pour l'attention qu'il a portée à ce travail ainsi que pour sa présence dans le jury.

Je remercie Monsieur le Professeur C.GIMENES Directeur des recherches à l'Institut National des Télécommunications pour les conseils qu'il m'a prodigués et pour sa présence dans le jury.

Mes remerciements vont également à Messieurs les Professeurs H.VU THIEN du Conservatoire National des Arts et Métiers de Paris et A.LEBRUN directeur de recherche au CRESMAT qui ont bien voulu faire partie de cette commission d'examen.

Qu'il me soit permis d'associer ces remerciements à Messieurs F.WAUQUIER, F.WATTRELOT, R.VAN ROMPU et P.ALTMAYER pour la sympathie qu'il m'ont manifestée tout au long de ce travail

Que les membres du personnel soit aussi remerciés, en particulier Madame R.CASTEGNIER et Monsieur J.P.DEHORTER

SOMMAIRE

SOMMAIRE.

CHAPITRE II.

CHAPITRE III.

INTRODUCTION

INTRODUCTION

La place de plus en plus importante accordée aux techniques numériques dans les réalisations électroniques d'aujourd'hui est le résultat des connaissances acquises en ce domaine ainsi que du développement des technologies correspondantes. L'intérêt du traitement numérique du signal n'est donc plus à démontrer. C'est en particulier le cas pour les techniques de filtrage numériques qui possèdent entre autres avantages sur les techniques analogiques d'être parfaitement reproductibles, d'être stables dans le temps et d'être aisément adaptables. La synthèse de tels filtres est donc un problème important que l'on peut décomposer en deux étapes principales. Dans un premier temps, on s'attache à définir la fonction de filtrage à réaliser compte tenu des spécifications données. C'est à dire la bande bande passante, l'atténuation hors bande, le gain, ou d'une manière plus générale le gabarit du filtre. Cette première étape débouche sur une caractérisation de la fonction de transfert entrée-sortie du filtre ce qui se traduit en général par l'obtention de la fonction de transfert en z. Il existe plusieurs méthodes pour arriver à cette fonction H(z). Parmi celles-ci on peut citer les techniques de digitalisation de filtres analogiques : méthodes des différences finies, de la réponse impulsionnelle invariante, transformation bilinéaire ou encore transformation directe en z). Ces méthodes sont utilisées pour la synthèses des filtres à réponse impulsionnelle infinie (RII). Dans le cas de filtres à réponse impulsionnelle finie (RIF) la fonction de transfert en z est obtenue en général à partir de la transformée de Fourier inverse de la réponse fréquentielle du filtre. Cette première étape peut être réalisée d'une manière automatique ou semi-automatique à l'aide d'ordinateurs. Elle est souvent couplée à un module d'optimisation du type algorithme des moindres carrés, algorithme de Remez ou autres.

Une fois la fonction de transfert déterminée, l'étape suivante consiste à en tirer une structure de filtre. Celle-ci est généralement donnée sous forme d'un graphe de fluence du signal ou sous forme des équations d'état du filtre que l'on exprime de façon standard par les relations matricielles

$$x(n+1) = A.x(n) + B.u(n)$$

$$y(n) = C.x(n) + D.u(n)$$

où u(n) et y(n) représentent respectivement les éléments des séquences d'entrée et de sortie du filtre à l'instant nT (T période d'échantillonnage) et x(n), le vecteur d'état du système au même instant. A, B, C, D sont les matrices d'état du filtre et l'ensemble peut

être considéré comme une représentation du filtre dans l'espace d'état. Les équations d'état sont reliées à la fonction de transfert en z par la relation :

$$H(z) = C.(z.I-A)^{-1}.B + D$$

Le principal problème à ce niveau de la synthèse des filtres vient du fait qu'à une fonction de transfert donnée, correspond une infinité de réalisations possibles. En effet, si T est une matrice non singulière de rang n, alors la représentation dans l'espace d'état définie par :

$$A' = T^{-1}.A.T$$

$$B' = T^{-1}.B$$

$$C' = C.T$$

$$D' = D$$

correspond à la réalisation d'un filtre ayant la même fonction de transfert que le filtre de représentation (A,B,C,D). En fait, ces réalisations de la même fonction de transfert sont réellement équivalentes si les opérateurs qui les constituent, travaillent en arithmétique à précision infinie. Dans la pratique, seules sont utilisées les arithmétiques travaillant sur des mots de longueur finie telles que l'arithmétique à virgule flottante ou l'arithmétique à virgule fixe. Cette dernière étant la plus répandue compte tenu de la relative aisance de sa mise en oeuvre. On est donc amené à devoir prendre en compte les différents effets de ce travail en précision finie; c'est à dire à tenir compte du problème des erreurs d'arrondi interne, de la sensibilité à la quantification des coefficients, du dépassement de capacité ou encore des cycles limites [1-8]. Ce faisant, on met en évidence la non équivalence qui existe entre les différentes réalisations possibles d'une même fonction de transfert. Naturellement le choix se portera sur la structure qui minimise le plus les effets du calcul en précision finie tout en nécessitant le moins possible d'opérateurs pour sa réalisation; ces deux critères pouvant être contradictoires [9].

La recherche d'une structure optimale nécessite donc dans un premier temps, une connaissance de la technique de quantification et des différentes méthodes qui permettent sa mise en oeuvre puisque la quantification est le principe qui est à l'origine des problèmes cités précédemment. Il convient ensuite de caractériser les principales sources de bruit présentes dans les filtres numériques à savoir les multiplieurs qui pour des raisons évidentes de dimensionnement des résultats, doivent faire appel aux techniques de quantification. Toutefois, lors de la synthèse d'un filtre, la grandeur intéressante à connaître en terme de bruit, est la valeur du bruit (ou de sa puissance) en sortie du filtre. L'étape suivante consiste donc à déterminer l'effet en sortie d'un filtre des différentes sources internes de bruit. On accède ainsi à une mesure de la puissance

de bruit en sortie d'un filtre et il devient possible de classer les diverses réalisations possibles d'une même fonction de transfert. Il reste alors à explorer les différentes techniques d'optimisation des filtres de manière à arriver à la structure à bruit minimal. Ces problèmes restent toutefois structurellement liés à la façon de réaliser les filtres à savoir au moyen d'opérateurs arithmétiques localisés (additionneurs et multiplieurs). La technique de l'arithmétique distribuée en proposant de nouvelles structures de filtres constitue une alternative possible à ce problème.

CHAPITRE I

I. INTRODUCTION.

La quantification est une opération qui fait correspondre à une variable d'entrée x, une valeur y choisie dans un ensemble fini de N valeurs prédéterminées. On la rencontre naturellement dans tous les problèmes de conversion analogique-numérique et d'une manière plus générale dans tous les processus faisant appel à l'arithmétique des mots de longueur finie. L'importance de ses effets et donc la place qu'elle occupe dans l'étude d'un processus dépend de ce processus lui-même. Le filtrage numérique est un de ceux où son rôle peut devenir prépondérant.

Une compréhension claire des effets de la quantification permet de déterminer par exemple le nombre de bits nécessaires par échantillon pour obtenir un processus travaillant avec une précision donnée. Soit encore, la puissance de bruit inévitable pour un nombre de bits fixé.

I.1. CLASSIFICATION DES DIFFERENTS TYPES DE QUANTIFICATIONS.

Il existe différents types de quantifications que l'on peut classer comme suit [10] :

- Quantification sans mémoire.

  Le quantificateur détermine la valeur de sortie uniquement à partir de la valeur de l'échantillon présent, indépendamment des autres valeurs, précédentes ou suivantes.

- Quantification vectorielle (ou bloc ou multidimensionnelle).

  A partir d'un ensemble de k échantillons, c'est à dire d'un vecteur de dimension k dans un espace euclidien, le quantificateur restitue un vecteur de sortie pris dans un ensemble fini.

- Quantification séquentielle.

  Le quantificateur stocke des informations sur les échantillons précédents et génère la sortie quantifiée en fonction de l'échantillon présent et des informations préalablement stockées.

Tout quantificateur fonctionne suivant l'une de ces trois techniques ou suivant une réunion de plusieurs d'entre elles.

I.2. QUANTIFICATION SCALAIRE SANS MEMOIRE.

La quantification scalaire peut être considérée comme un cas particulier de la quantification vectorielle (cas de la dimension 1). Cependant, son usage très largement répandu lui confère une place à part.

I.2.1. Définition.

Un quantificateur scalaire sans mémoire à N points est défini par un ensemble de N+1 niveaux de décision :

$$\{ X[0] , X[1] , \ldots , X[N] \}$$

et un ensemble de N niveaux de reconstruction :

$$\{ Y[1] , Y[2] , \ldots , Y[N] \}$$

si x représente un échantillon d'entrée (dans le cas général c'est un échantillon qui peut prendre une valeur quelconque dans l'intervalle $] - \infty, + \infty [$) l'opération de quantification est alors donnée par :

$$X[i] \ / \ X[i-1] \leq x < X[i] \implies Q(x) = Y[i]$$

Remarque : dans le cas où x prend ses valeurs dans l'intervalle non borné, il est nécessaire d'avoir :

$$X[0] = - \infty$$
$$X[N] = + \infty$$

Dans le cas où x prend ses valeurs dans un intervalle borné X[0] et X[N] seront données respectivement par la borne inférieure et la borne supérieure de l'intervalle de définition de x.

Un quantificateur est souvent déterminé par sa courbe caractéristique dont l'allure générale est celle d'une fonction en escalier (Figure I.1)

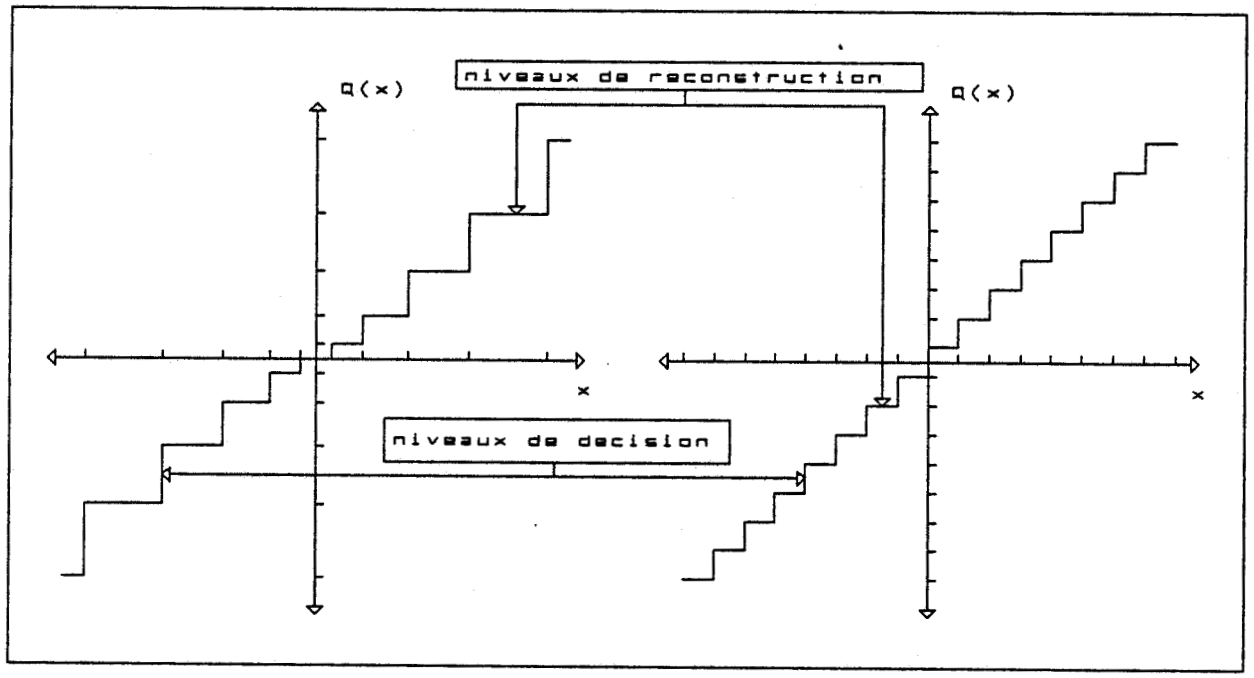


Figure I.1 - *Exemples de caractéristiques de quantificateurs (non uniforme et uniforme).*

L'opération de quantification Q provoque évidemment une altération du signal propagé x, que l'on désigne habituellement sous le nom de bruit de quantification et que l'on note e(t) tel que :

$$e(t) = Q[x(t)] - x(t) \tag{1.1}$$

de sorte que qu'il est possible de modéliser la quantification d'un signal x(t) sous la forme :

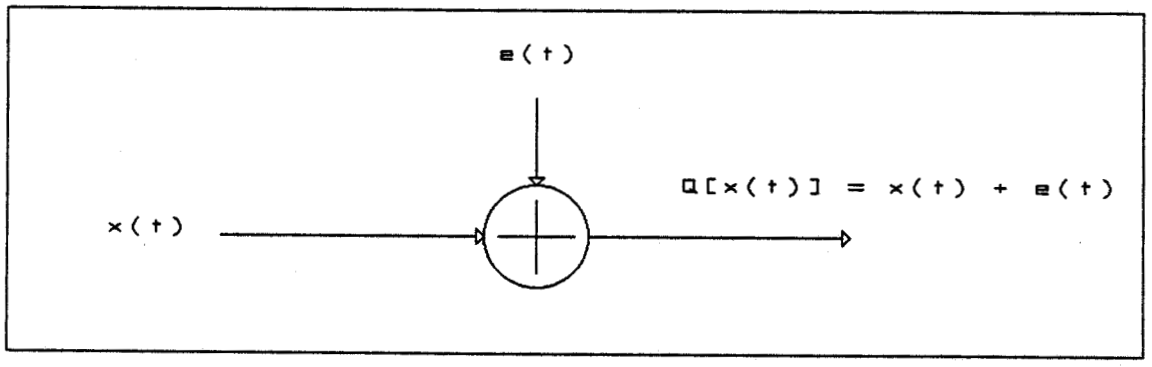


Figure I.2 - *Modélisation de la quantification.*

I.2.2. Caractérisation d'un quantificateur.

I.2.2.1. Rapport signal sur bruit.

Une mesure de la performance d'un quantificateur sera donné par l'étude des différents moments statistiques de la fonction e(t); en particulier par celle de la puissance de bruit ou encore en admettant l'hypothèse du bruit centré, par celle de la variance σ^2 de e(t) suivant la formule :

$$\sigma_e^2 = \int_{-\infty}^{+\infty} e^2 . p(e)\ de \tag{1.2}$$

où p(e) représente la fonction densité de probabilité du signal erreur e. Dans le cas où l'on sépare le domaine d'intégration en N intervalles cette formule devient :

$$\sigma_e^2 = \sum_{i=1}^{N} \int_{x_{i-1}}^{x_i} e^2 . p(e)\ de \tag{1.3}$$

σ_e^2 apparaissant comme la puissance du signal erreur, les performances du quantificateur pourront également être données par l'étude du rapport signal sur bruit (SNR) défini par :

$$SNR = 10 . \log \left[\frac{\sigma^2}{\sigma_e^2} \right] \tag{1.4}$$

si σ^2 représente la variance du signal d'entrée.

I.2.2.2. Robustesse d'un quantificateur.

La notion de rapport signal sur bruit permet de comparer différents quantificateurs relativement à un signal de puissance donné. Dans le cas où le quantificateur doit traiter des signaux d'entrée ayant des fonctions densité de probabilité de formes variées (c'est notamment le cas en transmission de la parole), on caractérisera son immunité vis à vis des variations de ces fonctions par la notion de robustesse. Celle-

ci se traduit en pratique par l'obtention d'une courbe donnant le rapport signal sur bruit en fonction de la puissance du signal d'entrée.

I.2.2.3. Facteur de crête pour un signal donné.

Dans le cas enfin où le signal d'entrée n'est pas borné, on caractérise le dépassement de capacité du quantificateur à l'aide du facteur de crête F_c suivant la formule :

$$F_c = 20.\log\left[\frac{A_m}{E_{eff}}\right] \qquad (1.5)$$

où A_m représente l'amplitude maximale du quantificateur et E_{eff} la valeur efficace du signal à traiter (soit encore σ la racine carrée de la variance de ce signal).

En pratique, le facteur de crête F_c sert à caractériser le quantificateur au moment de sa construction. A savoir, pour un signal de fonction de densité de probabilité donnée, on détermine A_m de manière à avoir une probabilité P de dépassement de capacité inférieure à une valeur préalablement fixée.

A_m est généralement pris telle que $P < 10^{-5}$ ce qui donne par exemple pour un signal gaussien de variance unité qui a donc une fonction densité de probabilité de la forme :

$$p(x) = \frac{1}{\sqrt{2.\pi}}\, e^{\frac{-x^2}{2}} \qquad (1.6)$$

une valeur du facteur de crête égale à 12.9 dB correspondant à une amplitude maximale A_m du quantificateur de 4.45.

I.2.3. Quantification uniforme.

Parmi tous les types de quantifications scalaires sans mémoire, on distingue plus particulièrement le cas du quantificateur de caractéristique uniforme (confère figure I.1), pour lequel les différents niveaux de décision sont équidistants. On appelle q quantum de base, la longueur de l'intervalle séparant deux niveaux de décision.

I.2.3.1. Techniques de la troncature et de l'arrondi.

Les quantifications par troncature ou par arrondi sont les deux techniques les plus courantes parmi les quantificateurs scalaires uniformes. Leurs caractéristiques dont données en figures I.3 et I.4 respectivement.

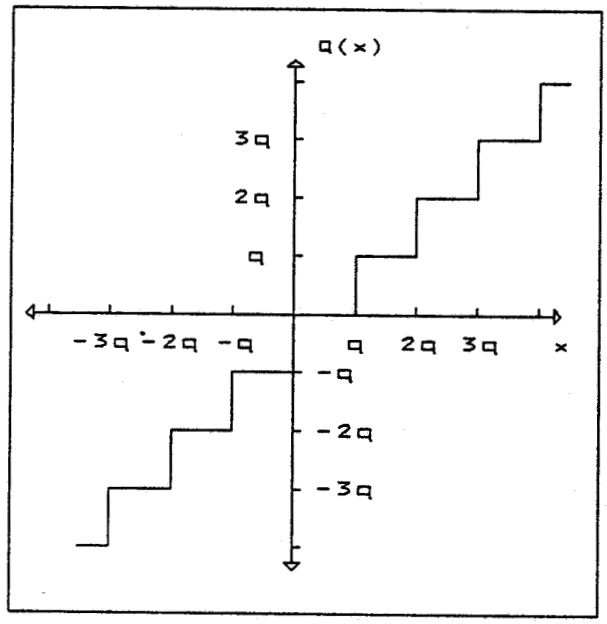


Figure I.3 - *Caractéristique de la troncature.*

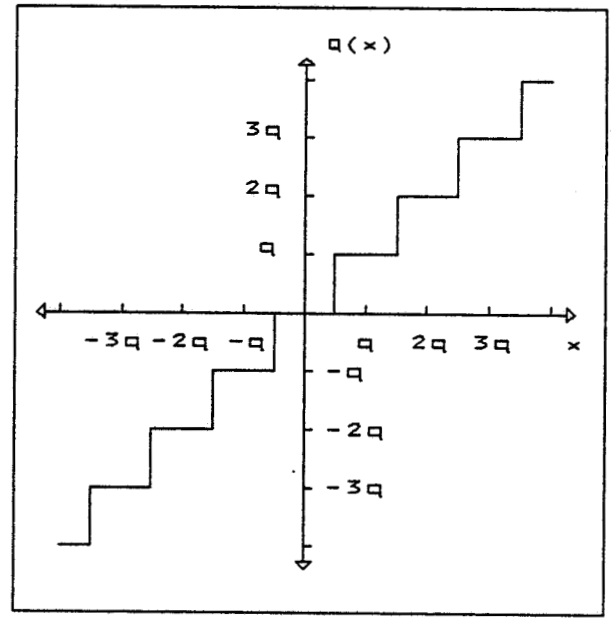


Figure I.4 - *Caractéristique de l'arrondi.*

Nous avons donc pour l'opération de troncature Q_t :

$$\text{si} \quad n.q < x \leq (n+1).q \qquad Q_t(x) = n.q$$

et pour l'opération d'arrondi Q_a :

$$\text{si} \quad (n-\tfrac{1}{2}).q < x \leq (n+\tfrac{1}{2}) \qquad Q_a(x) = n.q$$

I.2.3.2. Puissance de bruit pour la troncature et l'arrondi.

Pour calculer la valeur de la puissance de l'erreur due à la quantification soit encore la variance σ_e^2, on assimile généralement ce signal à un bruit de répartition d'amplitude uniforme sur l'intervalle de définition.

Dans le cas de la troncature, l'intervalle de définition de l'erreur est fonction du type d'arithmétique utilisée pour représenter les nombres. Cette dépendance est due à la dissymétrie de la caractéristique (figure I.3). Pour le cas de l'arrondi, la symétrie de la caractéristique (figure I.4) garantie l'indépendance vis à vis de l'arithmétique choisie. La figure I.5 donne sous forme comparative pour trois types d'arithmétiques, les fonctions respectives de densité de probabilité de l'erreur pour les techniques de troncature et d'arrondi.[11]

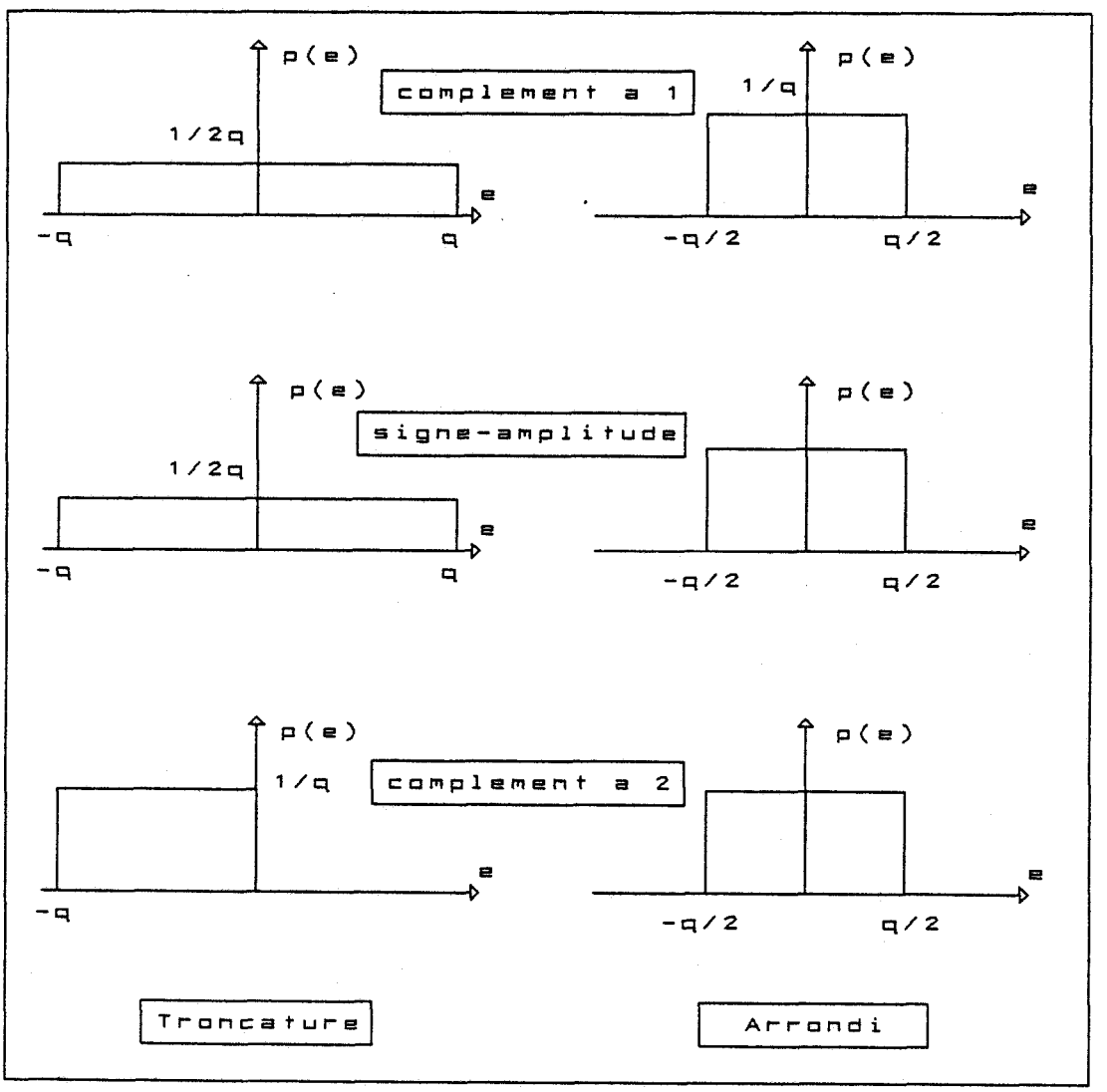


Figure I.5 - *Fonctions densité de probabilité de l'erreur pour trois types d'arithmétiques.*

A partir de ces fonctions de densité de probabilité il est possible de calculer les différentes variances σ_e^2 de l'erreur produite par l'opération de quantification. Ces valeurs sont regroupées dans le tableau 1.

	Troncature	Arrondi
Complément à 2	$\sigma_e^2 = q^2/12$	$\sigma_e^2 = q^2/12$
Complément à 1	$\sigma_e^2 = q^2/3$	$\sigma_e^2 = q^2/12$
Signe-Amplitude	$\sigma_e^2 = q^2/3$	$\sigma_e^2 = q^2/12$

Tableau 1. *Variance de l'erreur en fonction de l'arithmétique et du type de quantification.*

On constate que la technique de l'arrondi est moins bruyante que celle de la troncature et que de plus, sa puissance de bruit qui vaut $q^2/12$ est indépendante du type d'arithmétique utilisée pour représenter les nombres.

I.2.3.3. Rapport signal sur bruit.

Pour un quantificateur uniforme, si on appelle N le nombre de niveaux de décisions, les deux niveaux correspondants au cas de dépassement de capacité inclus, la valeur du quantum de base q est donnée par la relation :

$$(N-2).q = 2.A_m \tag{1.7}$$

où A_m représente l'amplitude maximale du quantificateur. On peut encore exprimer cette relation en fonction du facteur de crête F_c en supposant de plus que N est très supérieur à 2 :

$$q = \frac{2.\sigma}{N}\, e^{\left[\frac{F_c.\mathrm{Log}(10)}{20}\right]} \tag{1.8}$$

Soit encore :

$$q = \frac{2.\sigma}{N}\, e^{(\,0.115.F_c\,)} \tag{1.9}$$

Dans le cas de l'arrondi, le rapport signal sur bruit est donné par l'expression :

$$SNR = 10.\log\left[\frac{\sigma^2}{\left[\frac{q^2}{12}\right]}\right] \qquad (1.10)$$

Soit en remplaçant q par sa valeur en fonction de F_c

$$SNR = \frac{10}{Log(10)}\left[\, Log(3) + 2.Log(N) - 0.23.F_c \,\right] \qquad (1.11)$$

Si l'on prend maintenant $N = 2^n$ ce qui correspond au cas d'un quantificateur par arrondi de n-1 bits, on retrouve une expression classique [12] :

$$SNR = 6.02.n - 0.998.F_c + 4.77 \qquad (1.12)$$

Dans le cas d'un quantificateur par troncature un calcul analogue (pour une arithmétique signe-amplitude par exemple) conduit à l'expression :

$$SNR = 6.02.n - 0.998.F_c - 1.249 \qquad (1.13)$$

Le rapport signal sur bruit va donc varier linéairement en fonction à la fois du nombre n de bits et du facteur de crête F_c. Les fonctions SNR de n et de F_c pour la troncature et l'arrondi sont donc des plans parallèles, ce qui revient à dire que quelques soient n et F_c donnés, la technique de l'arrondi donne un meilleur rapport signal sur bruit que celle de la troncature, ceci compte tenu des origines respectives des deux plans.

I.2.3.4. Introduction du bruit de saturation.

Les formules rappelées précédemment ne font pas intervenir la notion de bruit de saturation. C'est à dire du bruit résultant du dépassement de capacité du quantificateur. D'une manière plus générale, le rapport signal sur bruit doit donc être exprimer sous la forme :

$$SNR = 10.\ \log\left[\frac{\sigma^2}{\sigma_e^2 + \sigma_s^2}\right] \qquad (1.14)$$

où σ_s^2 représente la puissance du bruit s de saturation.

Ce paramètre peut devenir prépondérant lors de la caractérisation d'un quantificateur si la dynamique de ce dernier n'est plus en correspondance avec celle du signal qui lui est appliqué. Ainsi, la figure I.6 représente une comparaison entre les valeurs que prend la fonction SNR = f(σ) suivant que l'on prend en compte ou non le bruit s de saturation (courbe II et I respectivement). C'est à dire en fait, la robustesse du quantificateur vis à vis des variations de la puissance du signal d'entrée. Ces courbes correspondent au cas d'un quantificateur par arrondi pour lequel $A_m = 2^{\frac{1}{2}}$ et n = 16 au quel on applique un signal d'entrée x de densité de probabilité gaussienne à moyenne nulle et de variance σ^2. Elles ont été obtenues à partir de la formule précédente en prenant respectivement σ_s^2 nul (courbe I) et σ_s^2 non nul (courbe II). Dans ce cas, la valeur de σ_s^2 est donnée par la relation suivante :

$$\sigma_s^2 = (\sigma^2 + A_m^2) \cdot \operatorname{erfc}\left[\frac{A_m}{\sigma.\sqrt{2}}\right] - \frac{\sigma.A_m.\sqrt{2}}{\sqrt{\pi}} . e^{\frac{-A_m^2}{2.\sigma^2}} \qquad (1.15)$$

(voir le détail des calculs en annexe 1).

A partir de cette même formule, il est possible de déterminer une relation donnant le maximum du rapport signal sur bruit en fonction de σ pour A_m et n fixés. Dans le cas du signal de densité de probabilité gaussienne on aboutit à la relation suivante :

$$\frac{q^2}{12} + A_m^2 \cdot \operatorname{erfc}\left[\frac{A_m}{\sigma.\sqrt{2}}\right] = \frac{\sigma.A_m.\sqrt{2}}{\sqrt{\pi}} \cdot e^{\frac{-A_m^2}{2.\sigma^2}} \qquad (1.16)$$

Cette relation permet donc, pour A_m et n donnés de déterminer quelle valeur de la variance du signal gaussien va fournir le meilleur rapport signal sur bruit. Sur l'exemple précèdent de la figure I.6, pour n = 16 et $A_m = 2^{\frac{1}{2}}$, la relation donne une valeur de σ de 0.225. En pratique cette relation permet surtout d'avoir la limite supérieure de σ au delà de laquelle il n'est plus possible de négliger l'influence du bruit de saturation dans le calcul du SNR, ceci pour A_m et n fixés.

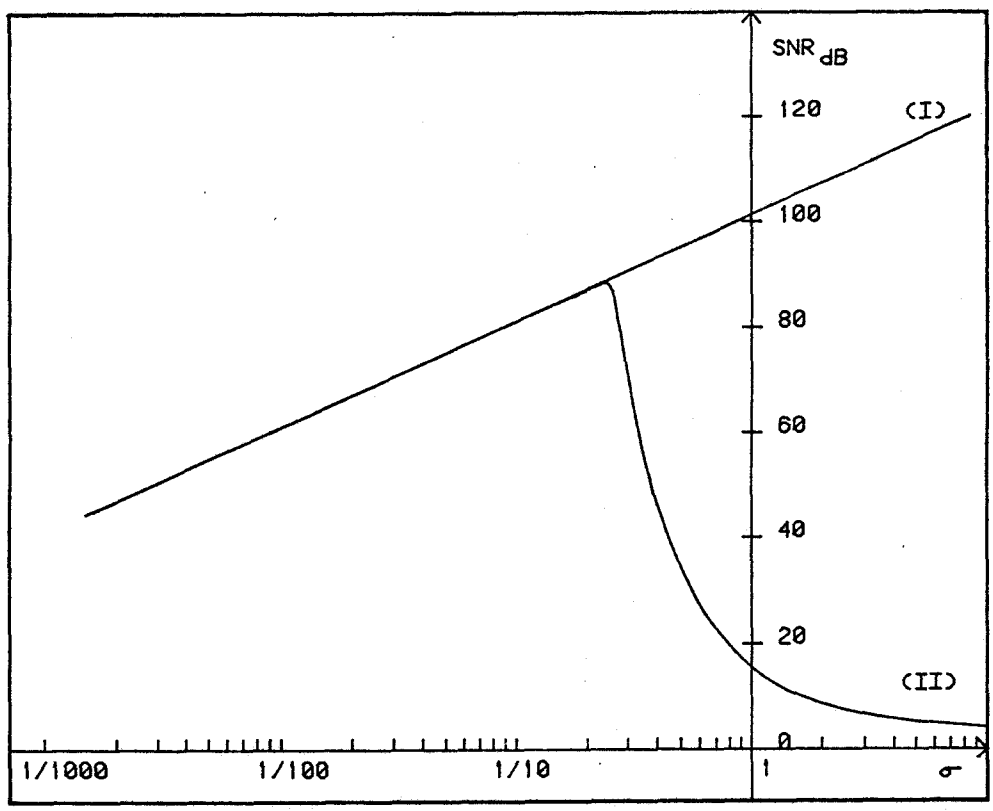


Figure I.6 - *SNR = f(σ) en fonction de σ_s.*

La figure I.7 illustre cette remarque pour $A_m = 1$ et n variant de 3 à 32 bits. A chaque valeur de n correspond la valeur de σ du signal gaussien qui donne le meilleur rapport signal sur bruit en sortie d'un quantificateur uniforme par arrondi. Cette valeur

correspond aussi à la limite d'influence du bruit de saturation. On constate par exemple que pour pouvoir négliger l'effet de ce bruit sur le résultat d'une quantification uniforme sur 16 bits d'un signal gaussien centré, il est nécessaire d'avoir la racine carrée de la variance de ce signal inférieure à 0.24 et que le rapport signal sur bruit vaudra au maximum dans ce cas 90 dB.

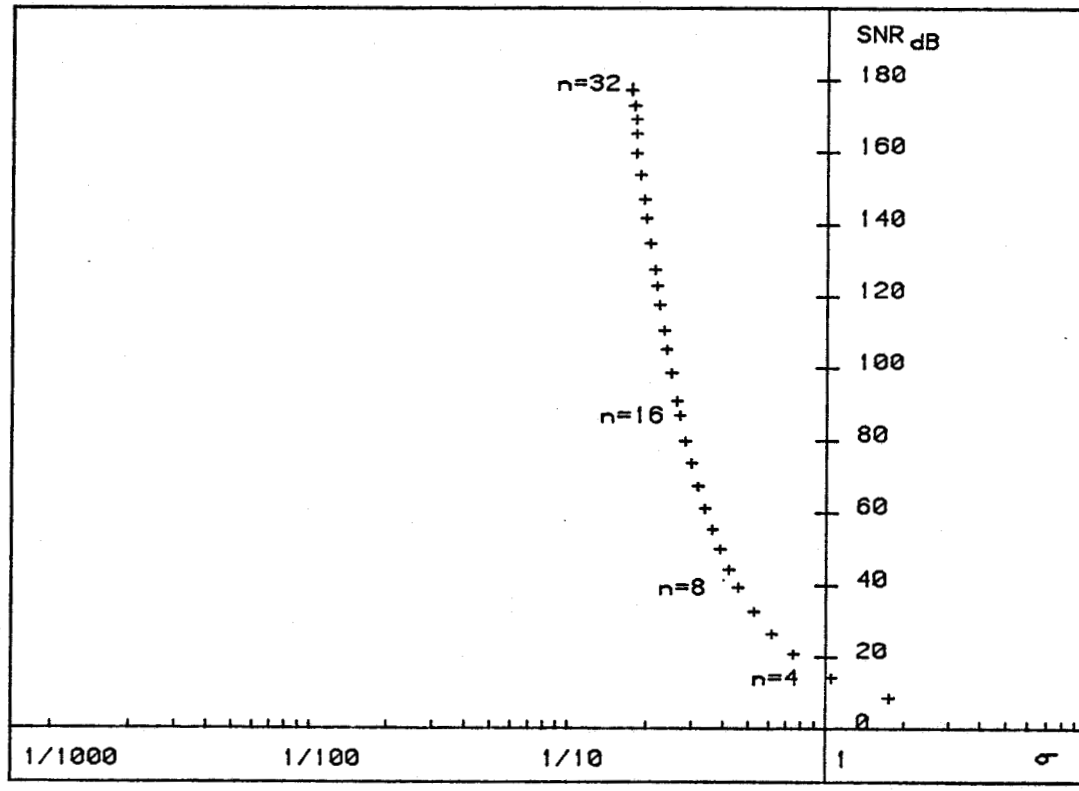


Figure I.7 - *Valeur maximale de SNR = f(σ) en fonction de n pour* $A_m = 1$.

Il résulte à posteriori, que du point de vue de la quantification, l'approximation sur 16 bits d'un signal gaussien centré par une fonction quantifiée de ce type qui prend ses valeurs dans [-1,+1] n'est acceptable qui si la racine carrée de la variance de ce signal est inférieure à 0.24.

I.2.4. Quantification non uniforme.

La quantification uniforme a l'avantage d'une mise en oeuvre aisée. Pour réaliser un arrondi il suffit par exemple de faire une simple troncature du nombre après avoir augmenté d'une unité le premier bit tronqué. Cependant ce type de quantificateur n'est à proprement parler bien adapté qu'aux signaux dont les densités de probabilité sont uniformes. Dans le cas contraire les performances de la quantification sont

dégradées. Pour ce types de signaux on peut alors utiliser des quantificateurs non uniformes afin d'obtenir de meilleurs résultats.

I.2.4.1. Modélisation d'un quantificateur scalaire non uniforme.

Il est possible de décomposer tout quantificateur non uniforme suivant le schéma de modélisation suivant [13] :

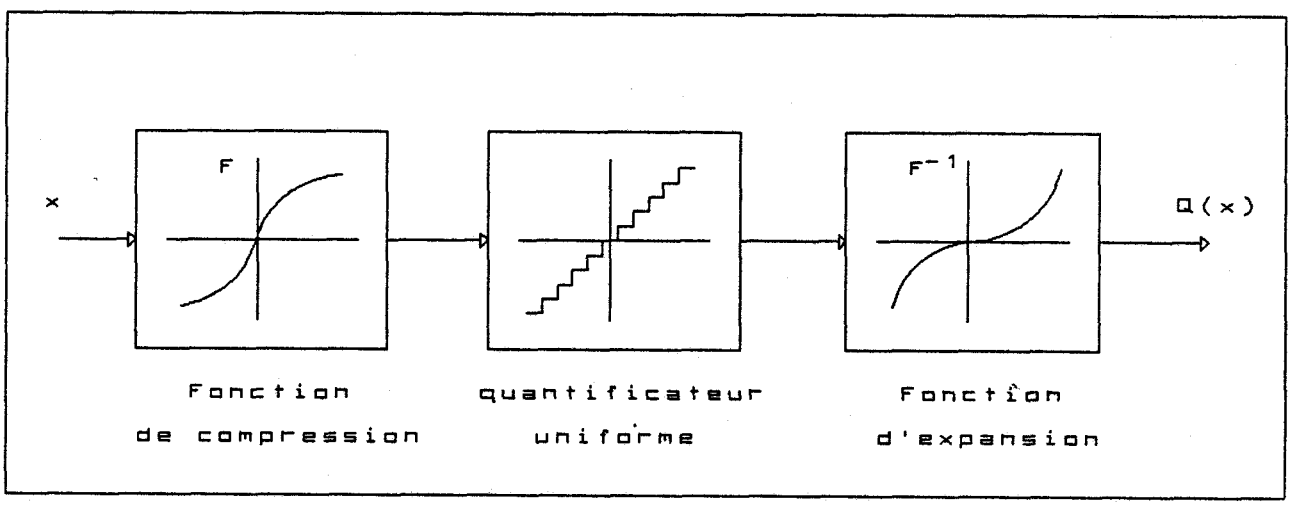


Figure I.8 - *Modélisation d'un quantificateur non uniforme.*

où F(x) est généralement appelée fonction ou loi de compression et $F^{-1}(x)$ fonction d'expansion. Naturellement cette fonction doit être bijective afin qu'il n'y ai pas d'ambiguïté dans la définition de F^{-1} et elle présentera de plus une propriété de symétrie impaire. L'étude d'un quantificateur non uniforme se réduit donc à celle de la fonction F.

I.2.4.2. Exemples de lois de compression.

Le quantificateur non uniforme est souvent très utile lorsque l'on cherche à améliorer la propriété de robustesse dans une opération de quantification. On montre [10] que le choix d'une loi de compression de la forme :

$$F(x) = A_m + c.\mathrm{Log}\left(\frac{x}{A_m}\right) \qquad \text{pour } x > 0 \tag{1.17}$$

où c est une constante et Am l'amplitude maximale du quantificateur, conduit à un rapport signal sur bruit indépendant de la densité de probabilité du signal d'entrée si l'on néglige le bruit de saturation.

Cette constatation trouve son application dans le domaine de la transmission de la parole où le même quantificateur doit traiter des signaux de niveaux de puissance très variables. En pratique une telle loi de compression n'est toutefois pas réalisable puisque par exemple F(0) n'est pas fini. Aussi utilise - t'on par approximation des lois de type logarithmique. Les deux lois principales sont :

- la loi en μ donnée par les relations :

$$F(x) = A_m \cdot \frac{\log\left[1 + \frac{\mu . x}{A_m}\right]}{\log(1 + \mu)} \qquad (1.18)$$

avec $F(x) = -F(-x)$

la valeur couramment retenue pour le paramètre μ en transmission de la parole est $\mu = 100$ pour des quantificateurs de 7 bits et $\mu = 255$ pour des quantificateurs de 8 bits.

- la loi en A [14] donnée par les relations :

$$F(x) = \begin{cases} \dfrac{A \cdot x}{1 + \log(A)} & \text{pour} \quad 0 \le x \le \dfrac{A_m}{A} \\ \dfrac{A_m + A_m . \log\left[\frac{A.x}{A_m}\right]}{1 + \log(A)} & \text{pour} \quad \dfrac{A_m}{A} \le x \le A_m \end{cases} \qquad (1.19)$$

le paramètre A prenant une valeur typique de 87.6 pour un quantificateur de 7 bits.

A titre d'exemple la robustesse d'un quantificateur 7 bits à loi de compression en μ avec $\mu = 255$ soumis à des signaux laplaciens est obtenue avec un quantificateur uniforme si ce dernier travaille sur 11 bits. Le quantificateur non uniforme permet donc dans ce cas précis de réaliser un gain de 4 bits par échantillon.

Dans la pratique, ces quantificateurs sont obtenus au moyen de fonctions d'approximations linéaires. On utilise ainsi les quantificateurs à loi en A à 13 segments et à loi en μ à 15 segments.

I.2.5. Quantification scalaire optimale.

La qualité de robustesse d'un quantificateur n'est vraiment importante que si ce dernier est destiné à traiter des signaux de densité de probabilité variable. Dans le cas où la densité de probabilité du signal d'entrée est bien connue et fixe, il est généralement plus intéressant "d'adapter" le quantificateur au signal d'entrée de manière à obtenir une performance de quantification maximale. Le critère à optimiser étant en général du type erreur quadratique moyenne, il s'agit donc de minimiser la valeur suivante :

$$E = \sum_{i=1}^{N} \int_{X(i-1)}^{X(i)} [\, Y(i) - x \,]^2 \, p(x) \, dx \qquad (1.20)$$

si x représente le signal d'entrée du quantificateur Q.

En pratique ce problème n'est pas trivial puisqu'il s'agit de déterminer conjointement l'ensemble des niveaux de décision { X(i) } et celui des niveaux de reconstruction { Y(j)} (confère § I.2.1). Toutefois quand le nombre N de niveaux de quantification est important il est possible d'arriver à une solution explicite [15] moyennant quelques hypothèses.

I.2.5.1. Quantificateur optimal à grand nombre de niveaux.

En effet si N est grand on peut admettre que sur chaque niveau de quantification, la densité de probabilité de x est sensiblement constante et vaut p[Y(j)]. Dans ce cas E peut s'exprimer sous la forme :

$$E = \sum_{i=1}^{N} p\,[\,Y(i)\,] \cdot \int_{X(i-1)}^{X(i)} [\,Y(i) - x\,]^2\,dx \qquad (1.21)$$

soit encore ici :

$$E = \frac{1}{3} \sum_{i=1}^{N} p\,[\,Y(i)\,] \cdot \left[\,[\,Y(i) - X(i)\,]^3 - [\,Y(i) - X(i-1)\,]^3\,\right] \qquad (1.22)$$

Les valeurs optimales des niveaux de reconstruction Y(i) sont alors données par la résolution des N équations aux dérivées partielles :

$$\frac{\partial E}{\partial Y(i)} = 0 \qquad (1.23)$$

ce qui conduit à des solutions de la forme :

$$Y(i) = \frac{X(i) - X(i-1)}{2} \qquad (1.24)$$

et à une valeur de E donnée par :

$$E = \frac{1}{12} \cdot \sum_{i=1}^{N} p\,[\,Y(i)\,] \cdot [\,X(i) - X(i-1)\,]^3 \qquad (1.25)$$

Les valeurs des X(i) sont obtenues en minimisant E par la méthode des multiplicateurs de Lagrange [16] en supposant X(0) et X(N) connus.

$$X(i) = \frac{[\,X(N) - X(0)\,] \cdot \int_{X(0)}^{a_i} p(x)^{-1/3}\,dx}{\int_{X(0)}^{X(N)} p(x)^{-1/3}\,dx} \qquad (1.26)$$

$$\text{avec} \quad a_i = \frac{i \cdot [\,X(N) - X(0)\,]}{N} + X(0)$$

Cette technique donne par exemple dans le cas d'un signal d'entrée de densité de probabilité de Laplace, un quantificateur optimal dont la loi de compression est de la forme [17] :

$$F(x) = \frac{A_m \cdot (\,1 - e^{-mx}\,)}{1 - e^{-mA_m}} \qquad \text{pour} \quad x > 0 \qquad (1.27)$$

I.5.2.2. Quantificateurs optimaux à nombre de niveaux quelconque.

si maintenant on n'admet plus l'hypothèse de N grand, il convient de reprendre les calculs précédents. Les équations aux dérivées partielles conduisent alors au système d'équations suivant [18] :

$$Y(i) = 2 \cdot X(i) - Y(i-1)$$

$$Y(i) = \frac{\int_{X(i-1)}^{X(i)} x \cdot p(x)\,dx}{\int_{X(i-1)}^{X(i)} p(x)\,dx} \qquad (1.28)$$

qui fournit une condition nécessaire pour l'obtention d'un quantificateur optimal .

Remarque : cette condition n'est pas suffisante pour garantir l'optimalité. Une condition suffisante a été donnée [19] qui s'exprime sous la forme :

$$\frac{d^2}{dx^2} \left[\log [p(x)] \right] < 0 \qquad (1.29)$$

La résolution du système précédent nécessite évidemment le recours à une méthode d'approximation itérative. Des solutions ont été tabulées pour le cas de densité de probabilité gaussienne [20] ou encore de densité de Laplace et Gamma [21]. Une table donnant les niveaux de décision et de reconstruction pour les densités uniformes, gaussiennes, laplaciennes et Rayleigh est présentée en annexe 2. L'expression de E_{min} est alors donnée par la relation :

$$E_{min} = \sum_{i=1}^{N} \left[\int_{X(i-1)}^{X(i)} x \, . \, p(x) \, dx - Y(i)^2 \, . \int_{X(i-1)}^{X(i)} p(x) \, dx \right] \qquad (1.30)$$

I.2.5.3. Approche par les lois de compression.

Compte tenu de la décomposition possible d'un quantificateur non uniforme au moyen de la technique de compression-expansion vue précédemment, une méthode pour adapter un quantificateur au signal x qu'il doit traiter et d'essayer de trouver la loi de compression F qui transforme la densité de probabilité de x en une densité uniforme.

Pour déterminer cette fonction F on a recourt à la théorie des fonctions de variables aléatoires.

Les propriétés des fonctions de variables aléatoires nous permettent de dire que si la fonction F est monotone alors si :

$$u = F(x)$$

et si p(x) représente la densité de probabilité de la variable x, la densité q de probabilité de la variable u est donnée par la relation :

$$q(u) = p\left[F^{-1}(u) \right] . \left| \frac{dx}{du} \right|$$

Or dans le cas présent, la fonction F est une fonction de compression donc toujours croissante sur son intervalle de définition. Le problème se ramène donc ici à trouver F telle que q(u) soit constante.

$$q(u) = \text{cste.} = \frac{1}{c}$$

La relation précédente peut encore s'écrire :

$$q(u) = p(x) . \left| \frac{dx}{dF(x)} \right| \qquad \text{car } u = F(x)$$

d'où puisque F est croissante, F'est toujours positive et la relation devient :

$$q(u) = \frac{p(x)}{| F'(x) |} = \frac{1}{c} = \frac{p(x)}{F'(x)}$$

d'où $F'(x) = c.p(x)$ ce qui conduit après intégration à :

$$F(x) = c \cdot \int_0^x p(v)\, dv \qquad (1.32)$$

c'est à dire que F est à une constante multiplicative près, la fonction de répartition de x. Cette technique donne des résultats tout à fait corrects. Le tableau 2 fournit les lois de compression correspondante aux fonctions de densité de probabilité gaussienne, laplacienne et Rayleigh [15].

LOI	p(x)	F(x)
Gauss	$\frac{1}{\sigma\sqrt{2.\pi}} . \exp\left[- \frac{x^2}{2.\sigma^2} \right]$	$\frac{1}{2} . \mathrm{erf}\left[\frac{x}{\sqrt{2.\sigma}} \right]$
Rayleigh	$\frac{x}{\sigma^2} . \exp\left[- \frac{x^2}{2.\sigma^2} \right]$	$\frac{1}{2} - \exp\left[- \frac{x^2}{2.\sigma^2} \right]$
Laplace	$\frac{\alpha}{2} . \exp\left[- \alpha \lvert x \rvert \right]$ avec $\alpha = \frac{\sqrt{2}}{\sigma}$	$\frac{1}{2} . \left[1 - \exp\left[- \alpha . x \right] \right]$ si $x \geq 0$; $- \frac{1}{2} . \left[1 - \exp\left[\alpha . x \right] \right]$ si $x < 0$

Tableau 2. *Lois de compression pour 3 types de densité de probabilité.*

I.3. QUANTIFICATION VECTORIELLE.

La deuxième grande classe de quantificateur est celle des quantificateurs vectoriels. Comme on l'a vu, la quantification scalaire d'une séquence d'échantillons est accomplie sur une base séquentielle. Chaque échantillon de la séquence est traité comme une variable scalaire et quantifié suivant des régles décrites précédemment. Il est toutefois possible de traiter conjointement un groupe d'échantillons successifs ou vecteur, de la séquence. Cette technique en plein développement trouve son application principale dans le cadre des transmissions, plus spécialement de la parole et des images [22].

I.3.1. Principe de la quantification vectorielle.

La quantification vectorielle peut être définie comme une application Q d'un espace de dimension k noté R^k; dans un sous-espace fini Y de cet espace. Ceci implique donc nécessairement une subdivision de l'espace de départ en un certain nombre de régions R_i qui réalisent une partition de l'ensemble R^k. Chaque région R_i étant définie par la relation :

$$R_i = \{ x \in R^k / Q(x) = y_i \} \qquad (1.33)$$

si y_i est un élément du sous-ensemble d'arrivée Y.

Un quantificateur vectoriel est donc complètement défini par l'ensemble des valeurs d'arrivée $Y = \{ y_1, y_2, \ldots, y_n \}$ et l'ensemble $\{ R_1, R_2, \ldots, R_n \}$ des régions de décisions qui sont associées respectivement aux y_i. Il convient cependant de faire une distinction supplémentaire. En effet, un quantificateur qu'il soit vectoriel ou non fournit une seule information mais celle-ci est disponible sous deux formes (la valeur de sortie y et l'index i qui la repère) suivant différentes possibilités [23] comme le montre la figure I.9.

Les problèmes fondamentaux à résoudre sont alors les suivants :

- Rechercher pour un vecteur (ou point) donné, le vecteur de sortie le plus proche au sens d'une distance à définir.

- Rechercher l'index d'un vecteur de sortie.

- Rechercher un point de sortie à partir de son index.

Ceci suppose naturellement une connaissance de la " géométrie " des points de sortie.

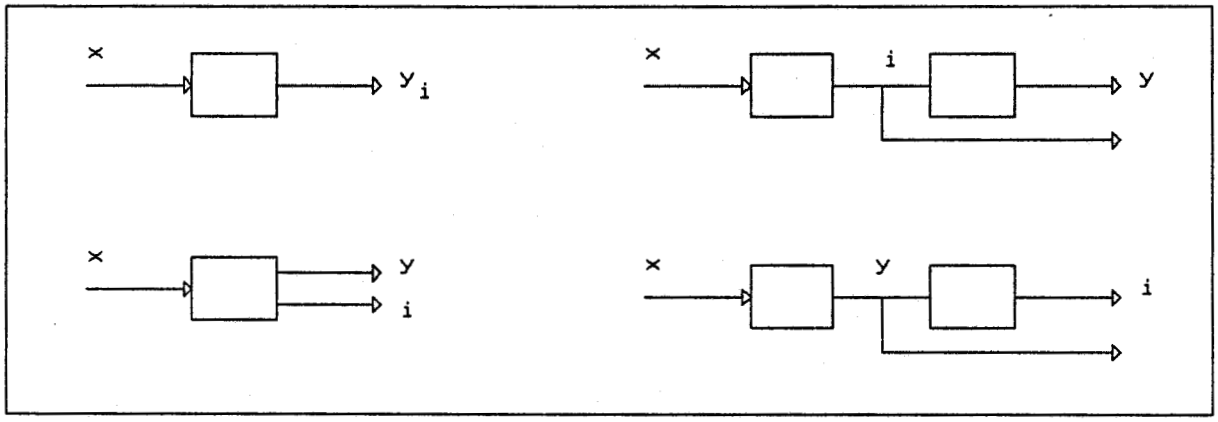


Figure I.9 - *Diagramme représentant différentes possibilités de sortie de l'information d'un quantificateur.*

I.3.2. Caractérisation d'un quantificateur vectoriel.

Dans la pratique un quantificateur vectoriel peut être décomposé en deux fonctions séparées, le codeur C et le décodeur D tels que :

$$Q = D \circ C$$

Cette décomposition peut toutefois être encore affinée [24]. Pour cela on définit un ensemble de fonctions S_i telles que :

$$S_i(x) = a_i \qquad \text{avec} \qquad a_i = \begin{cases} 1 & \text{si } x \in R_i \\ 0 & \text{si } x \notin R_i \end{cases} \tag{1.34}$$

Il résulte de ceci que l'opération de quantification Q peut se mettre sous la forme :

$$Q(x) = \sum_{i=1}^{n} y_i \cdot S_i(x) = \sum_{i=1}^{n} y_i \cdot a_i \tag{1.35}$$

On peut alors introduire une fonction F de génération d'index telle que :

$$F : \{0;1\}^n \longrightarrow J \qquad J : \text{ensemble des index} \tag{1.36}$$

$$(a_1\ a_2\ \ldots\ a_n) \longrightarrow j \qquad j : \text{le plus grand index tel que } a_i = 1$$

Cette fonction va faire correspondre au n-uple d'entrée (a_1, a_2, ... , a_n) l'indice j de la composante a_j qui est non nulle et qui caractérise le fait que le vecteur x appartient à la région de décision R_j. A partir de cela, le codeur C peut donc être décomposé suivant le schéma de la figure I.10 :

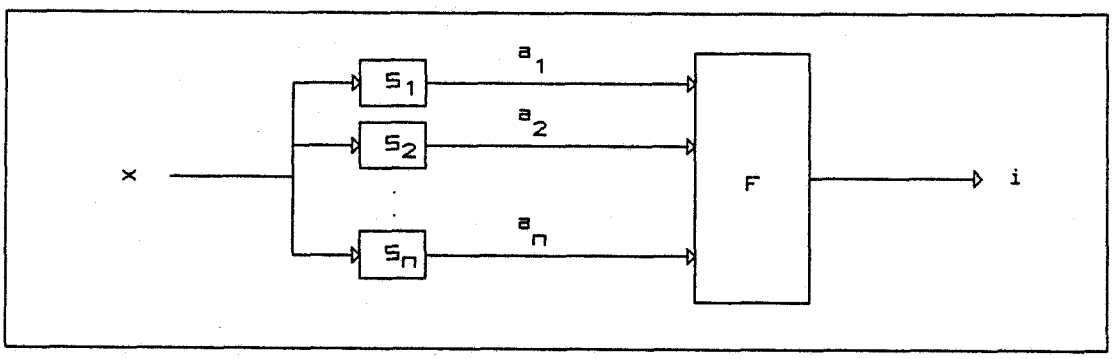


Figure I.10 - *Codeur.*

Il est possible de décomposer le décodeur de façon analogue. Pour cela on pose G la fonction de génération d'adresses, définie par :

$$G : J \longrightarrow \{0;1\}^n \tag{1.37}$$

$$j \longrightarrow (\delta_{1j}\ \delta_{2j}\ \ldots\ \delta_{nj}) \qquad \text{avec } \delta_{ij} \text{ le symbole de Kronecker}$$

et le décodeur D est alors donné par la relation :

$$D = \sum_{i=1}^{n} y_i \,.\, G(i) \qquad \text{tel que } D(j) = y_j \tag{1.38}$$

que l'on peut représenter par le schéma de la figure I.11.

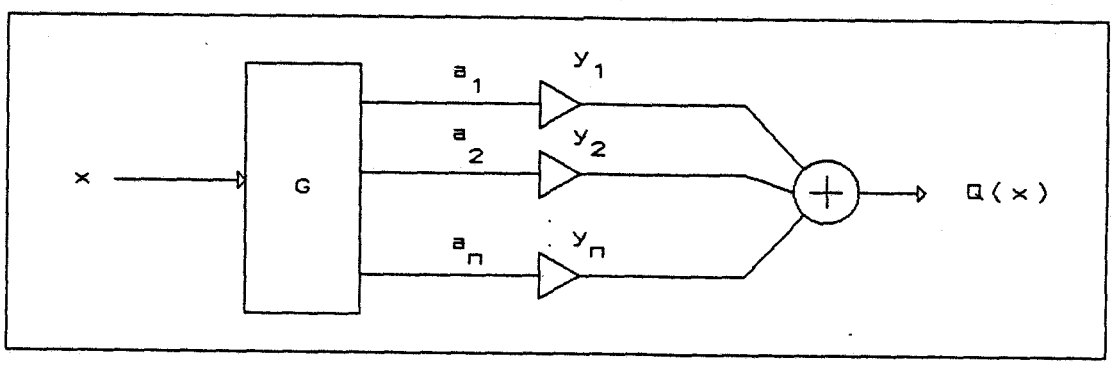


Figure I.11 - *Décodeur.*

Ce qui conduit à une structure générale du quantificateur qui à la forme suivante :

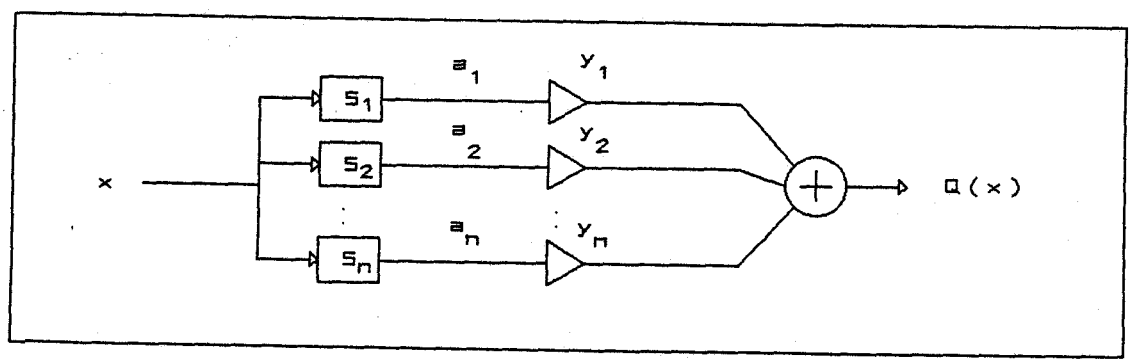


Figure I.12 - *Décomposition d'un quantificateur vectoriel.*

I.3.3. Intérêt de la quantification vectorielle.

La réalisation d'un quantificateur vectoriel est bien entendu plùs complexe que celle d'un quantificateur scalaire. L'utilisation de telles techniques ne se justifie donc que par un accroissement des performances de quantification. La diminution de l'erreur quadratique moyenne est l'une d'entre elles. Intuitivement, il est aisé de comprendre qu'il doit exister un quantificateur vectoriel donnant des résultats au moins

aussi bons que le meilleur des quantificateurs scalaires puisque ce dernier ne fait absolument pas intervenir une éventuelle corrélation entre les différents échantillons du signal à traiter. Ceci s'explique en observant le fait qu'un ensemble de N quantificateurs scalaires est équivalent à un quantificateur vectoriel particulier, c'est à dire dont le dictionnaire Y a une "géométrie" particulière, qui n'est pas forcément celle du meilleur quantificateur vectoriel possible.

Le problème de la recherche de quantificateurs vectoriels optimaux se pose donc également et il est possible de généraliser les équations données dans le cas des quantificateurs scalaires. Le problème se traduit alors en termes de définition de la géométrie des vecteurs de sortie qui forment le dictionnaire Y. L'hyper-espace correspondant est donc constitué d'un ensemble de points ou réseau qui représente les sommets des vecteurs de sortie. Les performances du quantificateur étant en relation directe avec la géométrie du réseau . Une hypothèse émise par A. Gersho veut que pour toute dimension il existe un réseau régulier qui atteint les performances optimales. Cette hypothèse jamais démentie jusqu'à présent n'a été vérifiée que pour les dimensions 2 et 3. Il reste possible par ailleurs de trouver des quantificateurs vectoriels non optimaux mais dont les performances sont supérieures à celles des quantificateurs scalaires. Il a été décrit [25] des quantificateurs vectoriels localement optimaux pour des signaux ayant des densités de probabilité de Gauss, Laplace ou Gamma, qui diminuent d'un facteur de 2 à 4.5 dB l'erreur quadratique moyenne par rapport aux quantificateurs scalaires optimaux correspondants. Enfin, un certain nombre de quantificateurs vectoriels particuliers fournissent d'assez bons résultats suivant le signal à traiter. Citons par exemple la technique de la quantification vectorielle sphérique qui consiste à quantifier séparément la norme et la phase du vecteur d'entrée. Il a été mis au point [26] un algorithme rapide qui applique cette technique sur un réseau de type Gosset d'ordre 8.

I.4. CONCLUSION.

Naturellement il convient de trouver un compromis entre les résultats souhaités au terme de la quantification et la complexité de mise en oeuvre du quantificateur. La technique de quantification vectorielle qui fournit d'excellents résultats semble plus adaptée au problème de la réduction des débits binaires en transmission numérique qu'au problème du filtrage pur. De façon analogue, la quantification séquentielle que nous n'avons pas particulièrement développé ici, reste une technique propre à des problèmes de type adaptatif tels que le codage MIC-Delta Adaptatif. Finalement, la technique de quantification scalaire même si elle est loin d'être parfaite reste d'un bon rapport compte tenu de la relative aisance de sa mise en oeuvre.

CHAPITRE II

II.1. INTRODUCTION.

La quantification est le principe qui est à l'origine des sources de bruit qui existent dans les structures numériques telles que les filtres. La technique utilisée détermine comme on l'a vu, la puissance de bruit de chacune de ces sources. Toutefois lors de de la synthèse d'un filtre, c'est la valeur de la puissance de bruit global en sortie du filtre qu'il est intéressant de connaître.

Ce bruit résulte des différents bruits dus aux opérations arithmétiques intervenant au cours du calcul du filtre. C'est à dire principalement aux multiplications puisqu'il est possible de prévoir un facteur d'échelle à l'entrée pour réduire au minimum un éventuel dépassement de capacité en sortie des additionneurs; l'effet de quantification des coefficients n'étant pas pris en compte pour des filtres à coefficients fixes. Par la suite nous ne considérerons donc que les sources de bruit engendrées en sortie des multiplieurs, et le bruit global qu'elles produisent en sortie de la structure qui les contient.

II.2. ETUDE DU BRUIT GENERE PAR UN MULTIPLIEUR NUMERIQUE.

II.2.1. Modélisation d'un multiplieur numérique.

La multiplication avec résultat sur B bits d'un signal x quantifié sur B bits par un coefficient fixe a, génère une erreur e donnée par la relation :

$$e = [\, a \,.\, x_Q \,]_Q - a \,.\, x_Q \qquad [\ ]_Q : \text{valeur quantifiée} \qquad (2.1)$$

Les valeurs que prend le signal erreur e dépendent évidemment du type de quantification utilisée pour ramener le résultat sur le format fixé. On adoptera ici la technique de quantification scalaire par arrondi qui est très répandue compte tenu de ses nombreux avantages (indépendance vis à vis de l'arithmétique utilisée pour représenter les nombres, puissance de bruit réduite par rapport à la technique de troncature et mise en oeuvre relativement aisée). Dans ce cas, le signal e est appelé bruit d'arrondi dû à la multiplication.

Du point de vue de la modélisation, un tel multiplieur se présente donc comme la somme du résultat d'un multiplieur à précision infinie avec un bruit e suivant le schéma de la figure II.1.

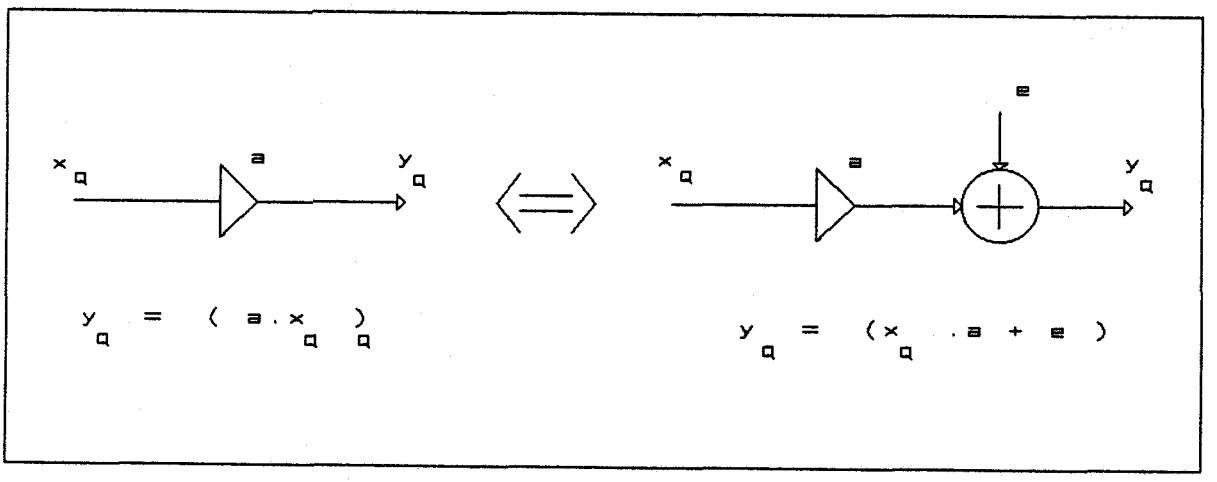


Figure II.1 - *Modélisation d'un multiplieur numérique.*

II.2.2. Puissance du bruit d'arrondi du à la multiplication.

Le bruit d'arrondi e présent à la sortie d'un multiplieur numérique est communément modélisé par un bruit blanc uniformément réparti sur l'intervalle

$$\left] -\frac{q}{2} , \frac{q}{2} \right[$$

où q représente le quantum de base du quantificateur et vaut 2^{-B} si l'on traite des mots binaires de B+1 bits. La puissance de bruit est alors donnée par la valeur de σ_e^2 suivant la formule :

$$\sigma_e^2 = \int_{-q/2}^{q/2} e^2 . p(e)\, de = \frac{q^2}{12} \qquad (2.2)$$

et ceci quelque soit l'arithmétique choisie (complément à 2, à 1, signe-amplitude ...) pour représenter les nombres comme on l'a vu au chapitre précédent.

II.2.3. Problème du bruit de saturation.

D'une manière générale on devrait aussi prendre en compte le bruit de saturation que peuvent générer les multiplieurs. La puissance de ce bruit dépend évidemment de la valeur du coefficient du multiplieur et du signal d'entrée x_Q.

Si l'on s'intéresse à un signal x_Q de densité de probabilité gaussienne, de variance σ^2 et de moyenne nulle, un calcul analogue à celui effectué au chapitre I (§ 2.3.4) conduit à une relation donnant la valeur de σ_s^2 en fonction du coefficient a du multiplieur, de la valeur maximale du quantificateur A_m et de σ.

$$\sigma_s^2 = (a^2\sigma^2 + A_m^2)\ .\ \mathrm{erfc}\left[\frac{A_m}{a\sigma\sqrt{2}}\right] - \frac{a\sigma . A_m . \sqrt{2}}{\sqrt{\pi}} . e^{\frac{-A_m^2}{2.\sigma^2 a^2}} \qquad (2.3)$$

Nous avons donc représenté le rapport signal sur bruit (SNR) en fonction de la racine carrée de la variance du signal d'entrée x_Q de densité de probabilité gaussienne, et ceci pour différentes valeurs du coefficient a. La quantification est faite sur 16 bits et A_m est pris égale à 1.

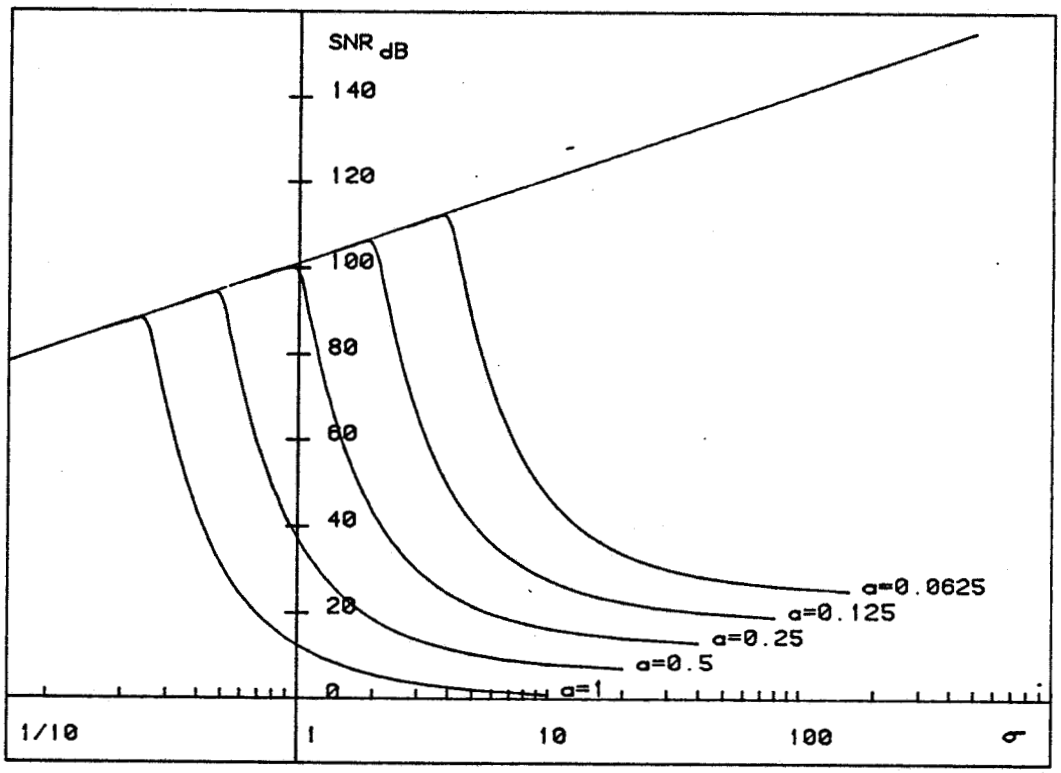


Figure II.2 - *Influence du bruit de saturation sur le SNR en fonction de a.*

On retrouve évidemment le fait que, a valeur de σ fixée, plus la valeur de a diminue, moins le bruit de saturation intervient. En particulier pour $A_m = 1$, un filtre travaillant sur 16 bits et dont tous les coefficients de ses multiplieurs sont inférieurs à 1 ne génère pas de bruit de saturation en sortie de ses multiplieurs s'il est soumis en entrée à un signal x_Q de densité de probabilité gaussienne de valeur σ inférieure à 0.24. Or nous savons d'après le chapitre I (§ 2.3.4) qu'une approximation d'un signal gaussien centré par un signal quantifié sur 16 bits variant dans l'intervalle [-1 , +1] n'est acceptable que si la racine carrée de la variance de ce signal est inférieure à 0.24. Nous utiliserons donc par la suite des signaux gaussiens centrés qui vérifient cette propriété. La génération de tels signaux est décrite en annexe 3 . Dans ces conditions, il est possible de considérer que la puissance de bruit se réduit à sa composante σ_e^2.

II.2.4. Simulation de la variance de l'erreur en fonction du coefficient.

La valeur $q^2/12$ de la variance du bruit d'arrondi en sortie d'un multiplieur suppose une indépendance de l'erreur vis à vis du coefficient multiplicatif a et du signal d'entrée x_Q. Nous avons donc simulé le fonctionnement d'un multiplieur afin de déterminer la validité et les limites de ces hypothèses. Pour cela, le signal d'entrée du multiplieur x_Q a été pris de densité de probabilité gaussienne dans l'intervalle] -1 , +1 [, de valeur moyenne nulle et d'écart type σ variable mais toujours inférieur à 0.24. Les simulations ont été faites pour un coefficient a variant dans le seul intervalle [0 , 1] puisque du fait de la symétrie et de la périodicité de la fonction erreur (figure II.3) il y aura identité du résultat sur l'intervalle [-1 , 0] . Enfin, le signal d'entrée x_Q et les résultats des multiplications sont quantifiés par arrondi sur 16 bits.

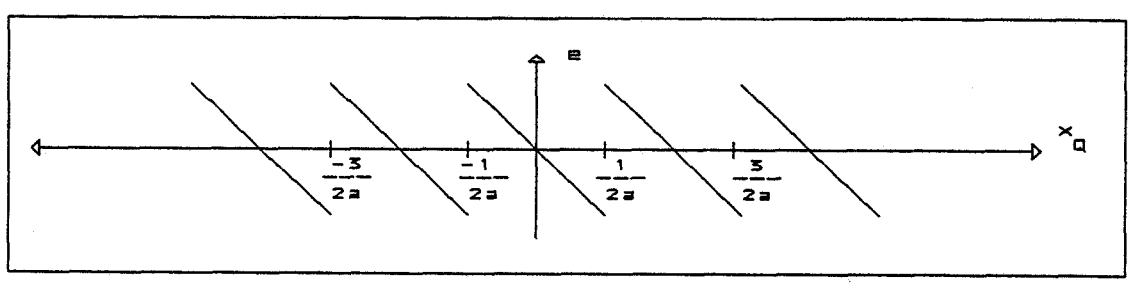


Figure II.3 - *Représentation de la fonction erreur d'un multiplieur numérique de coefficient a.*

II.2.5. Résultats des simulations.

La simulation permet d'obtenir la valeur expérimentale de la puissance de bruit d'arrondi σ_e^2. Il a été choisi, pour mettre clairement en évidence les phénomènes

de représenter la puissance de bruit réduite $12.\sigma_e^2/q^2$ en fonction du coefficient a du multiplieur; ceci pour les valeurs 10^{-4}, $2\ 10^{-4}$, $4\ 10^{-4}$ et 1/6 de σ_x l'écart type du signal d'entrée du multiplieur. Les résultats sont donnés par les figures II.4.a à II.4.d respectivement.

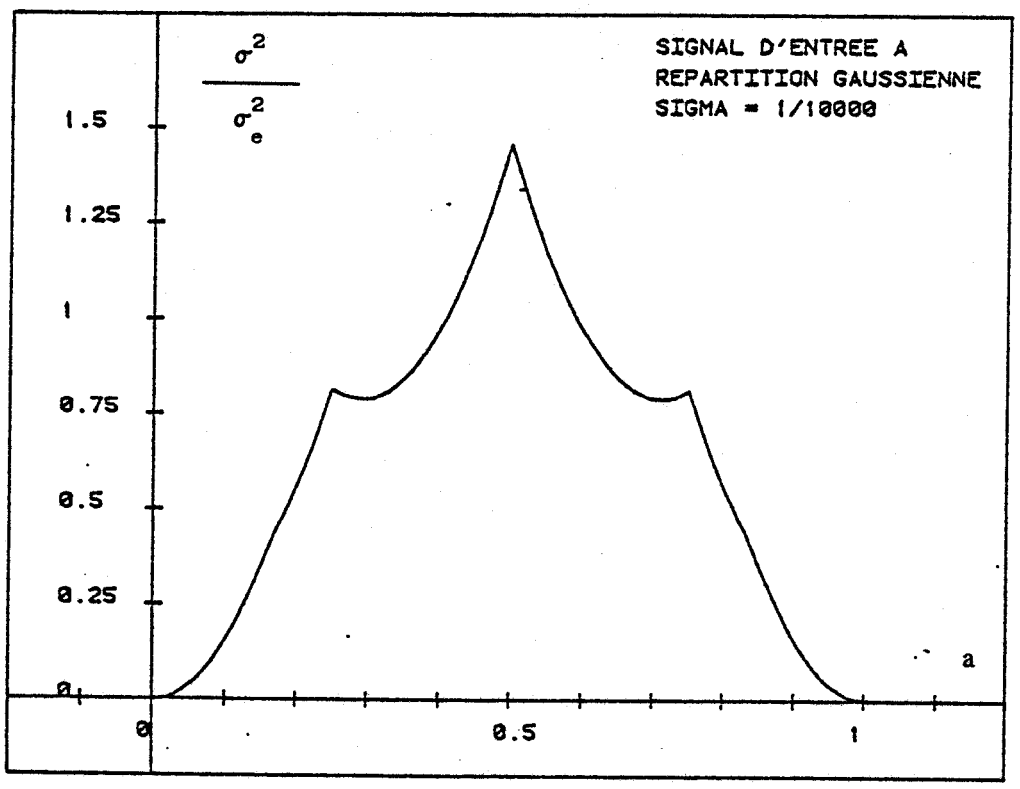


Figure II.4.a - *Puissance réduite de bruit en fonction de a pour* $\sigma_x = 10^{-4}$.

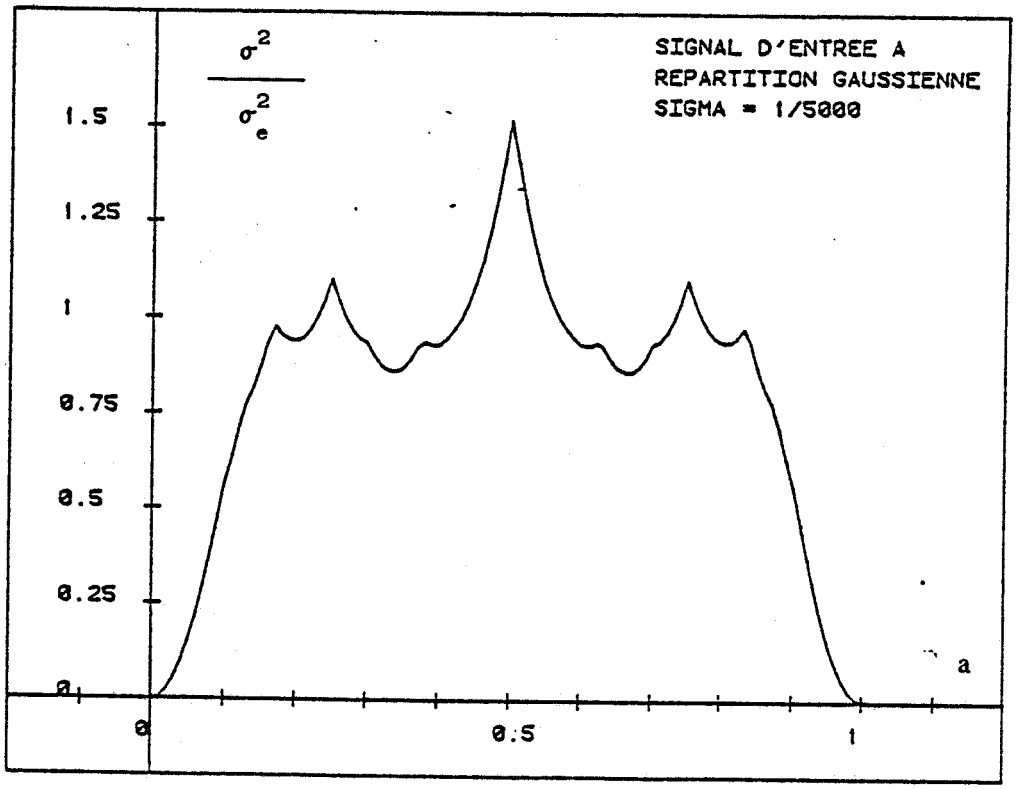


Figure II.4.b - *Puissance réduite de bruit en fonction de a pour* $\sigma_x = 2\ 10^{-4}$.

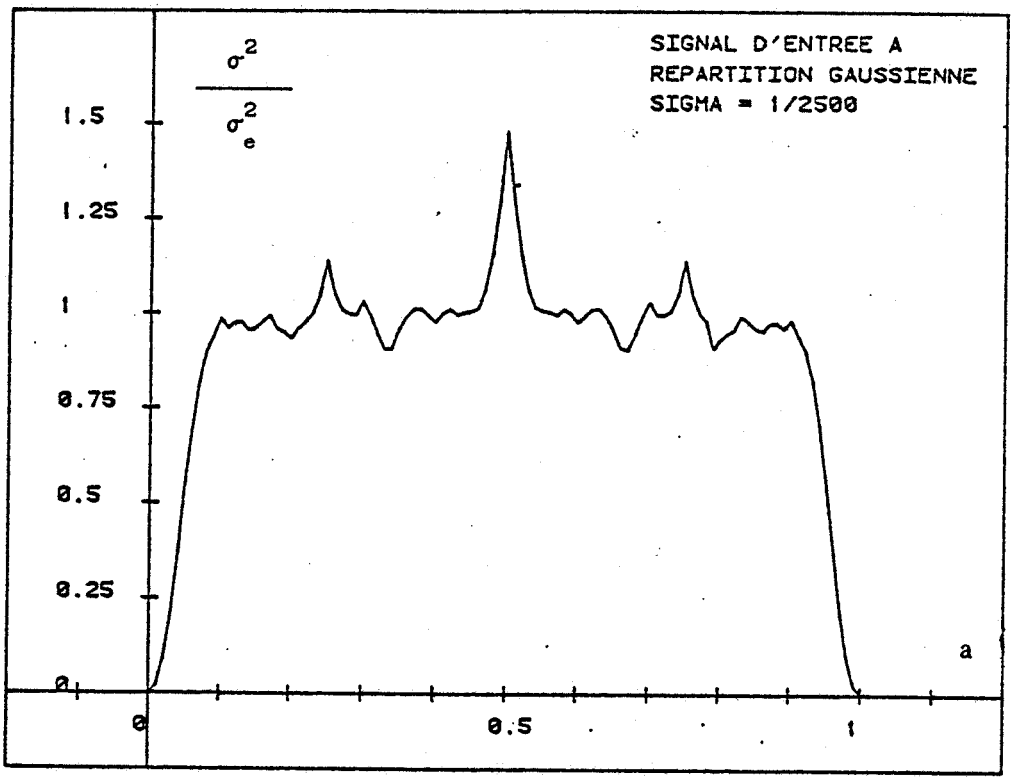


Figure II.4.c - *Puissance réduite de bruit en fonction de a pour* $\sigma_x = 4\ 10^{-4}$.

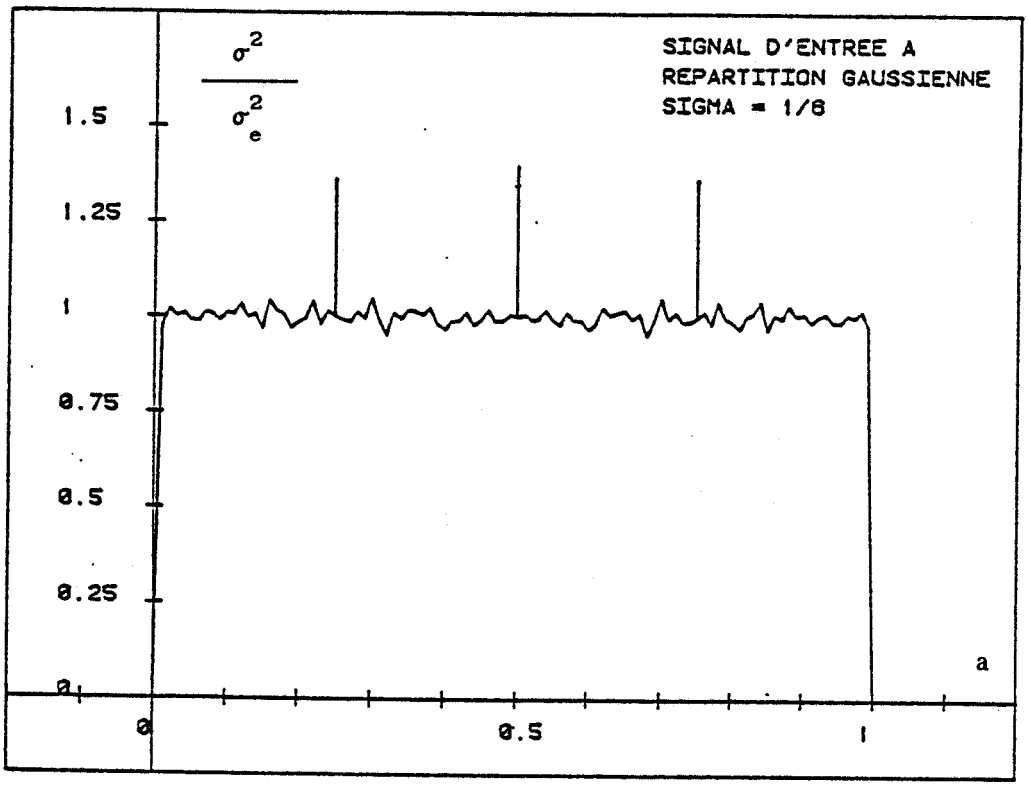


Figure II.4.d - *Puissance réduite de bruit en fonction de a pour* $\sigma_x = 1/6$.

Les simulations montrent que pour des signaux d'entrée de variance très inférieure à 1 (figures 4.a, 4.b, 4.c) donc de faibles niveaux d'amplitudes, il existe de très fortes variations au voisinage des valeurs entières (0 et 1) ainsi que pour certaines valeurs (0.5, 0.75, 0.25). Les variations étant d'autant plus importantes que la variance est faible. Par contre lorsque σ_x est proche de 1, la puissance réduite de bruit tend vers 1 c'est à dire la valeur théorique et ceci quelque soit la valeur du coefficient a (excepté certaines valeurs dont 0.5, 0.75, 0.25).

Il résulte de ceci que pour des signaux de faibles niveaux d'amplitude, c'est à dire proche du pas de quantification q, le modèle du bruit blanc uniformément réparti n'est plus valable [27]. L'erreur d'arrondi est alors fortement dépendante de la valeur du coefficient du multiplieur. On montre de plus [28][29] que dans ce cas la corrélation entre deux échantillons de bruit n'est plus négligeable. Pour des signaux ayant des niveaux d'amplitude assez importants vis à vis du pas de quantification, la simulation vérifie que le modèle du bruit blanc uniformément réparti est correcte [30] quelque soit le coefficient à l'exception toutefois de quelques valeurs. Il convient donc d'étudier plus en détail le cas de ces valeurs qui mettent en fait en évidence le problème supplémentaire de la discrétisation du coefficient du multiplieur.

II.3. CAS PARTICULIER DU COEFFICIENT A VALEURS DISCRETES.

II.3.1. Ensemble fini des valeurs de bruit.

Dans la pratique le coefficient du multiplieur est quantifié sur un nombre B donné de bits. Il ne peut donc prendre ces valeurs que dans un ensemble fini et non dans un continuum. De ce fait le bruit résultant de la multiplication du signal x_Q par le coefficient a ne peut également prendre qu'un nombre fini de valeurs. De plus le nombre de ces valeurs dépend du coefficient a. En effet si l'on pose :

$$a = \sum_{i=1}^{B} a_i . 2^{-i} \qquad a_i \in \{ 0 ; 1 \}$$

$$x_Q = \sum_{j=1}^{B} x_j . 2^{-j} \qquad x_j \in \{ 0 ; 1 \} \qquad (2.4)$$

alors $y = a \cdot x_Q$ peut encore s'exprimer sous la forme d'une somme de puissances de 2^{-1} :

$$y = \sum_{\substack{i \text{ et } j \\ \text{de } 1}}^{B} a_i . x_j . 2^{-(i+j)} \qquad (2.5)$$

Le bruit généré en sortie d'un multiplieur travaillant sur B bits est donc donné par la relation suivante :

$$e = \sum_{i=B+1}^{2.B} e_i . 2^{-i} \qquad (2.6)$$

Il ne peut donc prendre ses valeurs que parmi toutes les combinaisons possibles des puissances de 2^{-1} comprises entre 2^{-1} et 2^{-B} incluses soit dans le cas général au maximum 2^B valeurs possibles auxquelles il faut rajouter toutes les combinaisons négatives, soit un ensemble total T de valeurs possibles donné par :

$$T = (2^B - 1) . 2 + 1 = 2^{(B+1)} - 1 \qquad (2.7)$$

Il convient toutefois de faire remarquer que le calcul précédent suppose implicitement que l'on utilise une technique de quantification par troncature. En effet dans le cas de la troncature, la valeur absolue maximale de l'erreur est bien $q = 2^{-B}$; T est donc effectivement égale à $2^{(B+1)} - 1$. Dans la technique de la quantification par arrondi la valeur absolue maximale de l'erreur est cette fois de q/2 et il ne peut apparaître qu'une seule combinaison possédant un terme en 2^{-1} (ou -2^{-1}) d'où une valeur de T qui devient alors :

$$T = (2^{(B-1)} - 1) . 2 + 2 + 1 = 2^B + 1 \qquad (2.8)$$

Il est à noter de plus que toutes ces valeurs ne sont pas forcément atteintes mais qu'elles sont possibles.

Ces calculs restent cependant généraux parce qu'en réalité pour un multiplieur de B bits de coefficient a, il est possible que l'on ait :

$$\exists\ i_0 \quad \text{tel que} \quad \forall\ i > i_0 \quad a_i = 0 \tag{2.9}$$

Dans ce cas, on voit tout de suite que ce n'est plus parmi une base de B puissances de 2^{-1} que le bruit peut prendre ses valeurs mais parmi une base de i_0 puissances de 2^{-1}. Le nombre maximum de valeurs possibles pour le signal erreur devenant alors égale à :

$2^{(i_0+1)} - 1$ pour la troncature et

$2^{i_0} + 1$ pour l'arrondi.

Il devient alors évident que pour des valeurs de a telles que 0.5, 0.25, ou 0.75, l'hypothèse du bruit uniformément réparti dans l'intervalle [-q/2 , q/2] est difficilement admissible puisque par exemple pour a = 0.5 le bruit d'arrondi ne peut prendre que trois valeurs distinctes au maximum dans cet intervalle.

II.3.2. Simulations.

Nous avons donc simulé différents exemples de la fonction densité de probabilité du bruit d'arrondi e en sortie d'un multiplieur travaillant sur B = 16 bits afin de mettre en évidence les remarques faites précédemment. Pour cela nous avons représenté la densité de probabilité de la fonction $E = e \cdot 2^B$ pour différentes valeurs quantifiées du coefficient a du multiplieur. Dans un premier temps la variable d'entrée x_Q est de type gaussienne d'écart type 1/6 (figures II.5.a à II.5.d) puis d'écart type 10^{-4} (figures II.6.a à II.6.d).

Les fonctions densité obtenues sont bien discrètes et on constate que l'hypothèse de répartition uniforme de l'erreur n'est pas acceptable pour toutes les valeurs du coefficient a. Cette hypothèse étant d'autant moins vérifiée que l'indice i_0 (confère précédemment) est élevé.

Par ailleurs dans le cas où la variance du signal d'entrée est petite devant 1, l'hypothèse de répartition uniforme est également non vérifiée est cette fois ci pour pratiquement toutes les valeurs du coefficient quantifié a, comme cela avait été dit au paragraphe II.2.

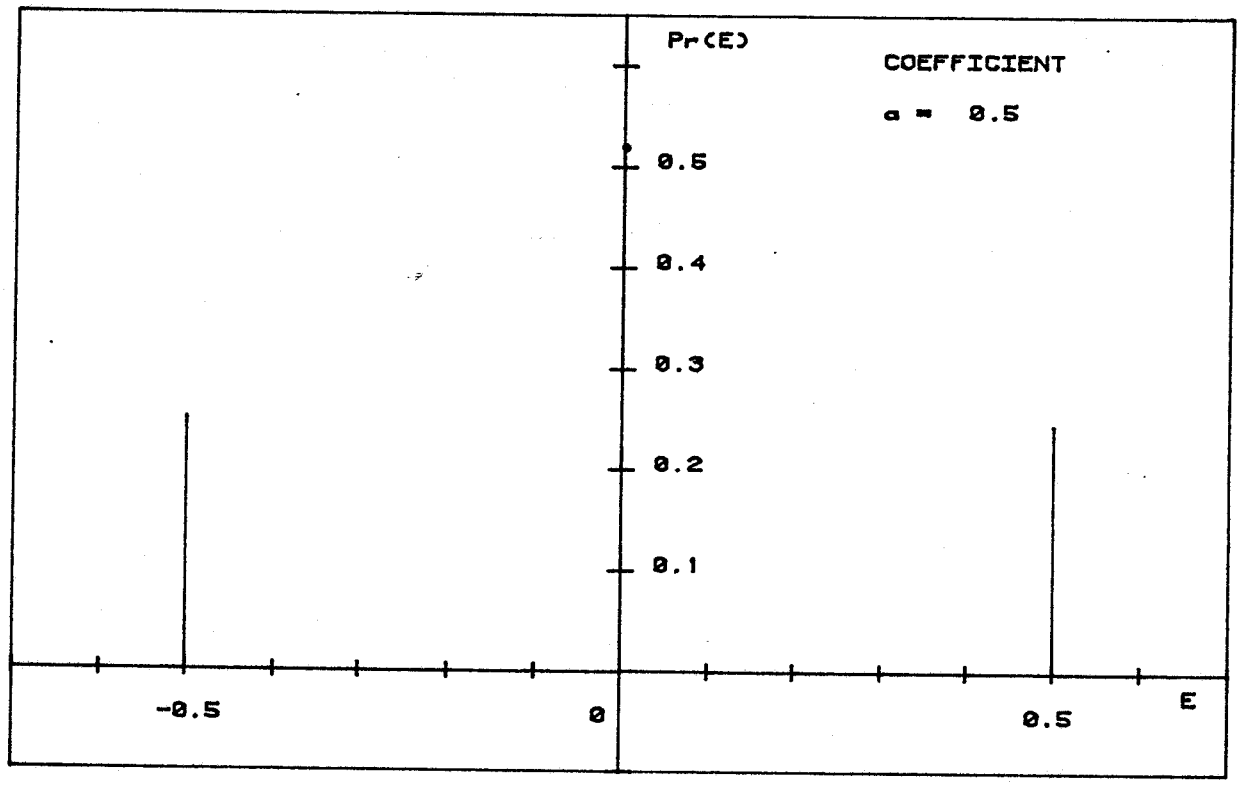


Figure II.5.a - *Densité de probabilité de* $E = e.2^B$ *pour* $a = 0.5$ *et* $\sigma_x = 1/6$.

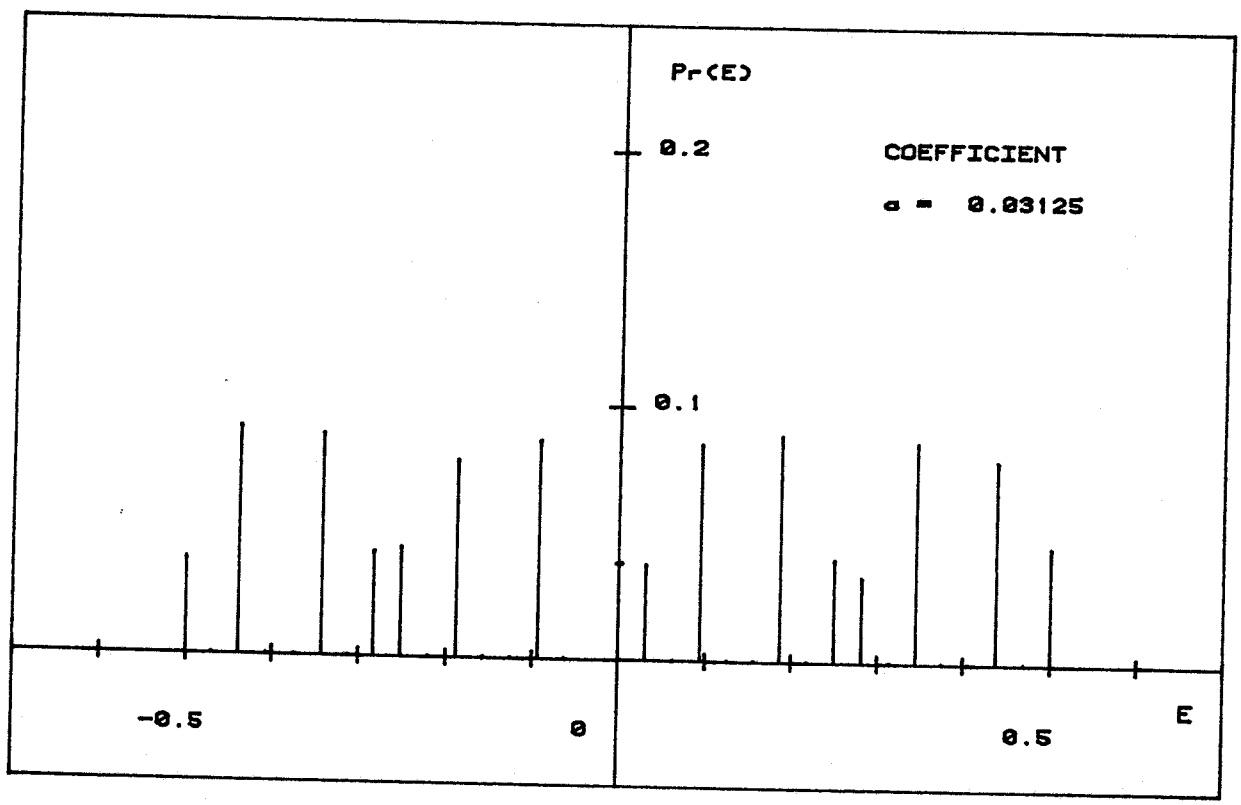


Figure II.5.b - *Densité de probabilité de* $E = e.2^B$ *pour* $a = 0.03125$ *et* $\sigma_x = 1/6$.

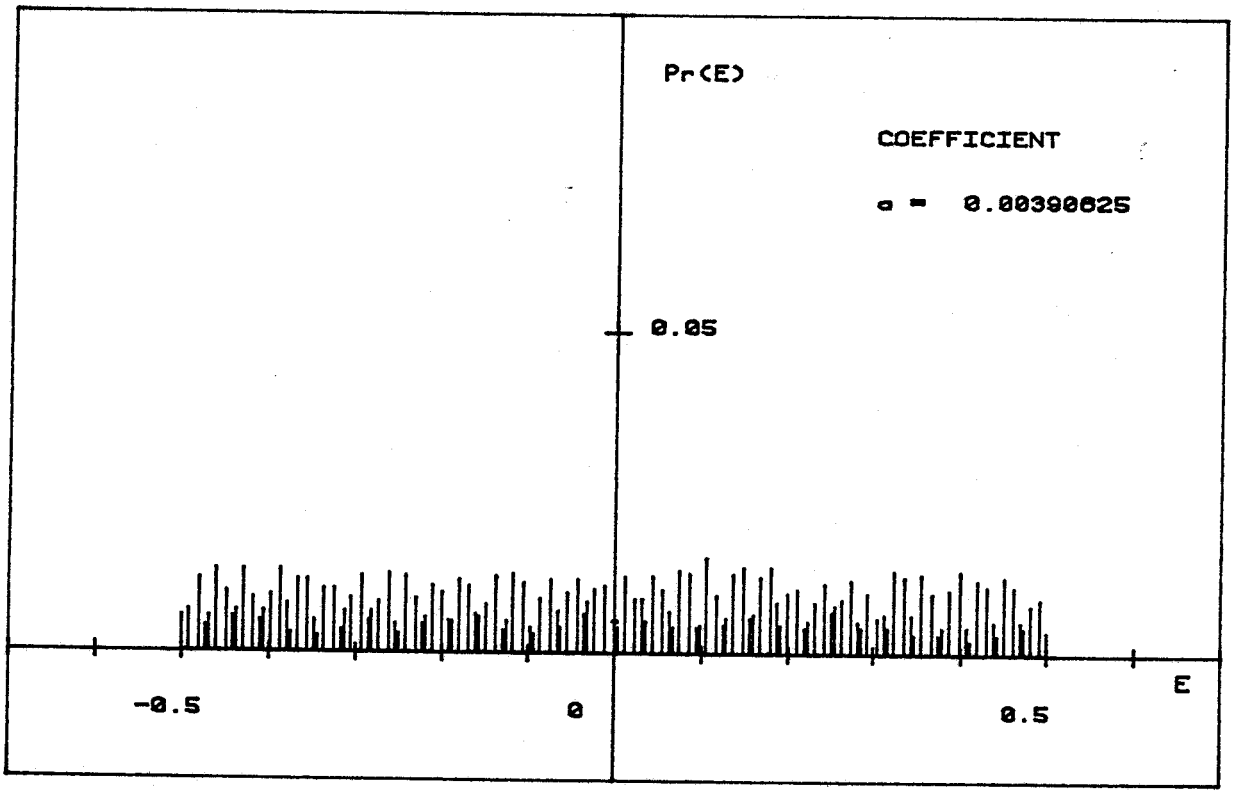


Figure II.5.c - *Densité de probabilité de $E = e.2^B$ pour $a = 0.00390625$ et $\sigma_x = 1/6$.*

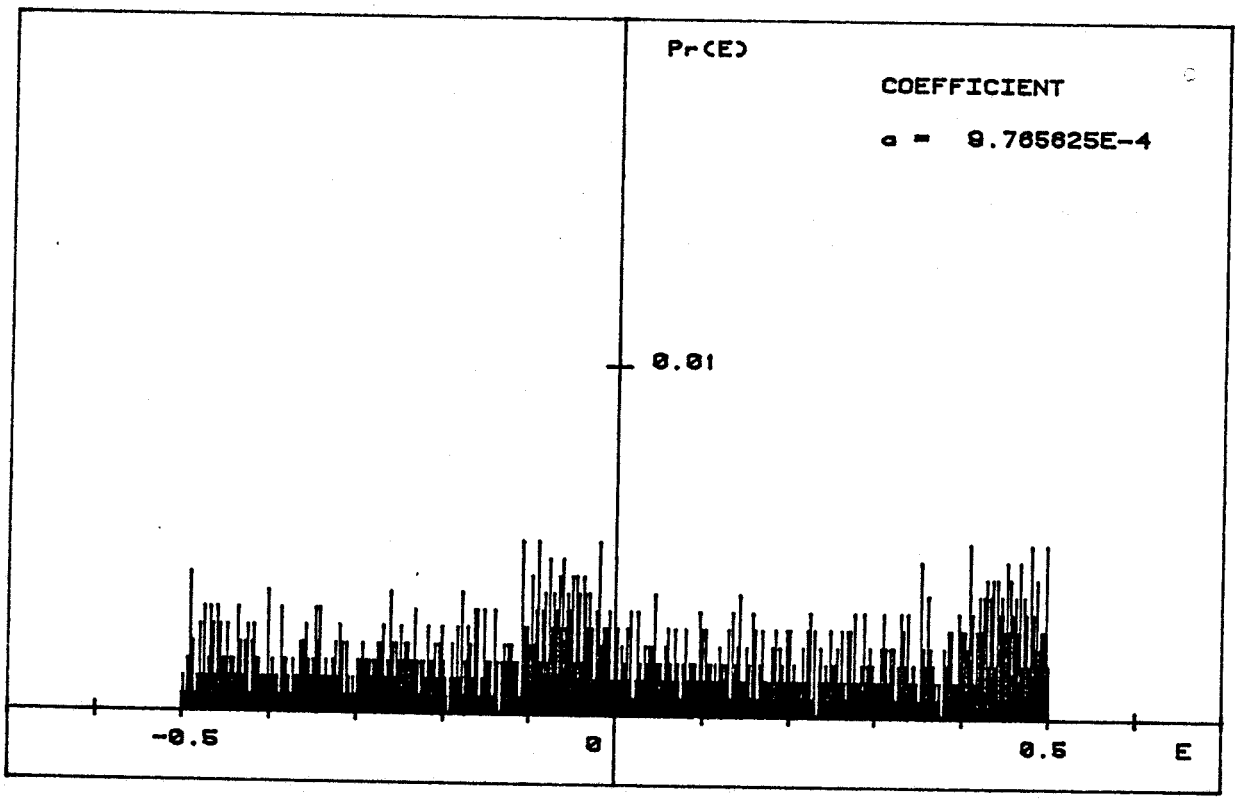


Figure II.5.d - *Densité de probabilité de $E = e.2^B$ pour $a = 9.765625\ 10^{-4}$ et $\sigma_x = 1/6$.*

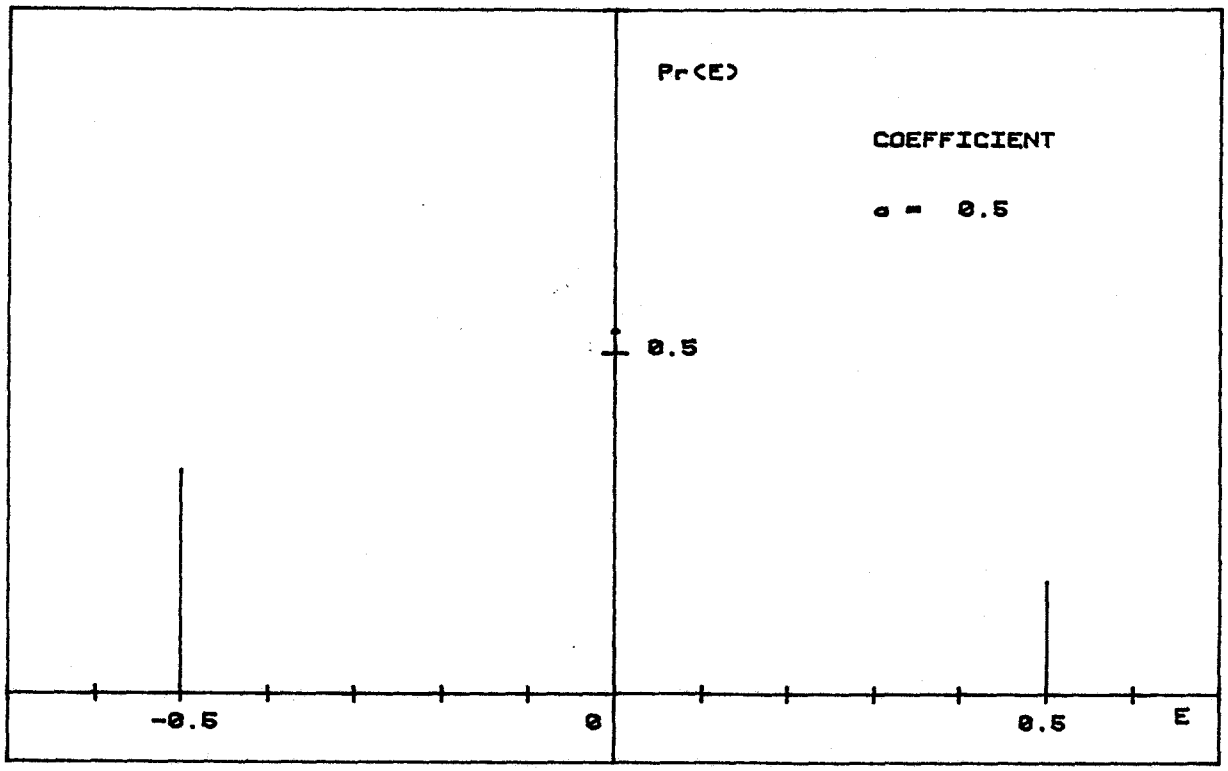


Figure II.6.a - *Densité de probabilité de* $E = e.2^B$ *pour* $a = 0.5$ *et* $\sigma_x = 2.5\ 10^{-3}$.

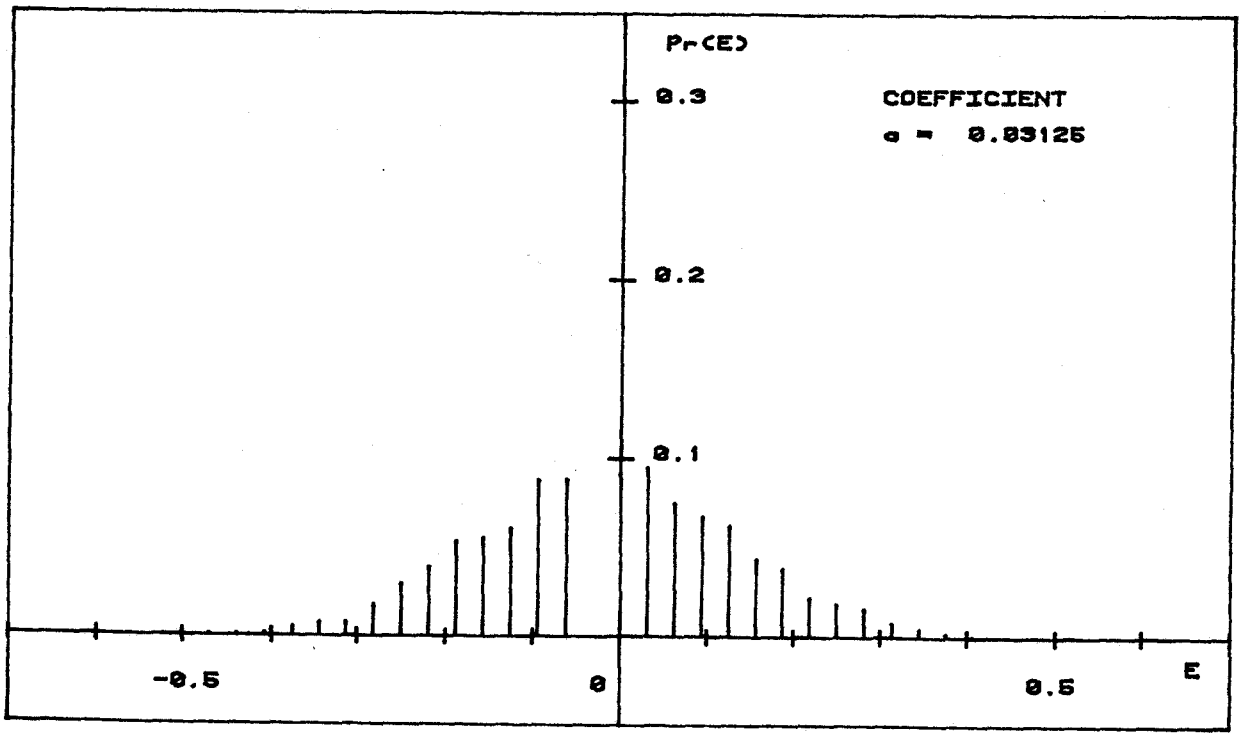


Figure II.6.b - *Densité de probabilité de* $E = e.2^B$ *pour* $a = 0.03125$ *et* $\sigma_x = 2.5\ 10^{-3}$.

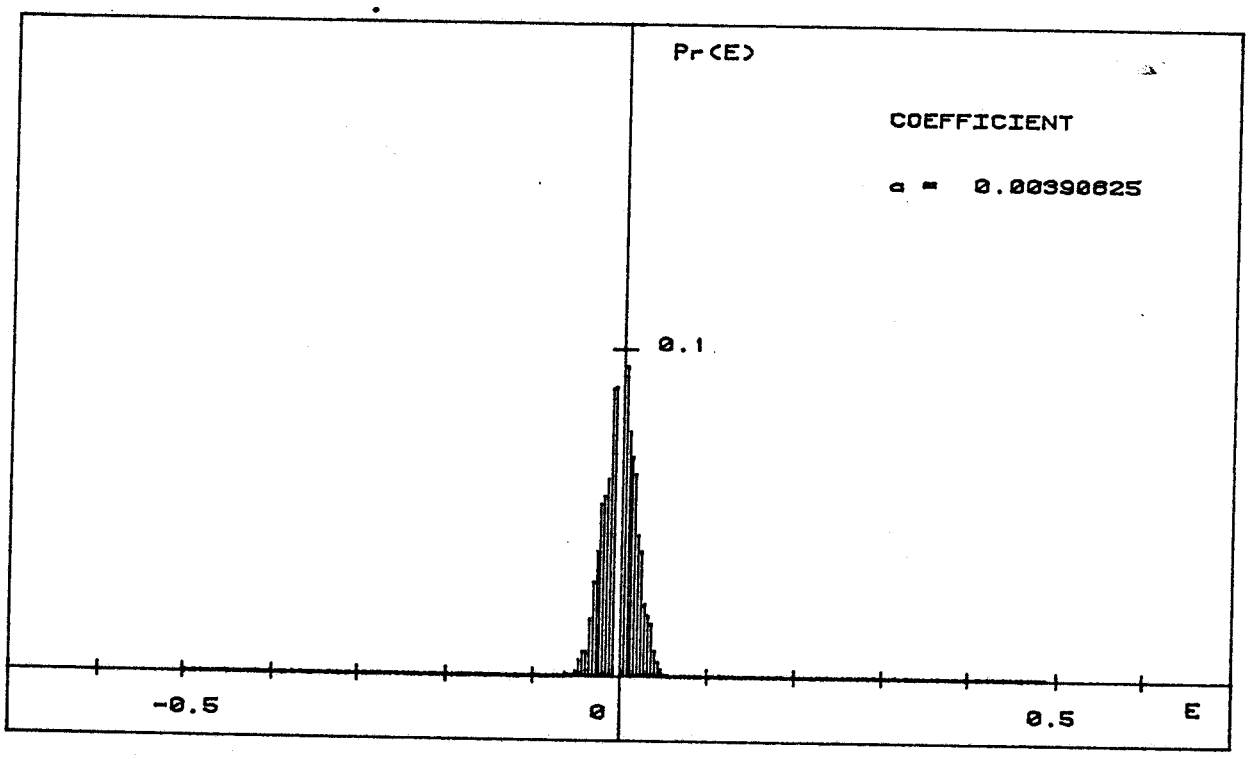


Figure II.6.c - *Densité de probabilité de $E = e.2^B$ pour $a = 0.00390625$ et $\sigma_x = 2.5\ 10^{-3}$.*

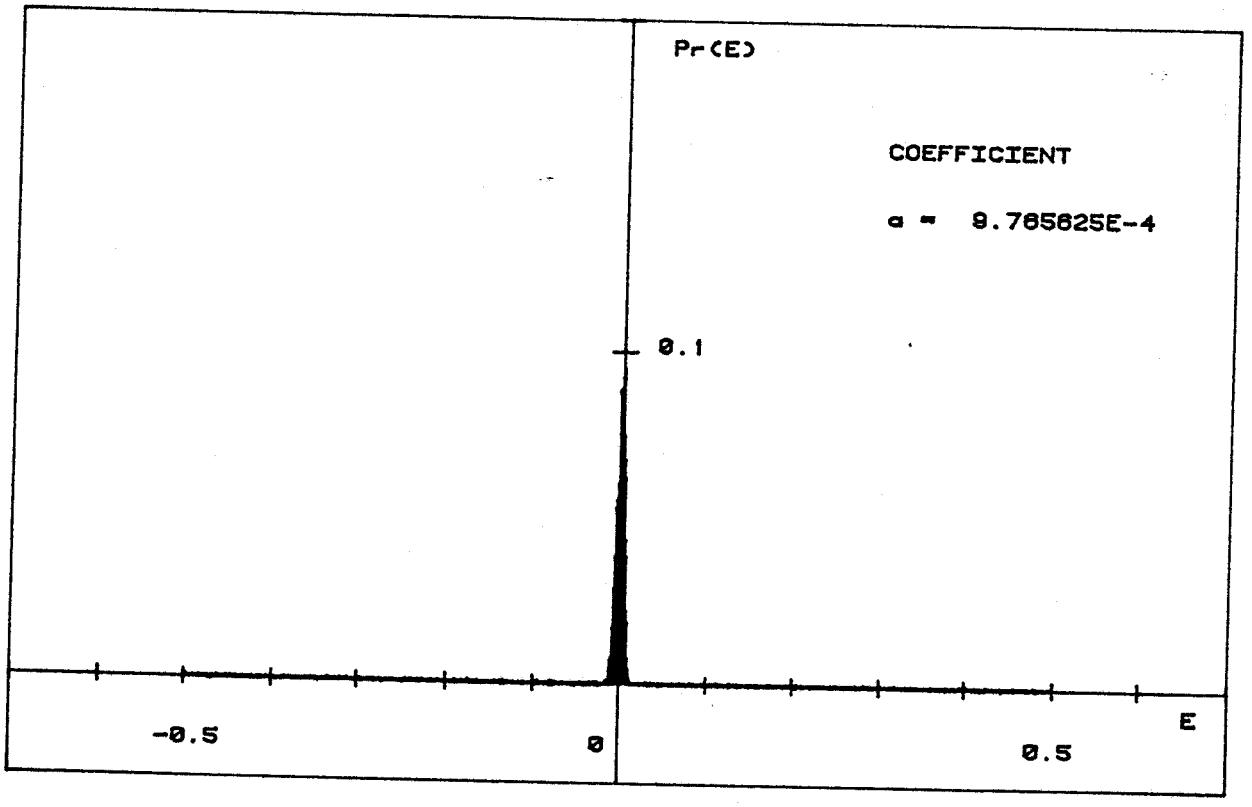


Figure II.6.d - *Densité de probabilité de $E = e.2^B$ pour $a = 9.765625\ 10^{-4}$ et $\sigma_x = 2.5\ 10^{-3}$.*

Maintenant que nous avons caractérisé les sources de bruits dus aux multiplications il est possible de s'intéresser au bruit global que peuvent générer ces sources en sortie d'une structure qui les contient.

II.4. RAPPEL DE LA METHODE CLASSIQUE D'EVALUATION DU BRUIT DE CALCUL DANS UN FILTRE NUMERIQUE RECURSIF.

La valeur à calculer est celle de la variance du bruit global en sortie du filtre. Ce bruit résulte des différents bruits dus aux opérations arithmétiques intervenant au cours du calcul du filtre.

Parmi les opérateurs arithmétiques que l'on considère, on suppose que la réalisation du filtre prévoit un facteur d'échelle à l'entrée pour réduire au minimum un éventuel dépassement de capacité en sortie des additionneurs. L'effet de quantification des coefficients n'est pas pris en considération pour des filtres à coefficients fixes. Par la suite, nous ne considérerons donc que les sources de bruit engendrées en sortie des multiplieurs.

Compte tenu des hypothèses communément admises (et avec les réserves qui ont été apportées aux paragraphes précédents) :

H.1. : Deux échantillons différents d'une même source de bruit sont supposés non corrélés.

H.2. : Deux sources différentes de bruit (c'est à dire deux sources associées à deux multiplieurs différents) sont supposées non corrélés.

H.3. : Chaque source de bruit est supposée non corrélée avec la séquence d'entrée.

et des propriétés des systèmes linéaires, la variance du bruit en sortie d'un filtre possédant M sources de bruit est donnée par la formule [11] [12] :

$$\sigma^2 = \sum_{k=1}^{M} \sigma_{e_k}^2 \cdot \oint_{\substack{\text{Cercle} \\ \text{unité}}} \frac{H_k(z).H_k(z^{-1})}{2.\pi.j} z^{-1}\, dz \qquad (2.10)$$

où $H_k(z)$ représente la transformée en z de la réponse impulsionnelle du filtre qui aurait pour entrée la $k^{ième}$ source de bruit e_k de variance σ_{ek}^2. Comme on l'a vu, ces bruits sont classiquement modélisés par des variables aléatoires uniformément réparties ce qui conduit pour une technique de quantification par arrondi, à des variances de la forme $q^2/12$, où $q = 2^{-b}$ si l'on traite des mots binaires de b+1 bits.

Des formules analytiques ont été développées [31], qui permettent le calcul des normes $\| H_k \|_2$ pour toute transmittance rationnelle jusqu'à l'ordre quatre. Le calcul peut aussi être fait au moyen d'une méthode qui fait appel à la représentation des filtres dans l'espace d'état [32] [33]. Cette méthode présente de plus l'avantage de pouvoir être appliquée quel que soit l'ordre du filtre.

II.5. CALCUL DE LA VARIANCE DU BRUIT EN SORTIE D'UN FILTRE AU MOYEN DE SA REPRESENTATION DANS L'ESPACE D'ETAT.

II.5.1. Expression de la variance du bruit en sortie d'un filtre.

Soit le filtre de fonction de transfert en z :

$$H(z) = \sum_{n=0}^{+\infty} h(nT).z^{-n} = \frac{N(z)}{D(z)} \qquad (2.11)$$

et ses équations d'état :

$$\begin{aligned} x(n+1) &= A.x(n) + B.u(n) \\ y(n) &= C.x(n) + D.u(n) \end{aligned} \qquad (2.12)$$

où x(n) est le vecteur d'état du filtre à l'instant nT
u(n) est l'échantillon d'entrée à l'instant nT,
y(n) est l'échantillon de sortie à l'instant nT.

on a alors :

$$h(n) = \begin{cases} D & \text{pour } n = 0 \\ C.A^{n-1}.B & \text{pour } n > 0 \end{cases} \qquad (2.13)$$

et

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

Pour calculer le bruit en sortie du filtre, dû aux multiplications, il faut considérer chacune des réponses $h_k(n)$ de la sortie du filtre qui a pour entrée la variable d'état correspondante, $x_k(n)$.

Si $u(n) = 0$ pour tout n et si $x_k(n) = \delta_k(n)$ avec δ_k la fonction de Dirac, on peut écrire:

$$y(n) = C.A^n.x(0) = (C.A^n)_k \qquad (2.14)$$

$h_k(n)$ est donc la $k^{ième}$ composante du vecteur ligne de $C.A^n$.

La variance de l'erreur en sortie du filtre, due aux m_k sources de bruit de multiplication ramenées au niveau de la variable d'état x_k est alors donnée par la relation :

$$\sigma_k^2 = m_k \cdot \frac{q^2}{12} \cdot \| h_k \|^2 \qquad (2.15)$$

On définit alors une matrice de bruit W, symétrique, de dimension N (si le vecteur d'état du filtre est de dimension N) qui vérifie l'équation :

$$W = A^T.W.A + C^T.C = \sum_{k=0}^{+\infty} (C.A^k)^T.(C.A^k) \qquad (2.16)$$

dont les éléments W_{ij} sont donnés par les produits $h_i.h_j$.

La variance du bruit en sortie du filtre peut donc encore s'écrire sous la forme :

$$\sigma^2 = \frac{q^2}{12} \cdot \sum_{k=1}^{N} m_k . W_{kk} \qquad (2.17)$$

II.5.2. Algorithme de calcul de W.

Le calcul des éléments de W est un problème bien connu [34][35][36]. Un algorithme pratique donné par [32] permet d'obtenir aisément ces éléments. Soit D(z) le polynôme d'ordre N représentant le dénominateur de la fonction de transfert en z du filtre :

$$D(z) = z^N + \sum_{i=1}^{N-1} b_i . z^{N-i} \qquad (2.18)$$

on définit $(r_0, r_1, r_2, \ldots, r_N)$ N+1 inconnues vérifiant la relation :

$$r_i + \sum_{j=1}^{N} b_j . r_{|i-j|} = \begin{cases} 1 & \text{pour } i = 0 \\ 0 & \text{pour } i \geq 1 \end{cases} \qquad (2.19)$$

ainsi qu'une matrice R, symétrique, de Toeplitz de dimension N dont le premier rang est donné par :

$$[\, r_0, r_1, r_2, \ldots, r_{N-1} \,]$$

On introduit d'autre part une matrice X dont les vecteurs colonnes sont définis par :

$$\begin{cases} X(1) = C^T \\ X(k+1) = A^T . X(k) + b_k . C^T \qquad \forall\, k \in [1,N] \end{cases} \qquad (2.20)$$

W est alors donnée par la relation :

$$W = X.R.X^T$$

II.6. VALEURS EXPERIMENTALES DE LA PUISSANCE DE BRUIT D'ARRONDI EN SORTIE D'UN FILTRE.

L'étude théorique précédemment décrite, demande une confirmation expérimentale. Nous avons mesuré, au moyen de simulations, la puissance de bruit d'arrondi en sortie de différents types de filtres dans le but de vérifier l'exactitude des résultats théoriques et de permettre d'en apprécier la portée pratique.

II.6.1. Description de l'instrumentation de calcul des valeurs expérimentales de la puissance de bruit.

Les mesures ont été faites à l'aide d'un logiciel [37] de simulation qui prend en compte les caractéristiques réelles des composants utilisés pour la réalisation des structures de circuits numériques. Ce logiciel à la réalisation duquel nous avons participé, a été conçu et mis au point au laboratoire sur une architecture V.M.E. (microprocesseur Motorola 68000) munie d'un système d'exploitation type VERSADOS. Il comprend d'une part une capture de schémas (logiciel écrit en Assembleur 68000) qui assure une interactivité entre l'utilisateur et la structure de donnée qui décrit le schéma du filtre en mémoire et d'autre part un logiciel simulateur (écrit en PASCAL). Ce dernier fournit à chaque instant les valeurs des sorties de chaque opérateur et à chaque période d'horloge, le vecteur d'état du système que constituent les contenus des mémoires internes.

Dans la phase d'initialisation du logiciel, l'utilisateur doit préciser le format des bus de transfert de données, des registres internes du filtre ainsi que la fréquence d'horloge de l'ensemble. On lui propose alors au choix, deux techniques de quantification : par arrondi ou par troncature. Il lui reste ensuite à créer le schéma du filtre qu'il désire réaliser. Pour cela il a à sa disposition les opérateurs élémentaires suivants:

- Additionneur.

- Multiplieur.

- Retard.

auxquels on ajoute deux symboles graphiques servant à définir l'entrée et la sortie du filtre. Une fois ces opérateurs graphiques placés, il doit les relier de manière à décrire complètement le filtre. Ceci se fait au moyen d'une commande d'édition de fils. Lors de la création d'un opérateur, il est demandé à l'utilisateur de préciser les paramètres suivants :

- Durée de propagation si l'opérateur est un additionneur.

- Durée de propagation et coefficient multiplicatif si c'est un multiplieur.

L'opérateur de retard a par convention une durée de propagation correspondant à une période de l'horloge du filtre.

Le logiciel de capture de schémas a été conçu de manière à avoir la plus grande souplesse possible d'utilisation. Dans ce but,il est possible à tout moment, d'effacer ou de créer des opérateurs ou des fils de liaison. Une nouvelle version de ce logiciel écrite en langage C sur un ordinateur IBM type AT3 permet également la définition d'un filtre ou d'une partie d'un filtre comme "macro" opérateur qu'il est possible de rappeler lors de la création d'un autre filtre. Cette commande supplémentaire est d'une grande utilité dans la mesure ou bon nombre de filtres sont constitués par la répétition en cascade ou en parallèle d'un même motif de base. On n'a donc plus à dessiner l'intégralité du filtre mais seulement à assembler plusieurs fois le même motif de base. Ceci permet de plus de s'affranchir du problème du nombre limité d'opérateurs représentables à l'écran.

A cette représentation graphique du filtre est associée une représentation en mémoire des différents opérateurs et des liaisons qui existent entre ces opérateurs.

Pour ce faire, lors de la création des opérateurs par l'utilisateur, le logiciel génère pour chaque opérateur graphique, une représentation en mémoire sous la forme d'une table qui est constituée de la façon suivante:

Numéro de l'opérateur
Type de l'opérateur
Durée de propagation (si nécessaire)
coefficient (si nécessaire)
Nombre d'opérateurs reliés en sortie
Numéro du premier opérateur relié
Numéro du deuxième opérateur relié
...
Numéro du dernier opérateur relié

Ces tableaux doivent être dynamiques puisque l'on ne connait pas à priori leur dimension.

Une fois le dessin du filtre fini, toutes ces tables mémoire sont réactualisées compte tenu des éventuels effacement de fils ou d'opérateurs qui peuvent avoir lieu en cours de dessin. Une table finale est alors transmise au simulateur qui regroupe toutes les tables des opérateurs en précisant le nombre d'opérateurs présents, le numéro de l'opérateur de sortie et l'unité de temps définie par l'utilisateur..

Le simulateur génère à partir de ces tables, quatre tables de travail.

- la table d'exécution qui est définie de la façon suivante :

Numéro de l'opérateur
Type de l'opérateur
Instant d'apparition du résultat
Nombre d'entrée de l'opérateur
Valeur présente sur l'entrée un
...
Valeur présente sur l'entrée deux
Valeur présente en sortie

- La table des équipotentielles qui contient pour chaque opérateur, les numéros des opérateurs reliés à la sortie de cet opérateur.

- La table des états qui est constituée par les sorties des opérateurs à chaque instant de calcul. Cette table est en faite doublée par une table donnant les sorties des opérateurs à l'instant de traitement précédent.

Le problème principal de la simulation vient du fait de la prise en compte des durées de propagation des différents opérateurs. Ceci se traduit principalement dans le cas de l'additionneur par le fait que plusieurs données peuvent se présenter à ses entrées de manière asynchrone. Cette difficulté a été levée en ajoutant au système, une table supplémentaire, baptisée table temporelle. Elle est constituée par les différents instants auxquels apparaissent les sorties des opérateurs. Ces instants sont calculés de manière relative par rapport au début d'une période d'horloge du filtre, ceci pour des raisons de commodité de traitements. Ils sont de plus, rangés par ordre croissant dans la table temporelle.

Le fonctionnement du simulateur est alors le suivant. Après avoir chargé les entrées des opérateurs à l'instant relatif t=0, on vient lire la première valeur dans la table temporelle. On effectue une comparaison entre cet instant τ_0 et tous les instants d'apparition des résultats des opérateurs dans la table d'exécution. Pour chaque opérateur ayant un instant d'apparition du résultat, égal à τ_0, on effectue l'opération correspondante au type de l'opérateur. On réactualise ensuite les données d'entrée des opérateurs dans la table d'exécution, au moyen de la table des équipotentielles. Pour chaque opérateur dont les entrées ont été modifiées à la suite de cette première étape, on réactualise l'instant d'apparition du résultat. Ceci est fait en ajoutant la valeur τ_0 à l'instant précédent d'apparition du résultat. On relit ensuite dans la table d'exécution tous les instants d'apparition du résultat de chaque opérateur et l'on recrée une nouvelle table temporelle. L'étape suivante du calcul consiste à comparer les deux tables d'état des sorties (instant présent, instant précédent). Si elles sont égales, on recommence le cycle d'opérations avec la même valeur τ_0 et ceci jusqu'à obtention de deux tables d'état différentes. Dans ce cas, le calcul se poursuit en prenant la valeur suivante τ_1 de la nouvelle table temporelle et en recommençant le cycle d'opérations à son début. Ceci jusqu'à épuisement de la table temporelle qui est réactualisée de façon dynamique ou jusqu'à ce que le compteur de temps relatif ait atteint la valeur d'une période de l'horloge du filtre. La comparaison de ces deux tables d'état conduit de plus à une détermination directe de la stabilité des données. Cette technique permet donc la simulation d'un système asynchrone qui ne peut prendre des valeurs qu'à des instants discrets toujours contenus dans une période d'horloge si le filtre est stable.

Grâce à ce simulateur, il a été possible d'obtenir aisément les valeurs expérimentales de la puissance de bruit en sortie de différentes structures de filtres du deuxième ordre dont les schémas sont donnés en figure II.7 :

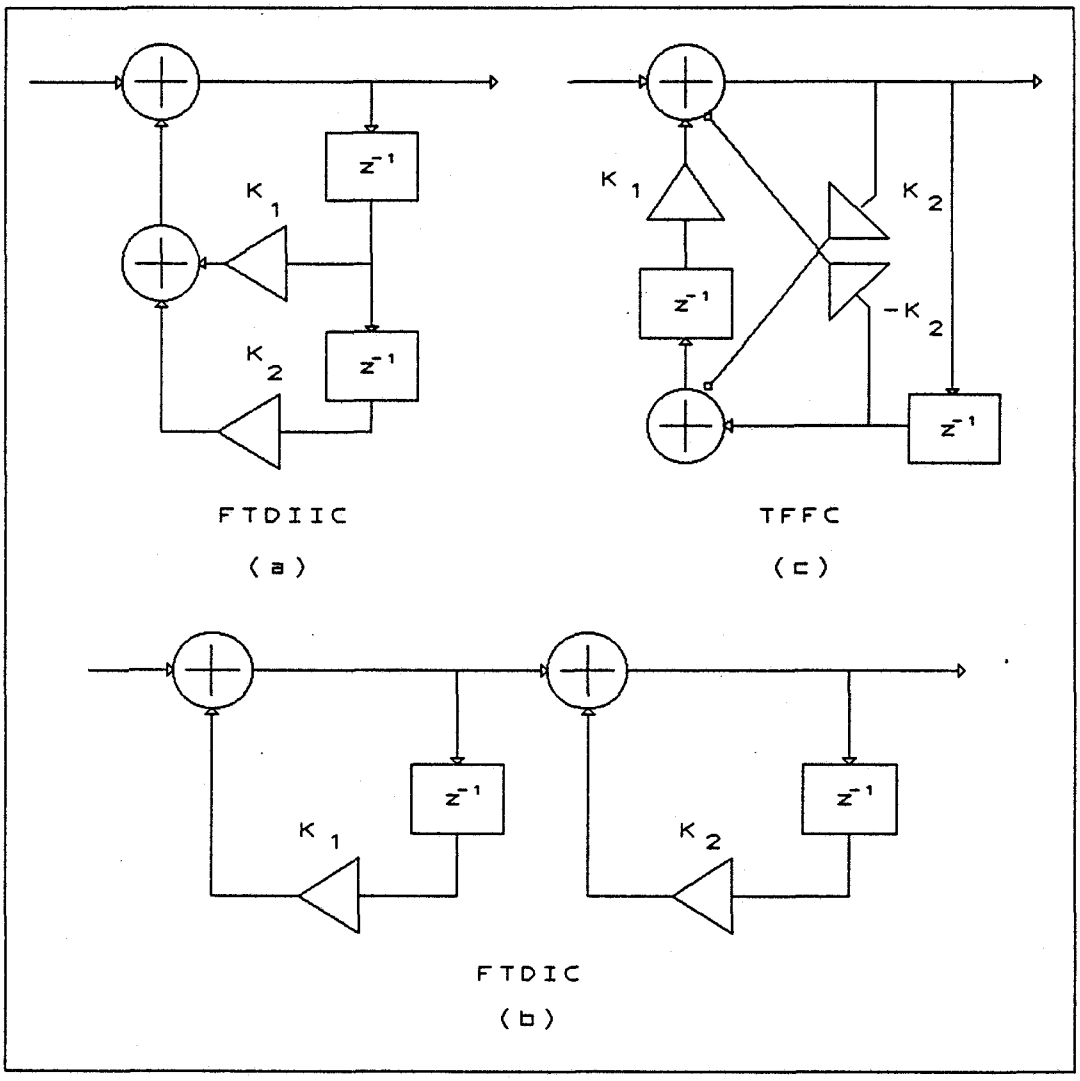


Figure II.7 - *Différentes structures de filtres du deuxième ordre expérimentées :*
(a) Directe d'ordre deux
(b) 1^er ordre en cascade
(c) Treillis en cascade.

Pour chacune de ces structures, nous avons comparé les valeurs des échantillons de sortie du simulateur avec ceux de la sortie d'un filtre (filtre théorique) analogue mais travaillant avec la précision de l'ordinateur (virgule flottante avec 52 bits de mantisse). Les échantillons d'entrée sont fournis par un générateur de bruit gaussien de valeur moyenne nulle et de variance unité. La génération d'un bruit de ce type est décrite en annexe 3 . Sachant que pour une puissance donnée c'est un signal de ce type

qui possède l'entropie maximale (confère annexe 4), il permet donc de caractériser au mieux l'effet sur la sortie, des différentes sources de bruit considérées.

La simulation ne devant rendre compte que des bruits dus aux multiplieurs, les résultats du simulateur ont été comparés à ceux du filtre théorique pour lequel les échantillons d'entrée et les coefficients ont été préalablement quantifiés de façon à n'introduire aucune autre source de bruit supplémentaire.

II.6.2. Résultats de la vérification expérimentale.

La puissance réduite de bruit en sortie du filtre est donnée par l'expression suivante :

$$\frac{\sigma^2}{(q^2/\ 12)} = \sum_{k=1}^{N} m_k .\ W_{kk} \qquad (2.22)$$

où m_k est le nombre de sources de bruit ramenées à la variable d'état x_k, W_{kk} est le kième terme de la diagonale de la matrice W qui n'est fonction que des seules valeurs des coefficients du filtre. Il en résulte que sous réserve des hypothèses **H1**, **H2** et **H3** , le rapport

$$\frac{12\ \sigma^2}{q^2} \qquad (2.23)$$

doit être indépendant du nombre de bits de quantification. La figure II.8 représente la puissance réduite de bruit en sortie d'un filtre du premier ordre en cascade (obtenue à l'aide du simulateur), en fonction du nombre de bit de quantification et ceci pour différents couples de coefficients K_1 et K_2. Toutes les simulations sont faites pour un signal d'entrée comprenant 2000 échantillons.

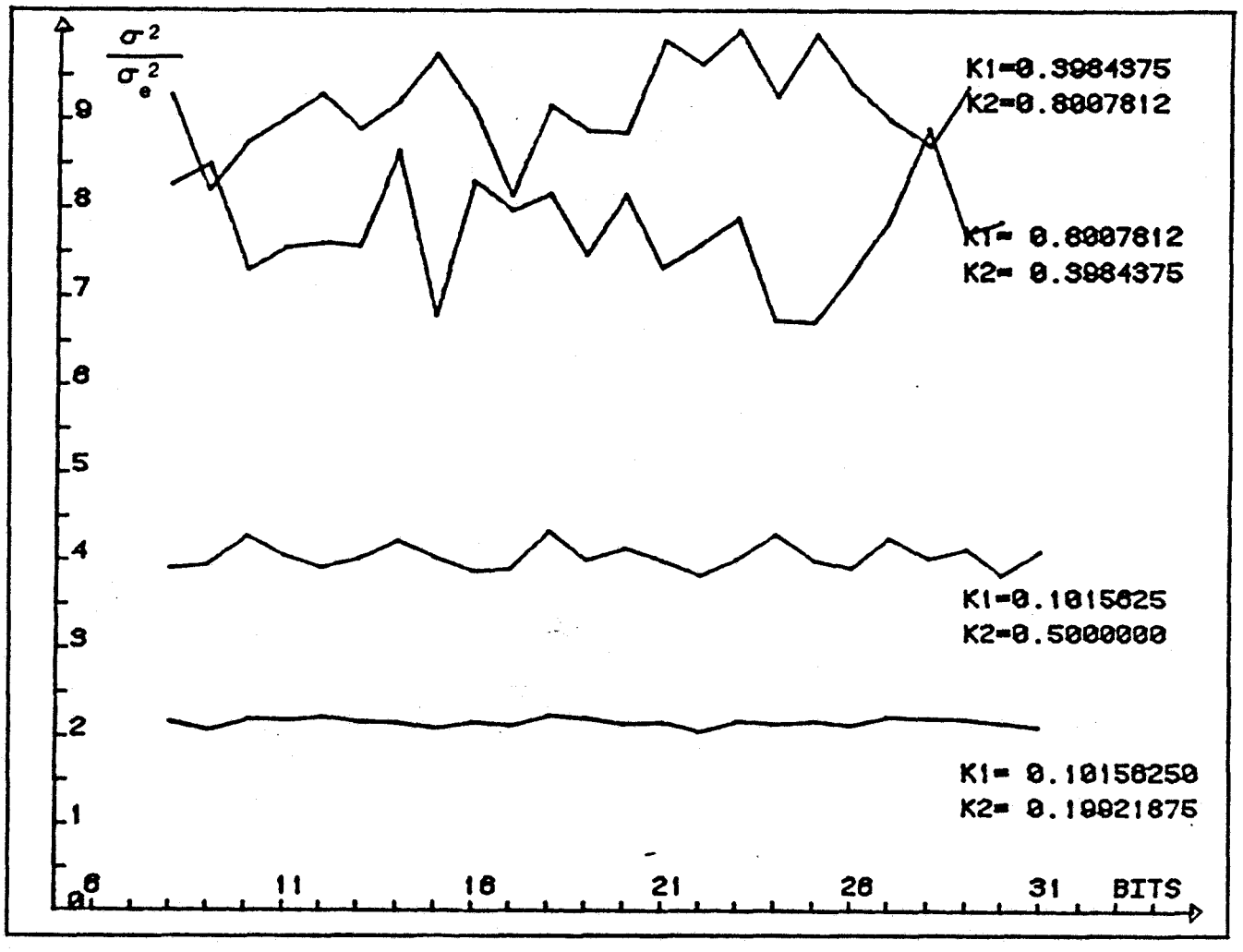


Figure II.8 - *Puissance réduite du bruit en sortie d'un filtre du premier ordre en cascade en fonction du nombre de bits de quantification.*

On constate sur cette figure que la fluctuation relative de la puissance réduite du bruit en sortie du filtre reste indépendante du nombre de bits de quantification.

Une seconde utilisation du simulateur a consisté à comparer, pour un nombre de bits donné (16 dans les exemples traités), les valeurs des puissances réduites de bruits obtenues à l'aide des expressions précédentes avec celles obtenues expérimentalement grâce au simulateur. Cette comparaison effectuée pour les trois types de filtres représentés par les schémas (a), (b) et (c) de la figure II.7 donne les résultats reproduits en figures II.9, II.10 et II.11.

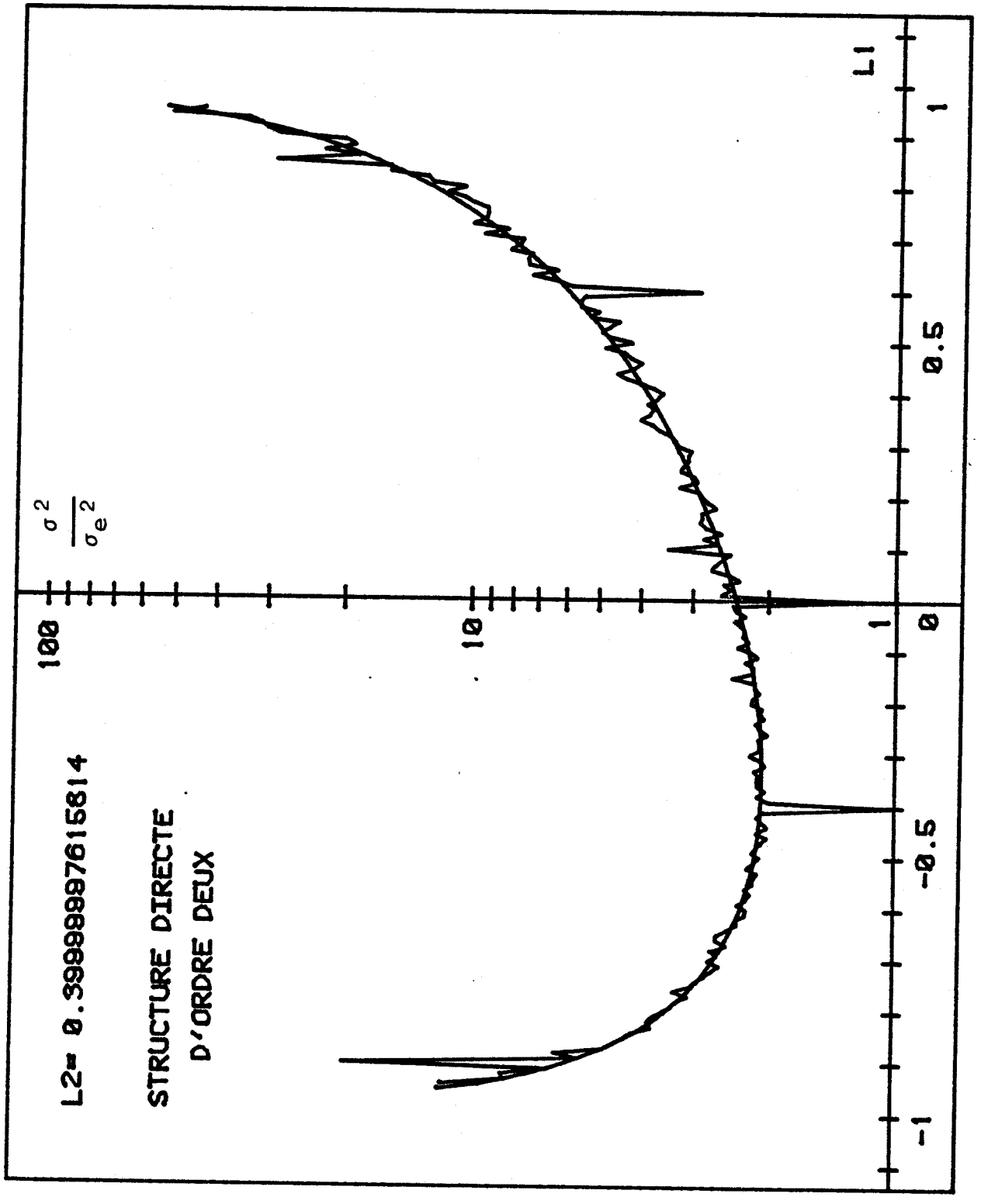


Figure II.9 - *Comparaison entre les résultats théoriques et expérimentaux de la puissance de bruit en fonction des pôles pour le filtre FTDIIC.*

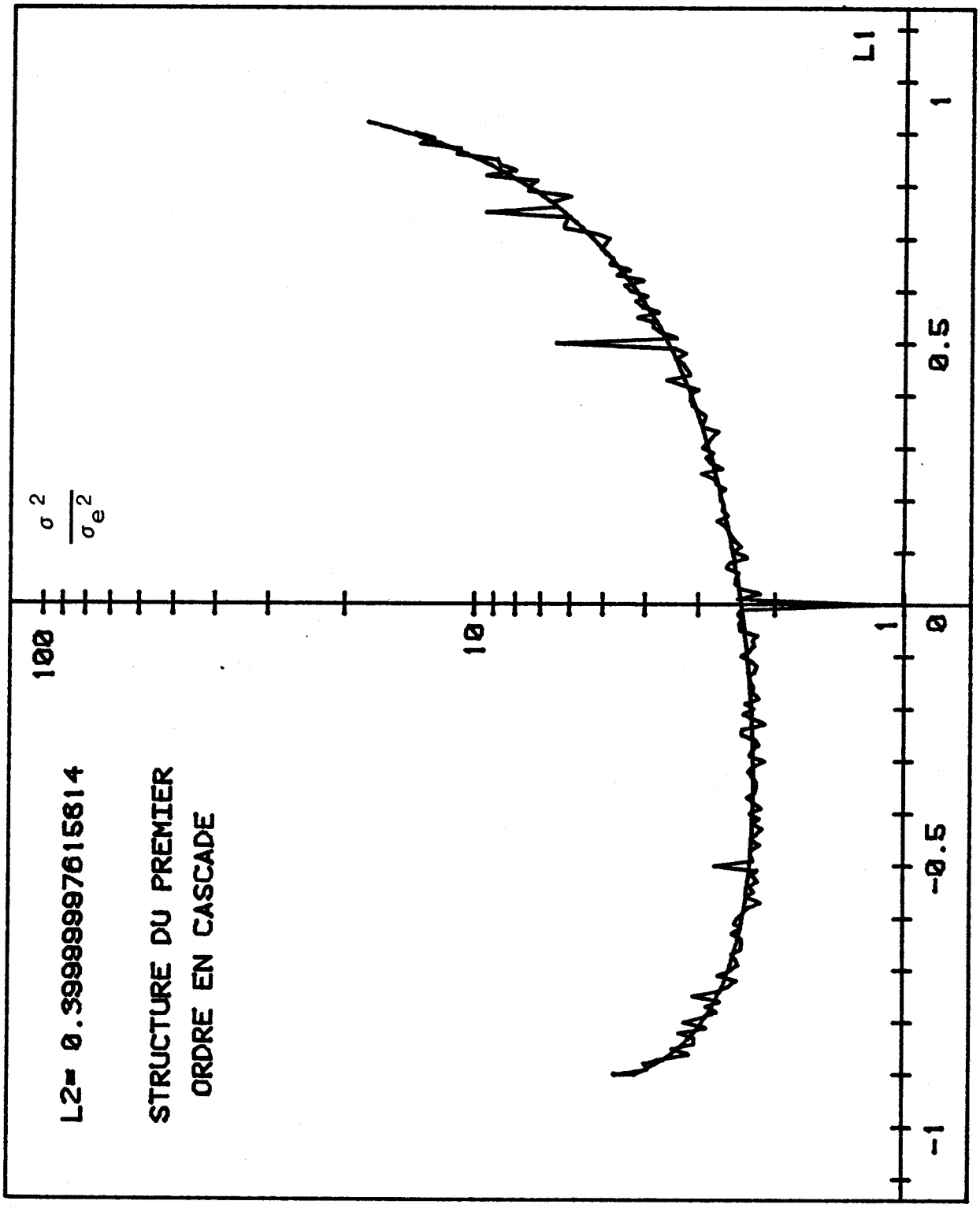


Figure II.10 - *Comparaison entre les résultats théoriques et expérimentaux de la puissance de bruit en fonction des pôles pour le filtre FTDIC.*

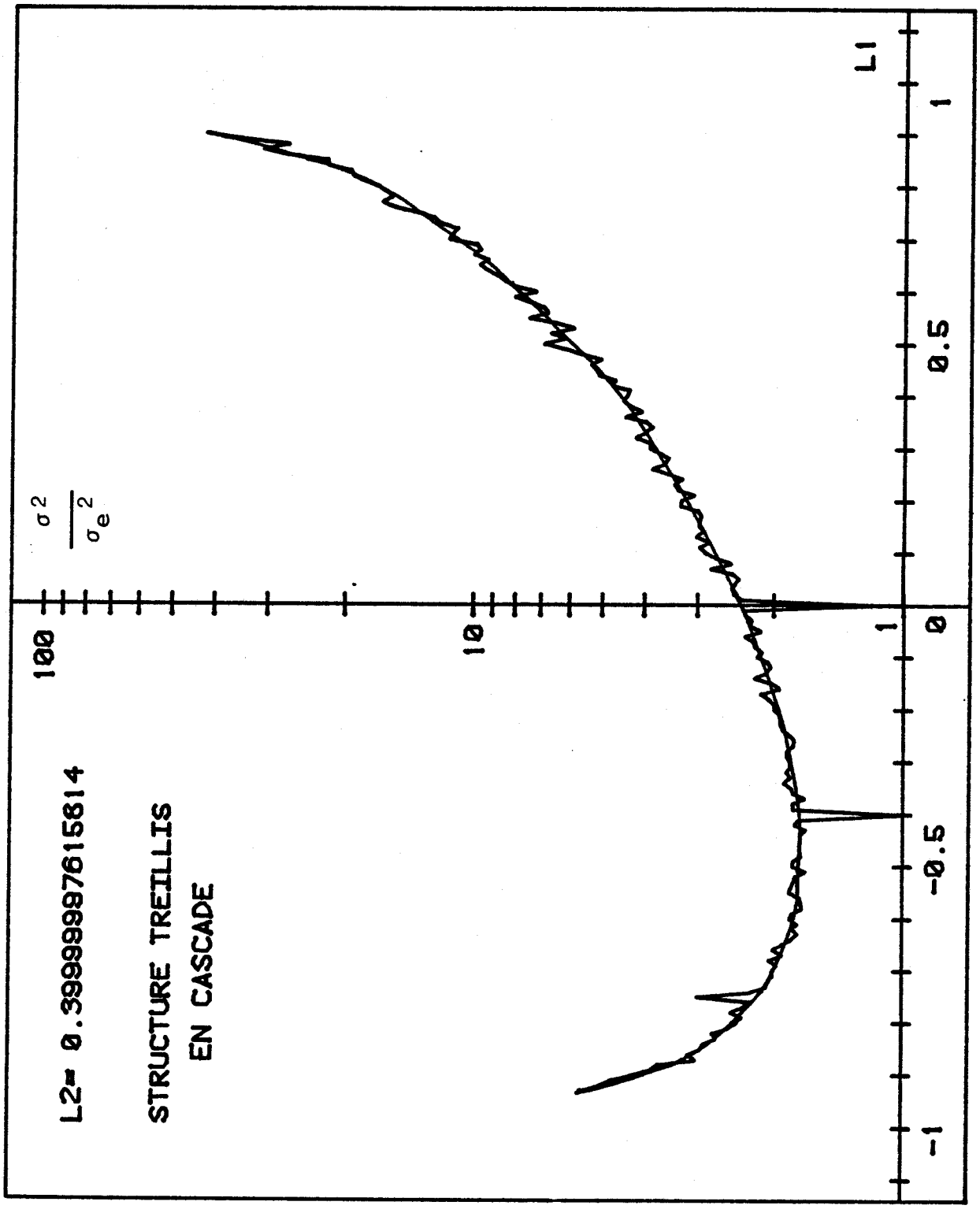


Figure II.11 - *Comparaison entre les résultats théoriques et expérimentaux de la puissance de bruit en fonction des pôles pour le filtre FTTC.*

On remarque sur ces figures l'excellente correspondance des résultats issus des expressions théoriques et des simulations des filtres réels. Toutefois on constate quelques points de divergence qui correspondent au restrictions qui ont été faites dans les paragraphes II.3 en ce qui concerne la quantification des coefficients des multiplieurs. Ces effets sont principalement visibles dans les cas où les filtres présentent des multiplieurs ayant pour coefficient 0.5, 0.25 ou 0.75 principalement.

II.6.3. Etude comparative des bruits de multiplication engendrés par trois types de structures.

Cette étude a été effectuée en considérant trois structures qui réalisent une même fonction de transfert H(z). Les filtres sont généralement construits à l'aide de structures de premier et du second ordre. Nous avons donc étudié systématiquement un filtre de fonction de transfert du type :

$$H(z) = \frac{1}{1 + b_1 . z^{-1} + b_2 . z^{-2}} \qquad (2.24)$$

qui a été simulée pour les trois types de réalisations représentées par les schémas (a), (b) et (c).

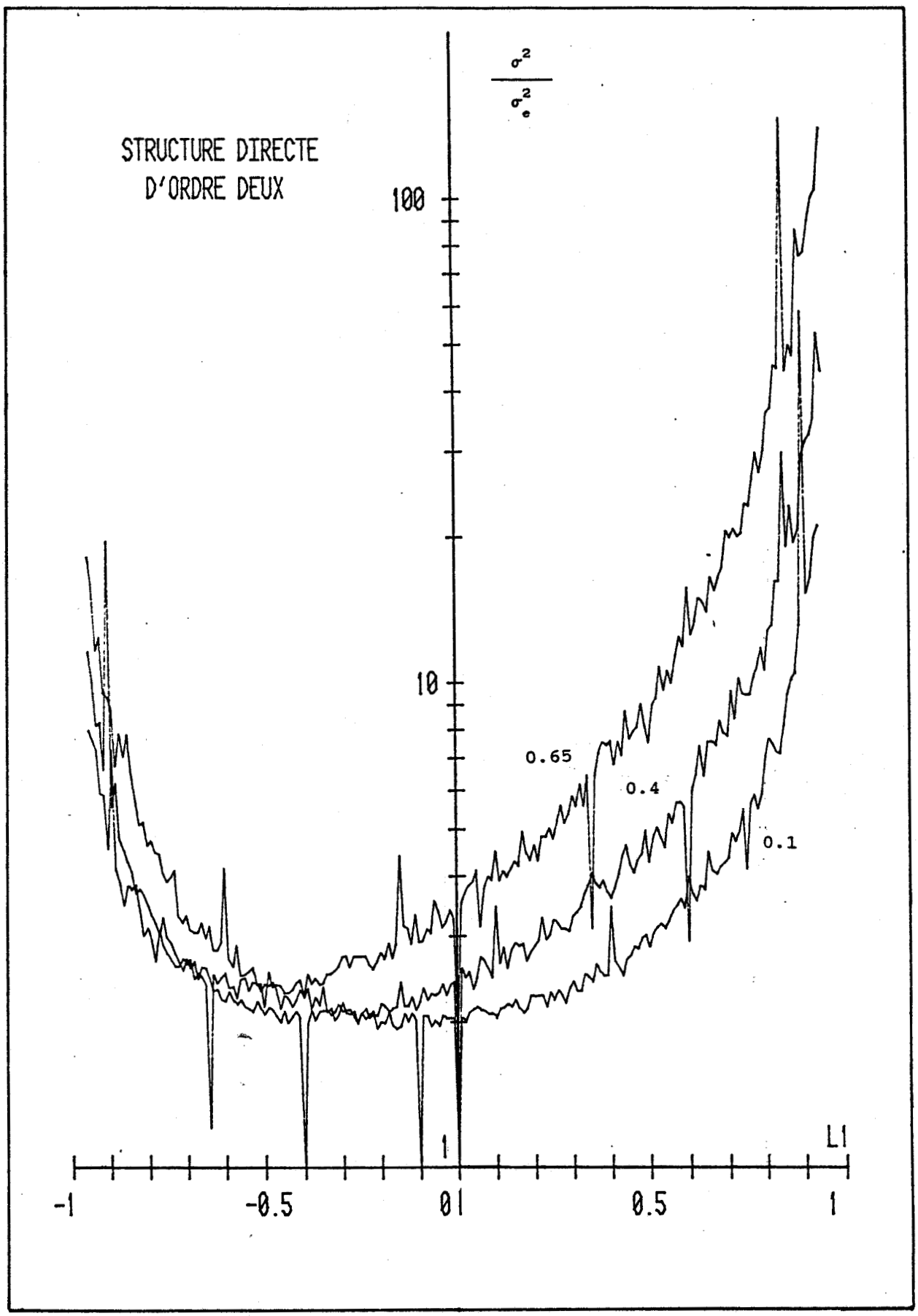


Figure II.12 - *Puissance réduite de bruit en sortie du filtre FTDIIC en fonction du pôle L_1 pour les valeurs 0.1,0.4 et 0.65 du pôle L_2.*

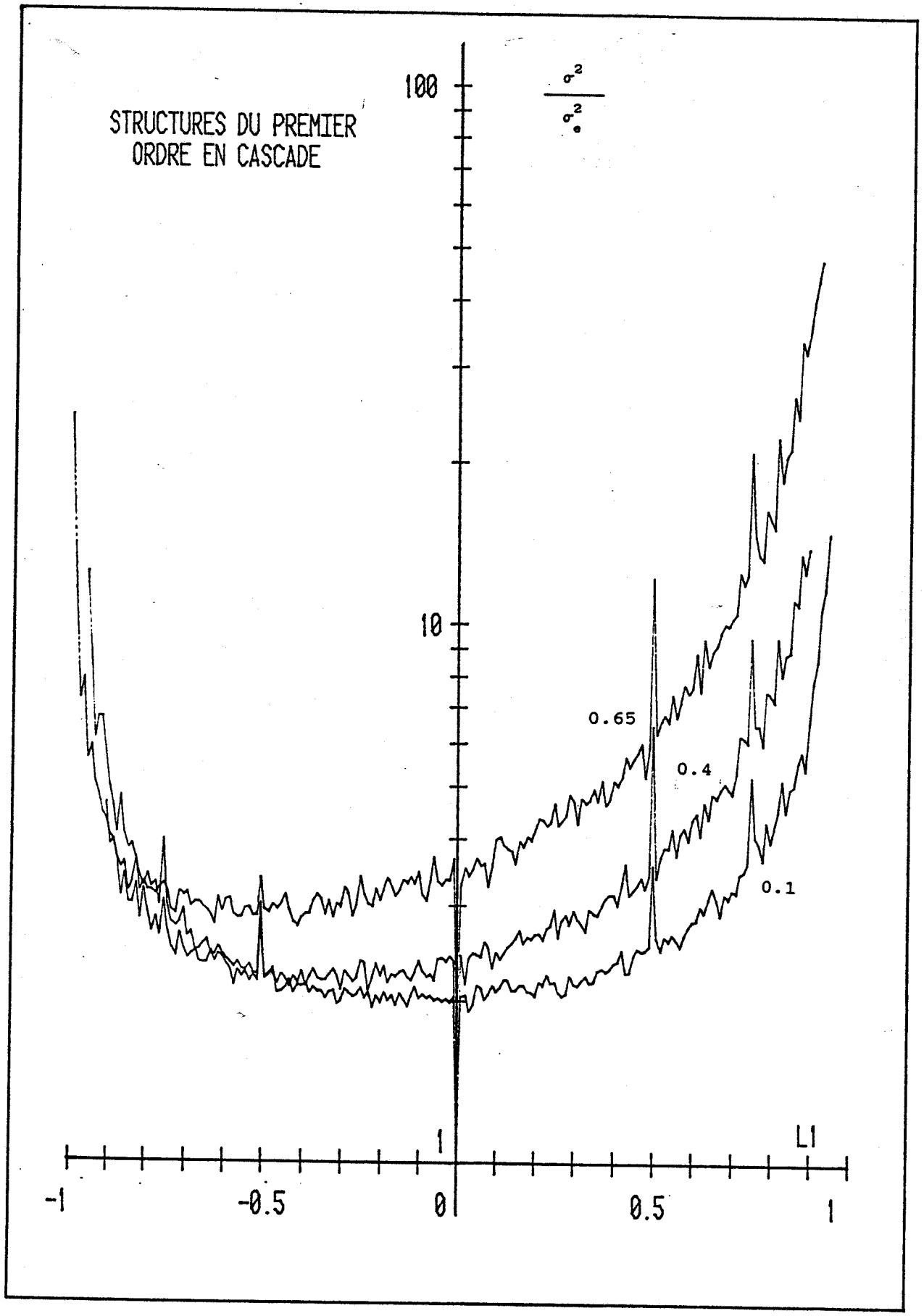


Figure II.13 - *Puissance réduite de bruit en sortie du filtre FTDIC en fonction du pôle L_1 pour les valeurs 0.1,0.4 et 0.65 du pôle L_2.*

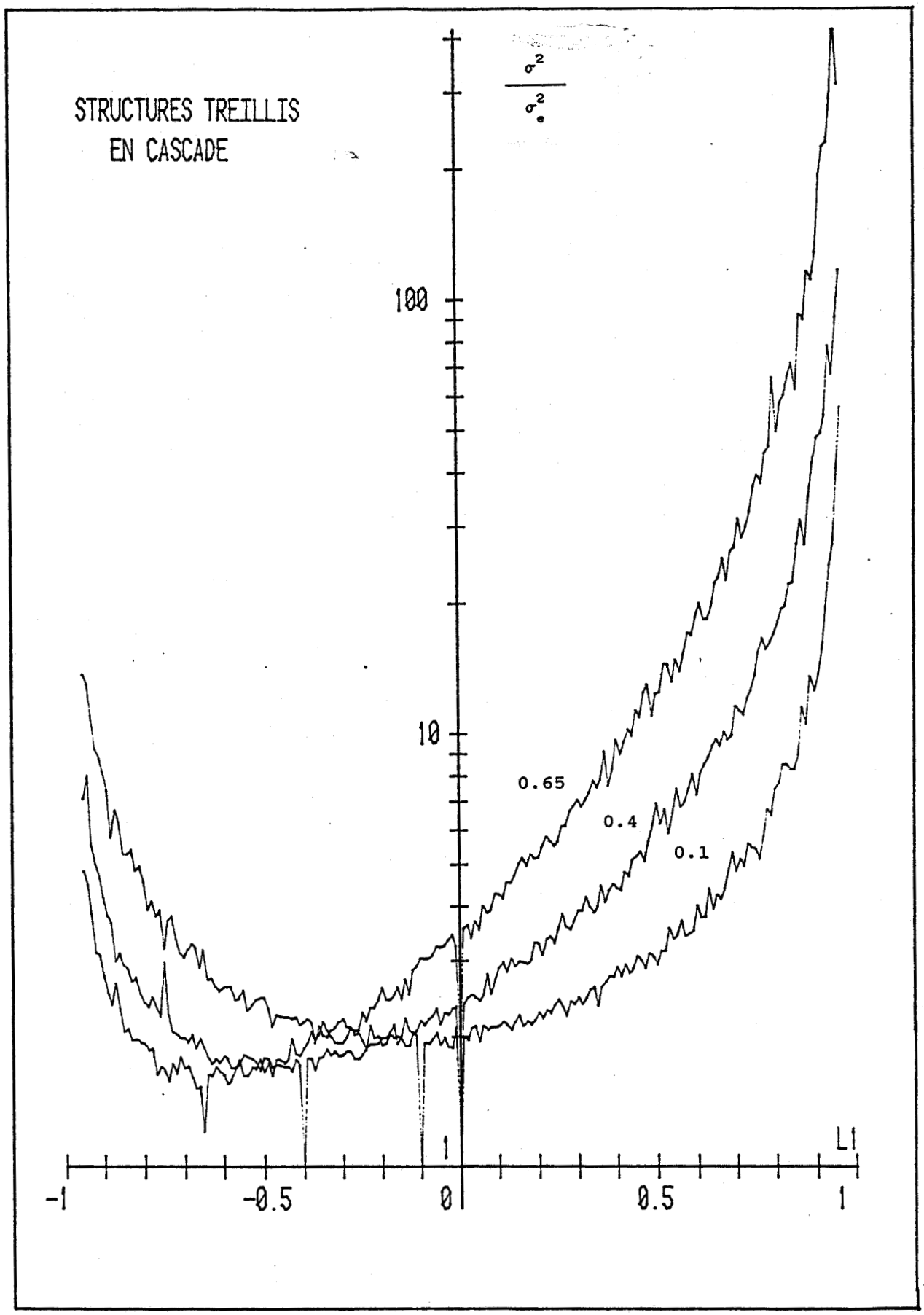


Figure II.14 - *Puissance réduite de bruit en sortie du filtre FTTC en fonction du pôle L_1 pour les valeurs 0.1,0.4 et 0.65 du pôle L_2.*

Ne sont données ici que les courbes dans le cas où le pôle L_2 est positif car on vérifie expérimentalement que la puissance de bruit réduite présente une symétrie de représentation par rapport à l'axe $L_1 = 0$, suivant la relation :

$$\frac{\sigma^2}{\sigma_e^2} = f(L_1 , L_2) = f(-L_1 , -L_2) \qquad (2.25)$$

prévisible à partir des estimations théoriques.

Le fait de représenter la puissance réduite de bruit en sortie du filtre en fonction des pôles de la fonction de transfert permet de pouvoir comparer immédiatement les différentes structures entre elles. Dans ce cas, si nous définissons une relation d'ordre notée : " > " par a > b si la puissance réduite de bruit en sortie de la structure a est plus faible que celle en sortie de b et ceci pour le même couple de pôles L_1, L_2, les courbes permettent de donner immédiatement les relations suivantes :

$$\begin{array}{lll} \text{si } L_2 > 0 & \text{si } L_1 < 0 & \text{FTTC} > \text{FTDIIC} > \text{FTDIC} \\ & \text{si } L_1 > 0 & \text{FTDIC} > \text{FTDIIC} > \text{FTTC} \end{array} \qquad (2.26)$$

si L2 < 0 les résultats sont inversés d'après la remarque de symétrie.

$$\begin{array}{ll} \text{si } L_1 < 0 & \text{FTDIC} > \text{FTDIIC} > \text{FTTC} \\ \text{si } L_1 > 0 & \text{FTTC} > \text{FTDIIC} > \text{FTDIC} \end{array} \qquad (2.27)$$

CHAPITRE III

III.1. INTRODUCTION.

Comme nous l'avons vu, la réalisation d'un filtre de caractéristiques données n'est pas unique et les performances en terme de bruit de ces différentes réalisations ne sont pas identiques. Il doit donc être possible de trouver une structure à bruit minimale.

III.2. OPTIMISATION DE LA STRUCTURE D'UN FILTRE NUMERIQUE.

III.2.1. Valeur minimale de la puissance réduite de bruit.

. Considérons un filtre donné par ces équations d'état :

$$x(n+1) = A.x(n) + B.u(n)$$
$$y(n) = C.x(n) + D.u(n)$$

il a été montré [39] que la valeur minimale de la puissance réduite de bruit σ^2 / σ^2_e en sortie d'un filtre d'ordre N pouvait se mettre sous la forme :

$$\frac{\sigma^2}{\sigma^2_e} = N. \left[\det (KW) \right]^{1/N} \qquad (3.1)$$

avec W la matrice de bruit précédemment définie et K la matrice dite de covariance du filtre que l'on définie et calcule de façon analogue à W suivant la relation :

$$K = AKA^T + BB^T = \sum_{k=0}^{\infty} (A^k B).(A^k B)^T \qquad (3.2)$$

Cette valeur théorique ne peut toutefois être atteinte que si toutes les valeurs propres de la matrice KW sont identiques. Dans le cas contraire, la valeur minimale que pourra prendre la puissance réduite de bruit après transformation est donnée par :

$$\frac{\sigma^2}{\sigma_e^2} = \left[\sum_{i=1}^{N} \Theta_i \right]^{2/N} \qquad (3.3)$$

où les Θ_i sont les racines carrées des valeurs propres de la matrice KW.

III.2.2. Techniques d'optimisation.

On sait que si (A,B,C,D) est la représentation dans l'espace d'état d'un filtre de fonction de transfert en z, H(z),alors le filtre de représentation (A',B',C',D') avec :

$$\begin{aligned} A' &= T^{-1}AT \\ B' &= T^{-1}B \\ C' &= CT \\ D' &= D \end{aligned} \qquad (3.4)$$

où T est une matrice d'ordre N non singulière, à la même fonction de transfert. Il existe une infinité de transformations T et parmi celles-ci il doit s'en trouver une au moins qui minimise la puissance réduite de bruit. On montre [39] que cette transformation T_{min} peut se mettre sous la forme :

$$T_{min} = T_0.R_1.L.R_0^T \qquad (3.5)$$

avec T_0,R_1,L et R_0 qui vérifient :

$$T_0^{-1} K T_0^{-T} = I \quad \text{c'est à dire} \quad K = T_0 T_0^T \qquad (3.6)$$

$$R_0 \left[L^{-1} L^{-1} \right] R_0^T = \begin{bmatrix} 1 & & x \\ & 1 & \\ x & & 1 \end{bmatrix}$$

$$R_1^T \left[T_0^T W T_0 \right] R_1 = \begin{bmatrix} \Theta_1^2 & & 0 \\ & \ddots & \\ 0 & & \Theta_N^2 \end{bmatrix} \tag{3.7}$$

$$L = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_N \end{bmatrix} \qquad \text{avec} \qquad \lambda_i = \frac{\left[\sum_{j=1}^{N} \Theta_j \right]^{1/2}}{N.\Theta_i}$$

La matrice T_{min} est donc obtenue en calculant les valeurs propres de la matrice KW et en procédant à des techniques de diagonalisation et de réduction des termes diagonaux des matrices. Le filtre ainsi conçu a la propriété supplémentaire de prendre en compte la mise à l'échelle des données.

IL existe d'autres méthodes pour arriver à une structure optimale en terme de bruit. Citons par exemple, les techniques qui consistent à minimiser la sensibilité de la réponse fréquentielle du filtre tout en arrivant à une structure à bruit minimal [40]. Ces techniques sont également basées sur la recherche d'une transformation T adéquate. Il faut toutefois remarquer que ces techniques d'optimisation provoquent en général une augmentation non négligeable de la complexité du filtre. La méthode de réduction de sensibilité conduit par exemple, pour un filtre à $2n+1$ multiplieurs à une structure optimale comportant $(n+1)^2$ multiplieurs. Il est donc nécessaire de limiter cette augmentation.

III.2.3. Réduction du nombre de multiplieurs d'une structure optimale.

Il est possible de réduire le nombre de multiplications non triviales qui existent dans une structure optimale [41]. Cette technique est basée sur le remplacement de certains coefficients de multiplieurs par des puissances entières de 2 tout en contrôlant la dégradation du bruit d'arrondi. Pour ce faire on considère dans la structure optimale un coefficient proche d'une puissance entière de 2. On remplace alors ce coefficient par la puissance de 2 correspondante. Il faut ensuite compenser cette approximation au moyen d'une transformation T qui va répercuter sur les coefficients restant, le changement ainsi effectué tout en conservant les contraintes de minimisation du bruit et éventuellement de mise à l'échelle. Pour cela, on considère que la transformation T est paramètrée et que pour toute transformation $T(\alpha)$ les contraintes doivent être identiques ce qui se traduit par le fait que les dérivées de ces contraintes

par rapport à α doivent être nulles. Dans le cas où cela n'est pas possible, on abandonne le coefficient traité à sa valeur initiale et on recommence l'opération avec un autre coefficient. On procède ainsi par itération sur tous les coefficients jusqu'à leur épuisement. Il faut toutefois noter qu'à chaque étape de la transformation il est nécessaire de rajouter une contrainte supplémentaire correspondant à l'invariance après la nouvelle transformation, des coefficients déjà ramenés à des puissances entières de 2.

On arrive ainsi à transformer un certain nombre de coefficients en puissances entières de 2 ce qui simplifie de beaucoup les circuits de multiplication et ceci au prix d'une légère augmentation de la puissance réduite de bruit. En effet la structure obtenue bien que très proche, n'est plus optimale. On peut citer l'exemple [42] d'une structure d'ordre 3 qui nécessite 7 multiplieurs et qui conduit à une structure optimale de 16 multiplieurs que l'on peut ramener à 11 multiplications non triviales et 5 multiplications sous forme de puissances de 2. Ceci se fait au prix d'une augmentation de la puissance de bruit de 4.3 % que l'on doit rapprocher des 384 % de diminution obtenue par l'optimisation.

III.3. COMPENSATION SPECTRALE DE L'ERREUR.

Parmi les techniques de réduction de bruit en sortie d'un filtre, on peut également citer la technique de la compensation spectrale d'erreur [43]. Il s'agit plus, comme son nom l'indique, d'une technique de compensation d'erreur que d'un procédé d'optimisation. Pour ce faire, il est nécessaire d'ajouter au filtre un circuit de traitement des erreurs. Le principe est de reporter l'erreur en action sur la sortie du filtre et en contre réaction sur les additionneurs internes comme le montre la figure III.1.

Ceci suppose toutefois que la quantification s'effectue au niveau des additionneurs. Il faut de plus que le circuit supplémentaire travaille en double précision afin de pouvoir prendre en compte les erreurs du circuit de départ. Ce circuit est alors accordé de manière à minimiser l'erreur de quantification.

Cette technique fonctionne assez bien pour des filtres basses bas de type Bessel, Butterworth ou Tchebychev synthétisés à partir d'une transformation bilinéaire. C'est à dire qui possède un pôle en z = -1. L'intérêt de cette technique reste cependant limité puisqu'elle complique considérablement le filtre et que le résultat obtenu et en fait comparable à celui que fournirait un filtre travaillant avec un arithmétique de plus grande précision.

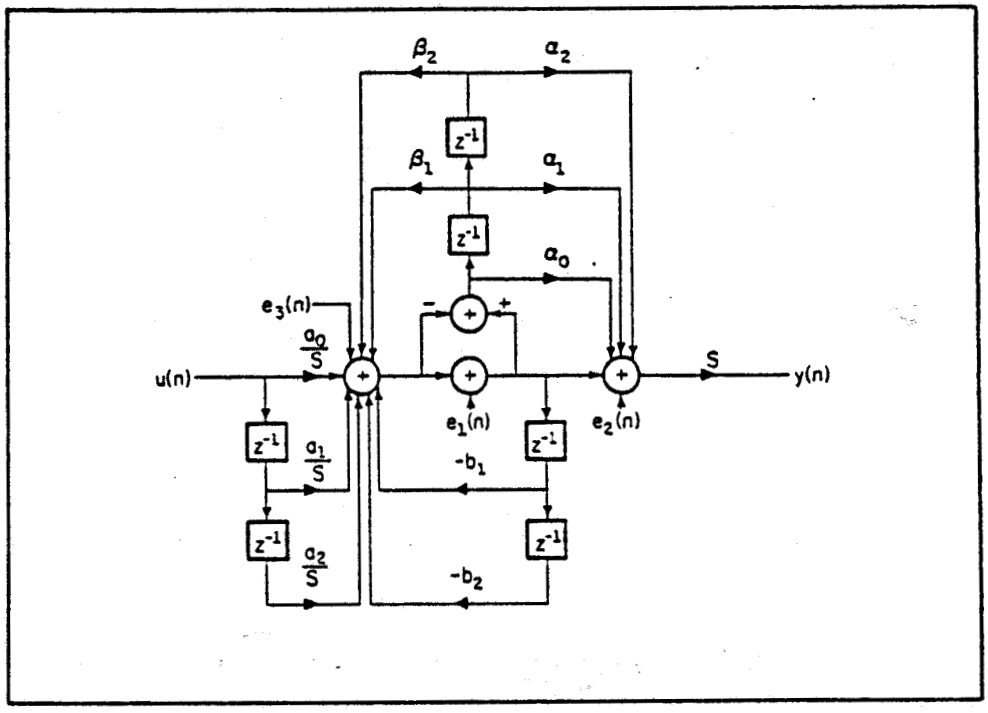


Figure III.1 - *Compensation d'erreur.*

III.4. ARITHMETIQUE DISTRIBUEE.

La technique de l'arithmétique distribuée n'est pas à proprement parler une méthode de réduction de bruit. Elle rentre plutôt dans une ligne de pensée qui consiste à chercher une alternative au problème du bruit engendré par les multiplieurs classiques.

III.4.1. Multiplieurs classiques.

Il existe plusieurs façon de réaliser un multiplieur numérique. Nous rappelons dans un premier temps (d'après [12]) trois des techniques les plus courantes.

III.4.1.1. Le multiplieur série.

Ce multiplieur qui réalise le produit P = C.D est donné par le schéma suivant :

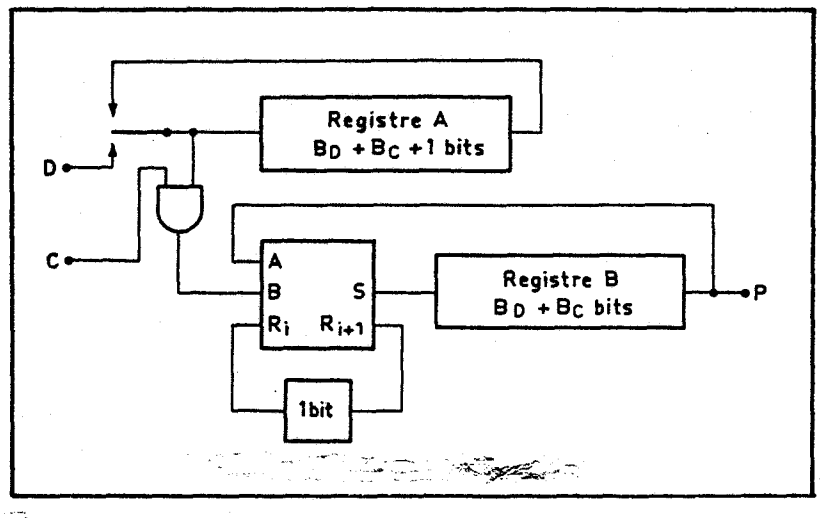


Figure III.2 - *Multiplieur série.*

Il a l'avantage d'être aisément intégrable sur circuit monolithique mais reste limité dans sa capacité de calcul.

III.4.1.2. Le multiplieur série-parallèle.

Ce multiplieur représenté en figure III.3 est d'une plus grande efficacité de calcul. On le trouve également sous sa variante à accès série qui permet une présentation série des 2 termes à multiplier.

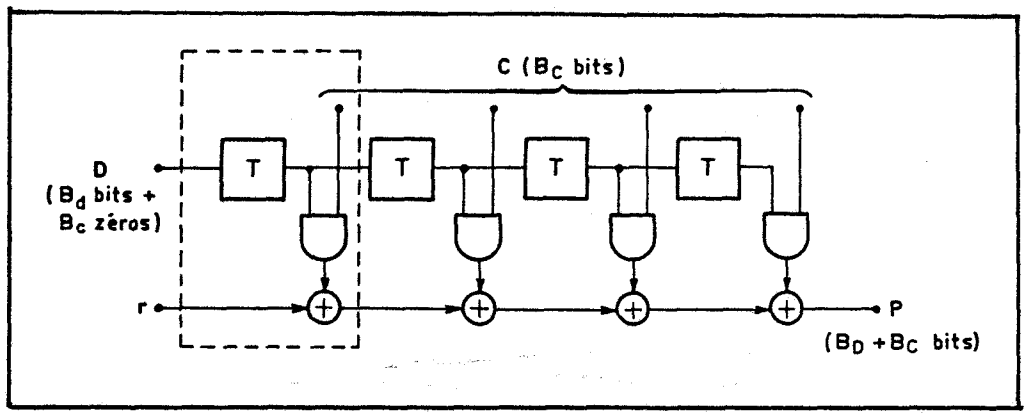
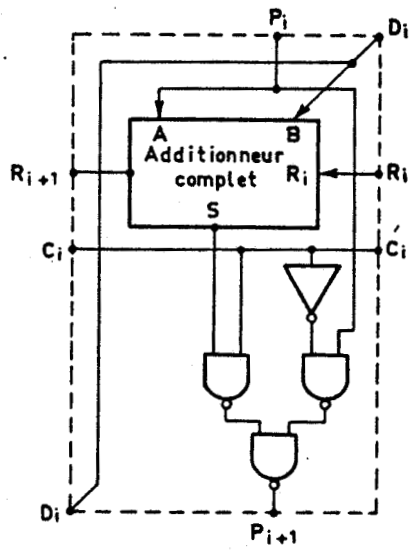


Figure III.3 - *Multiplieur série parallèle.*

III.4.1.3. Le multiplieur parallèle.

C'est le multiplieur qui a la plus grande rapidité de calcul compte tenu de la mise en parallèle des opérations comme le montre la figure III.4. Il a par ailleurs le désavantage d'être complètement figé quant au nombre de bits des facteurs qu'il doit traiter.



Cellule élémentaire.

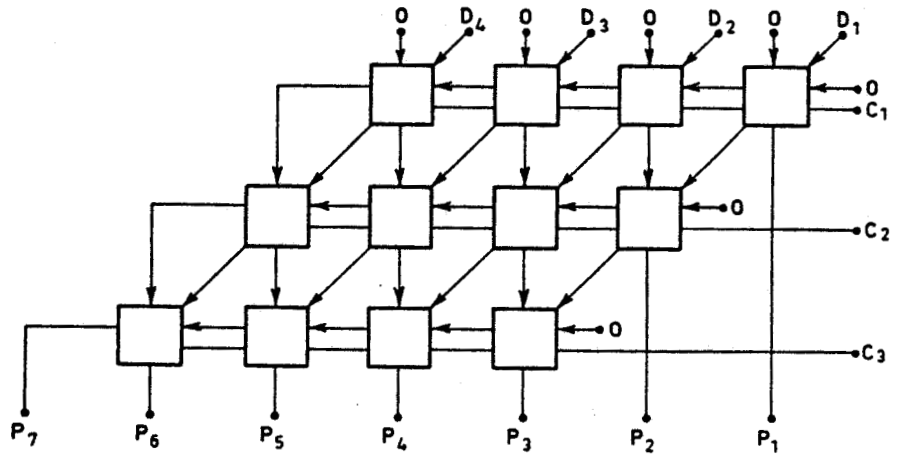


Figure III.4 - *Multiplieur parallèle.*

Quelque soit la forme adoptée, il résulte de ce qui précède que les multiplieurs classiques génèrent un bruit d'arrondi de façon structurelle. Les sources de bruit étant localisées au niveau de ces opérateurs arithmétiques. Partant de cette constatation, la technique de l'arithmétique distribuée consiste dans son principe, à considérer une relation arithmétique non plus comme une succession d'opérations arithmétiques localisées mais comme un tout ne possédant qu'un nombre de réponses fini.

III.5. PRINCIPE DES FILTRES A ARITHMETIQUE DISTRIBUEE (FAD).

III.5.1. Structure FAD conventionnelles.

Considérons donc la réalisation de la fonction de filtrage numérique définie par la relation :

$$y(n) = \sum_{i=0} a_i . x(n-i) + \sum_{j=0} b_j . y(n-j) \qquad (3.8)$$

où y(n) est la sortie du système à l'instant nT_e et x(n) l'entrée au même instant. Tous les nombres seront pris dans leur représentation en complément à 2 sur un format de B bits. On adoptera de plus la notation suivante :

x(i,n) ϵ { 0,1 } représente le $i^{ème}$ bit de x(n)

y(i,n) ϵ { 0,1 } représente le $i^{ème}$ bit de y(n)

le bit 0 étant le bit de signe.

Si l'on définit une fonction phi par :

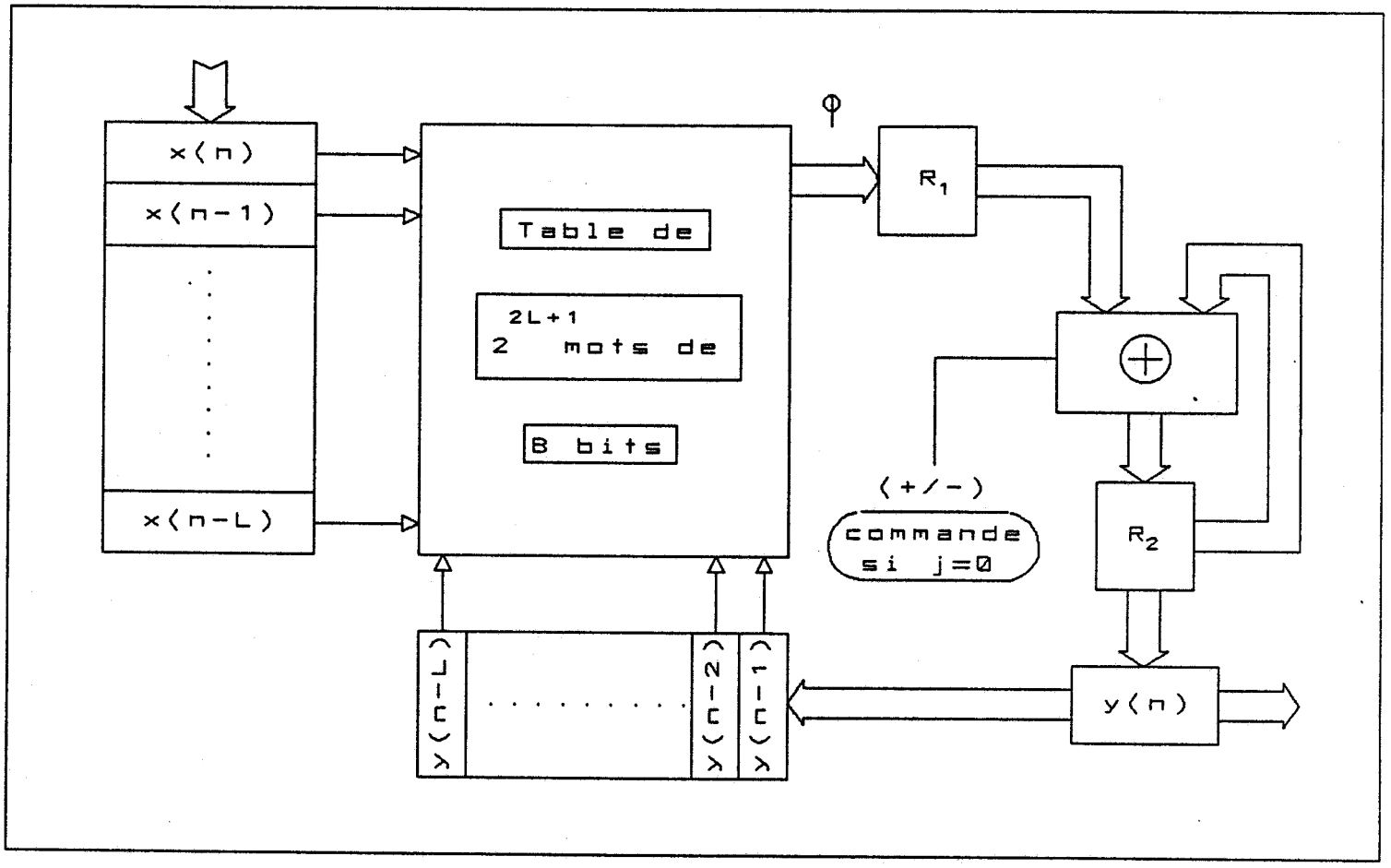
$$\phi_n(i) = \sum_{j=1}^{L} a_j . x(i,n-j) + \sum_{j=1}^{L} b_j . y(i,n-j) \qquad (3.9)$$

il devient possible de ré-écrire la relation (3.8) sous la forme :

$$y(n) = -\phi_n(0) + \sum_{i=0}^{B-1} \phi_n(i) . 2^{-i} \qquad (3.10)$$

(il est nécessaire de faire une distinction pour le cas i=0 puisqu'il correspond au bit de signe des mots en complément à 2).

La fonction phi ainsi définie se présente en fait comme la somme des a_i, b_j pondérés par des 0 et des 1. Le nombre de combinaisons possibles de ces sommes pondérées est fini. Dans notre cas il est de 2^{2L+1}. Il est donc possible de pré-calculer ces résultats et de les stocker dans une table mémoire. La figure III.5 donne alors une réalisation possible de la fonction de filtrage définie par l'équation (3.8).



III.5 - *Structure de filtre à arithmétique distribuée.*

A chaque top d'horloge un nouveau vecteur :

$$\{ x(i,n), x(i,n-1), \ldots, x(i,n-L), y(i,n-1), \ldots, y(i,n-L) \}$$

adresse un mot de la table générant la fonction phi. Le résultat est un mot de B bits qui est stocké dans R_1. On lui additionne (ou soustrait si i=1) le précédent

résultat stocké dans R_2 après l'avoir décalé d'un bit vers la droite. Cette opération est répétée B fois en commençant par les bits de poids faible des x et des y. Finalement le registre R_2 contient un mot qui donnera après retour sur B bits la valeur de y à l'instant nT. La valeur de y une fois transférée, les registres R_1 et R_2 sont réinitialisés à zéro en vue du démarrage d'un nouveau groupe de B cycles du type précédents.

Cette réalisation matérielle n'est évidemment pas la seule possible. De nombreuses variantes ont été proposées [44][45] qui répondent en général à des besoins de diminution des tailles mémoires ou d'augmentation des vitesses de traitement. Ces variantes se caractérisent par la mise en pratique de deux techniques principales :

a - La somme de l'équation (3.8) qui donne y(n) peut être décomposée en k sommes partielles de chacune m_i termes de sorte que l'on ait bien entendu :

$$\sum_{i=1}^{k} m_i = L \qquad (3.11)$$

Ceci aura pour effet de permettre une réalisation ne nécessitant que k mémoires de 2^{mi} mots respectivement au lieu d'une seule de 2^L mots. C'est à dire une taille mémoire globale de :

$$\sum_{i=1}^{k} 2^{m_i}$$

au lieu de 2^L. Si l'on se fixe par exemple L = 10 et k = 2 et $m_1 = m_2 = 5$ la réalisation précédente nécessite une taille mémoire de 1024 mots alors qu'après partitionnement de la somme y, la taille mémoire nécessaire n'est plus que de 64 mots (2.2^5).
Ce gain se fait naturellement au détriment du nombre de composants présents dans la réalisation et nécessite de plus l'addition des deux résultats générés par les k tables-mémoires. Au total le gain en taille mémoire se fait au détriment du nombre de composants et de la vitesse de traitement ce qui correspond à une diminution de la fréquence maximale d'utilisation du filtre.

b - L'autre technique consiste à adresser les tables mémoires avec plusieurs bits de chaque mots au même top d'horloge, au lieu d'un seul comme précédemment. Ce faisant on augmente la vitesse d'exécution du système mais cette fois au détriment de la taille mémoire nécessaire pour réaliser la structure. Si l'on veut utiliser p bits par mots la taille mémoire nécessaire passe de 2^L à 2^{pL}.

Naturellement ces deux techniques peuvent coexister au sein d'une même réalisation pour fournir le meilleur compromis possible entre la vitesse d'exécution et la taille mémoire nécessaire. La réalisation matérielle doit aussi prendre en compte des paramètres supplémentaires telles que l'encombrement des circuits ou la puissance consommée par le système. De ce point de vue il est également nécessaire de trouver un compromis entre les deux techniques précédemment décrites. Des études comparatives ont été faites qui tiennent compte de ces données (confère annexe 5).

III.5.2. Structures FAD « Direct II ».

En s'inspirant des travaux de A. PELED et B. LIU [44] et en utilisant une propriété de la représentation dans l'espace d'état des structures dite « Directe » d'ordre L, F.J. TAYLOR a proposé un nouveau type de réalisations dite « Direct II » utilisant les techniques de l'arithmétique distribuée [46]. En effet dans l'espace d'état les structures directes ont pour équations:

$$X(n+1) = A.x(n) + B.u(n) \tag{3.12}$$

$$Y(n) = C.x(n) + d.u(n) \tag{3.13}$$

avec

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 \\ \cdot & & & & & \\ \cdot & & & & & \\ -a_1 & -a_2 & -a_3 & -a_4 & \dots & -a_L \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \cdot \\ \cdot \\ k \end{bmatrix}$$

$$C = \begin{bmatrix} c_1 & c_2 & c_3 & \dots & c_L \end{bmatrix} \qquad d = d_0$$

En utilisant le fait que la variable d'état $x_i(n+1) = x_{i+1}(n)$ on peut réaliser les équations (3.12) et (3.13) séparément suivant le schéma de la figure III.6.

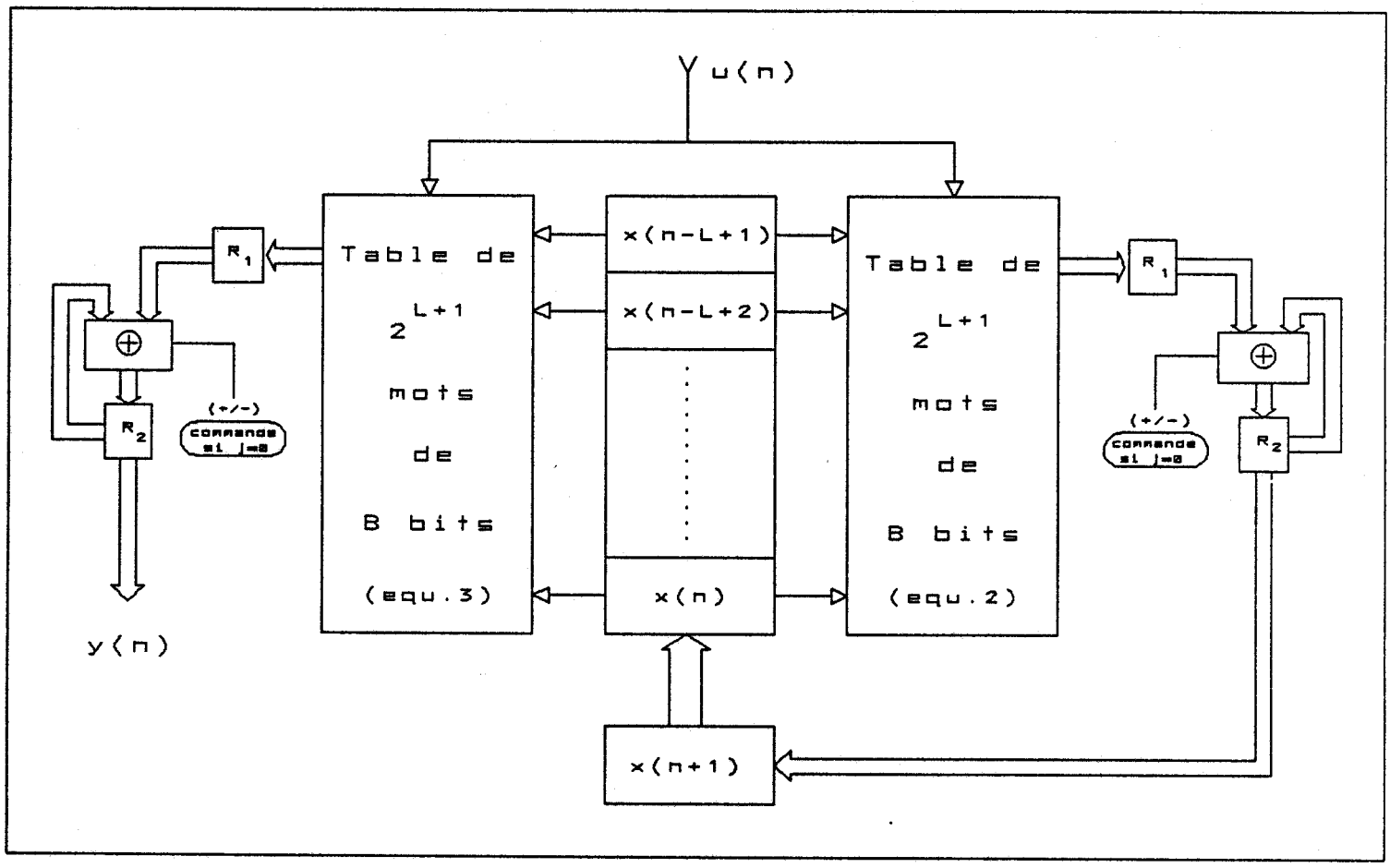


Figure III.6 - *Structure Directe à arithmétique distribuée.*

F.J. TAYLOR montre que l'on peut estimer, moyennant les hypothèses habituelles, la variance du bruit d'arrondi en sortie de cette structure par la formule :

$$\sigma_e^2 = \frac{7.q^2}{36} \left\| H(z) \right\|^2 + \frac{q^2}{9} \qquad (3.14)$$

avec q le quantum de base et H(z) la fonction de transfert en z du filtre.

III.5.3. Structures FAD pour une représentation quelconque dans l'espace d'état.

La structure précédente n'est utilisable que pour un type particulier de représentation dans l'espace d'état. Nous proposons maintenant une structure permettant la réalisation d'un filtre ayant une représentation quelconque dans l'espace d'état. Les matrices d'état sont cette fois quelconques, de la forme :

$$A = (a_{ij})$$

$$B = (b_i)$$

$$C = (c_j)$$

$$D = d$$

et toujours les relations :

$$\begin{aligned} x(n+1) &= A.x(n) + B.u(n) \\ y(n) &= C.x(n) + d.u(n) \end{aligned} \qquad (3.15)$$

En admettant que le système soit d'ordre L, on peut écrire les L+1 équations suivantes :

$$\left\{ \begin{array}{l} x_1(n+1) = \sum_{i=1}^{L} a_{1i}.\, x_i(n) + b_1.u(n) \\ \quad \vdots \\ x_k(n+1) = \sum_{i=1}^{L} a_{ki}.\, x_i(n) + b_k.u(n) \\ \quad \vdots \\ x_L(n+1) = \sum_{i=1}^{L} a_{Li}.\, x_i(n) + b_L.u(n) \\ y(n) = \sum_{i=1}^{L} c_i.x_i(n) + d.u(n) \end{array} \right. \qquad (3.16)$$

que l'on peut réaliser sous la forme d'une structure à arithmétique distribuée utilisant L+1 tables mémoires suivant le schéma de la figure III.7.

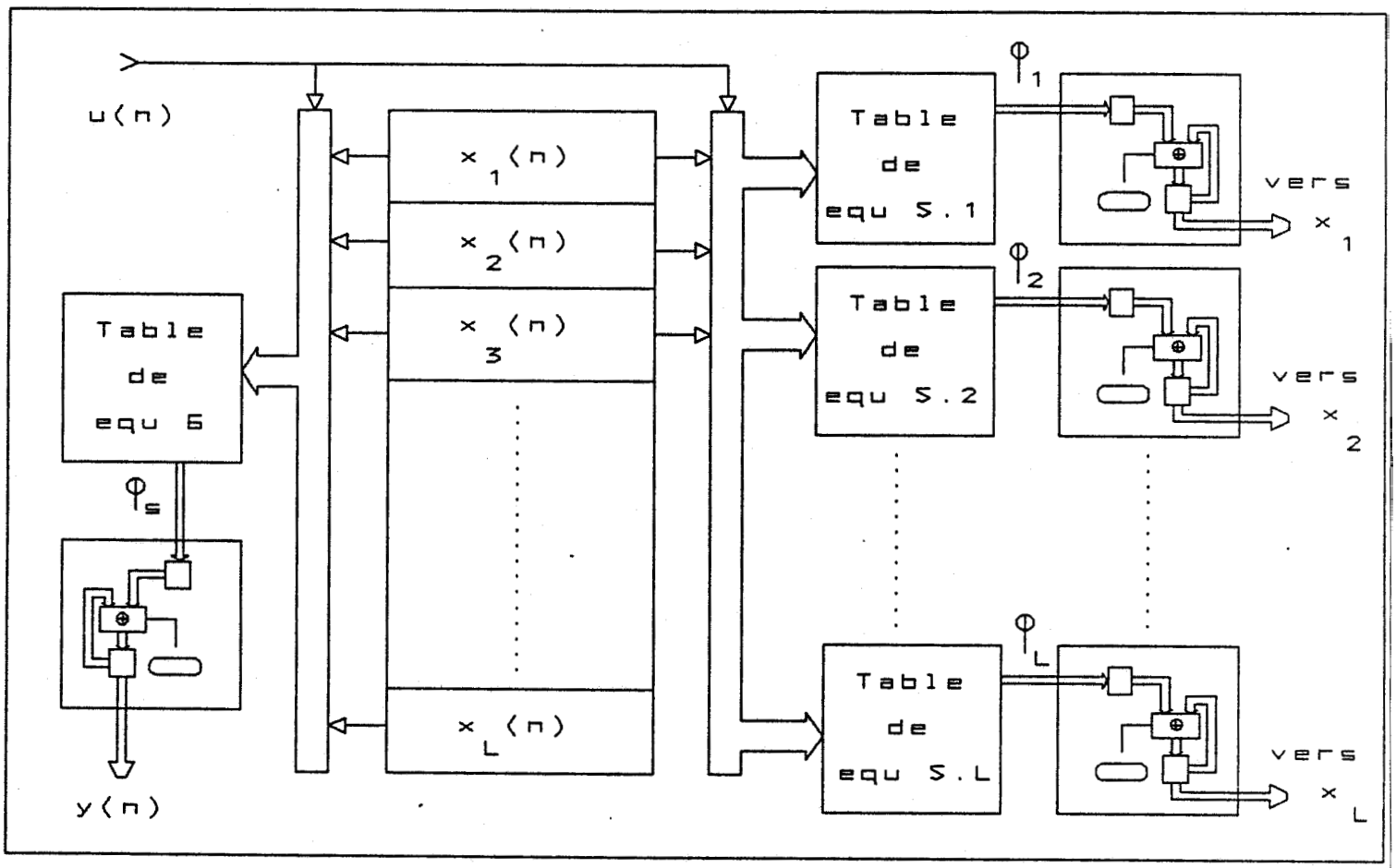


Figure III.7 - *Structure utilisant l'arithmétique distribuée d'un filtre à représentation quelconque dans l'espace d'état.*

III.6. ANALYSE DU BRUIT DANS LES STRUCTURES PROPOSEES.

III.6.1. Résolution dans le cas général.

Considérions d'abord le cas des L premières équations du système (3.16) donné précédemment. Elles correspondent à la partie récursive du système.

Posons les relations suivantes (une variable surligné signifie une valeur réellement calculée par rapport à une valeur théorique) :

$$\begin{aligned} e_i' &= \bar{x}_i - x_i \\ e_i'' &= \overline{\phi(i)} - \phi(i) \\ e''' &= \text{erreur d'arrondi en sortie du registre avec décalage à droite.} \end{aligned} \tag{3.17}$$

Nous avons pour la $k^{ième}$ variable $x_k(n)$ en théorie :

$$x_k(n+1) = \sum_{i=1}^{B-1} \phi_{nk}(i).2^{-i} - \phi_{nk}(0) \tag{3.18}$$

Mais en réalité on calcule :

$$\overline{x_k(n+1)} = \sum_{i=1}^{B-1} \overline{\phi_{nk}(i).2^{-i}} - \overline{\phi_{nk}(0)} \tag{3.19}$$

que l'on peut décomposer comme suit :

$$\overline{x_k(n+1)} = \sum_{i=1}^{B-1} \overline{\phi_{nk}(i)}.2^{-i} - \phi_{nk}(0) + \sum_{i=1}^{B-1} e''_i(n) \cdot 2^{-i} + e''_0(n) \quad (3.20)$$

soit encore en remplaçant la fonction phi par son expression :

$$\overline{x_k(n+1)} = \overbrace{\sum_{i=1}^{B-1} \left[\left(\sum_{j=1}^{L} a_{kj}.x_j(i,n) + b_k.u(i,n) \right) \right] \cdot 2^{-i} - \phi_{nk}(0) \ \ldots}^{x_k(n+1)}$$
$$\ldots + \sum_{i=1}^{L} a_{ki}.e'_i(n) + e'''(n) + \sum_{i=1}^{B-1} e''_i(n) \cdot 2^{-i} + e''_0(n) \quad (3.21)$$

Dans cette équation on reconnait la valeur développée de $x_k(n+1)$. On en déduit donc immédiatement une expression de $e'_k(n+1)$:

$$e'_k(n+1) = \overline{x_k(n+1)} - x_k(n+1) \quad (3.22)$$

d'où :

$$e'_k(n+1) = \sum_{i=1}^{L} a_{ki}.e'_i(n) + e'''(n) + \sum_{i=1}^{B-1} e''_i(n) \cdot 2^{-i} + e''_0(n) \quad (3.23)$$

Si l'on pose :

$$\delta_k(n) = e'''(n) + \sum_{i=1}^{B-1} e''_i(n) \cdot 2^{-1} + e''_0(n) \qquad (3.24)$$

$e'_k(n+1)$ peut encore se mettre sous la forme :

$$e'_k(n+1) = \sum_{i=1}^{L} a_{ki} \cdot e'_i(n) + \delta_k(n) \qquad (3.25)$$

Il est alors possible de regrouper dans le premier membre de l'équation tous les termes en e' :

$$e'_k(n+1) - a_{kk} \cdot e'_k(n) = \sum_{\substack{i=1 \\ i=k}}^{L} a_{ki} \cdot e'_i(n) + \delta_k(n) \qquad (3.26)$$

L'équation devient après transformée en z compte tenu du théorème du retard:

$$e'_k(z) \cdot \left[1 - a_{kk} \cdot z^{-1} \right] = \sum_{\substack{i=1 \\ i=k}}^{L} a_{ki} \cdot e'_i(z) \cdot z^{-1} + z^{-1} \cdot \delta_k(z) \qquad (3.27)$$

Soit encore :

$$e'_k(z) = \sum_{\substack{i=1 \\ i=k}}^{L} \frac{a_{ki}.z^{-1}}{1 - a_{kk}.z^{-1}} \cdot e'_i(z) + \frac{z^{-1}}{1 - a_{kk}.z^{-1}} \cdot \delta_k(z) \qquad (3.28)$$

Cette relation étant valable quelque soit k, on obtient en fait un système de L équations à L inconnues (les $e'_k(z)$) que l'on doit résoudre par rapport aux $\delta_k(z)$. Ceci conduit donc à des solutions de la forme :

$$e'_k(z) = \sum_{i=1}^{L} H_{ki}(z) \cdot \delta_i(z) \qquad (3.29)$$

Il reste maintenant à traiter la L+1 nième équation du système de manière à obtenir une expression du bruit e(n) en sortie de la structure. De façon tout à fait analogue au calcul précédent, il est possible de mettre e(n) sous la forme :

$$e(n) = \overline{y(n)} - y(n) = \sum_{i=1}^{L} c_i.e'_i(n) + \delta(n) \qquad (3.30)$$

ce qui conduit après transformation en z à une équation de la forme :

$$e(z) = \sum_{i=1}^{L} c_i.e'_i(z) + \delta(z) \qquad (3.31)$$

soit en remplaçant les $e_i(z)$ par leur expression précédemment calculée :

$$e(z) = \sum_{i=1}^{L} c_i.\left[\sum_{j=1}^{L} H_{ji}(z) \cdot \delta_j(z) \right] + \delta(z) \qquad (3.32)$$

que l'on peut encore écrire sous la forme :

$$e(z) = \sum_{i=1}^{L} T_i(z) \cdot \delta_i(z) + \delta(z) \qquad (3.33)$$

Si l'on cherche maintenant à calculer la variance du bruit en sortie de la structure on voit qu'il est nécessaire de calculer la variance des différentes variables $\delta_k(n)$. Nous avons :

$$\delta_k(n) = e'''(n) + \sum_{i=1}^{B-1} e''_i(n) \cdot 2^{-i} + e''_0(n) \qquad (3.34)$$

si l'on admet que e" et e"' sont des bruits centrés de variance $q^2/12$ et indépendants, on a (le problème de l'indépendance de ces variables est traité dans [47]) la variance de ce signal est donnée par la relation suivante :

$$\sigma^2_{\delta_k} = \Delta_k^2 = \frac{q^2}{12} + \sum_{j=1}^{B-1} \frac{q^2}{12} \cdot 2^{-2j} + \frac{q^2}{12} \qquad (3.35)$$

soit encore :

$$\Delta_k^2 = \frac{q^2}{12} \cdot \left[1 + \sum_{J=0}^{B-1} 2^{-2j} \right] \qquad (3.36)$$

ce qui conduit à une expression de la forme :

$$\Delta_k^2 \simeq \frac{7}{3} \cdot \frac{q^2}{12} = \Delta^2 \qquad \text{si} \quad B \gg 8 \tag{3.37}$$

qui est indépendante de k. Si l'on suppose l'indépendance des variables δ_k, la variance de l'erreur en sortie du filtre est donc donnée par la relation :

$$\sigma_e^2 = \sum_{i=1}^{L} \| T_i(z) \|^2 \cdot \Delta^2 + \Delta^2 \qquad \text{avec} \quad \Delta^2 = \frac{7}{3} \cdot \frac{q^2}{12} \tag{3.38}$$

Il est également possible de mettre la fonction erreur sous la forme d'un produit matriciel en utilisant les matrices d'état du système :

$$e(z) = c^T.(z.I - A)^{-1} . R + \delta(z)$$

$$R = \begin{bmatrix} \delta_1(z) \\ \delta_2(z) \\ \vdots \\ \delta_L(z) \end{bmatrix} \tag{3.39}$$

Ces résultats peuvent être appliqués aux cas particuliers des trois représentations dans l'espace d'état que nous avons étudiées dans le chapitre précédent. Afin de vérifier les calculs théoriques et surtout la validité des hypothèses qui permettent de résoudre les équations, nous avons comparé ces résultats théoriques aux résultats obtenus par simulation sur ordinateur des structures FAD correspondantes. La valeur simulée de la puissance de bruit est calculée par différence entre le résultat fourni par une structure qui travaille sur un nombre de bits pré-défini et une structure théorique qui travaille en virgule flottante avec la précision de l'ordinateur. Les signaux d'entrée sont les mêmes que ceux utilisés pour les simulations des filtres de structure classique.

III.6.2. Cas de la représentation FTDIC.

Pour ce type de représentation, correspondant à une réalisation de filtre classique représentée en figure II.7 (b) on a l'ensemble des matrices d'état données par :

$$A = \begin{bmatrix} k_2 & 0 \\ k_2 & k_1 \end{bmatrix} \qquad B = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad C = \begin{bmatrix} k_2 & k_1 \end{bmatrix} \qquad D = 1 \tag{3.40}$$

La variance de l'erreur, c'est à dire la puissance de l'erreur en sortie de la structure FAD correspondante est alors donnée par la relation :

$$\sigma_e^2 = \left[\frac{k_1^2}{1 - k_1^2} + \frac{k_2^2 . (1 + k_1 . k_2)}{(1 - k_1^2) . (1 - k_2^2) . (1 - k_1 . k_2)} \right] . \Delta^2 + \Delta^2 \tag{3.41}$$

avec la relation suivante qui lie les coefficients K_1 et K_2 aux pôles L_1 et L_2 :

$$\begin{cases} k_1 = L_1 \\ k_2 = L_2 \end{cases} \tag{3.42}$$

La figure III.8 donne une comparaison entre les résultats théoriques obtenus au moyen de la formule précédente et les résultats simulés, ceci pour une valeur du pôle L_2 égale à 0.4 et une taille des mots de 16 bits.

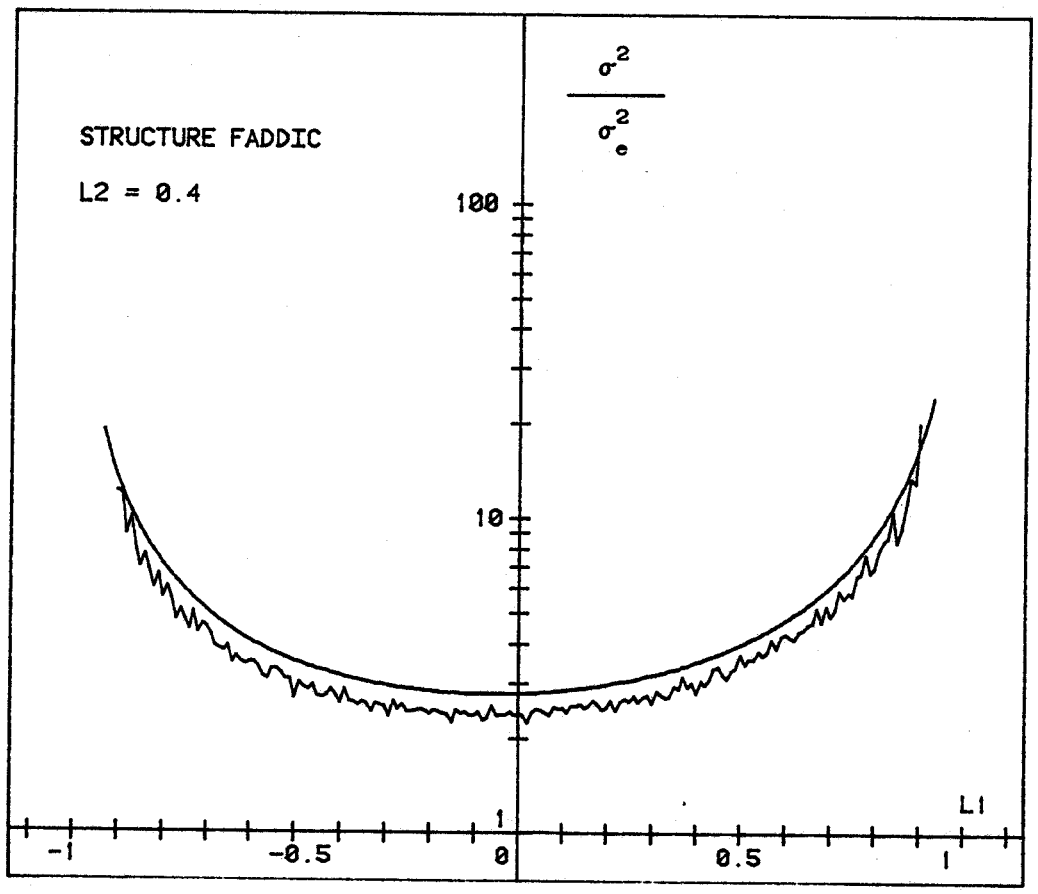


Figure III.8 - *Puissance réduite de bruit en fonction du pôle L_1 pour n = 16 et L_2 = 0.4. Valeur théorique et simulée. Cas de la représentation FADDIC.*

III.6.3. Cas de la représentation FTDIIC.

De façon analogue ce type de représentation, correspondant à une réalisation de filtre classique représentée en figure II.7 (a) fournit un ensemble des matrices d'état données par :

$$A = \begin{bmatrix} 0 & 1 \\ k_2 & k_1 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad C = \begin{bmatrix} k_2 & k_1 \end{bmatrix} \qquad D = 1 \quad (3.43)$$

La variance de l'erreur en sortie de la structure FAD correspondante est donnée par la relation suivante :

$$\sigma_e^2 = \left[\frac{k_1^2 + 2.k_1.k_2.(L_1 + L_2) + k_2^2 + (k_1^2 + k_2^2).L_1.L_2}{(1 - L_1^2).(1 - L_2^2).(1 - L_1.L_2)} \right] . \Delta^2 + \Delta^2 \qquad (3.44)$$

avec une relation liant les coefficients aux pôles de la fonction de transfert qui est la suivante :

$$\begin{cases} k_1 = L_1 + L_2 \\ k_2 = -L_2.L_1 \end{cases} \qquad (3.45)$$

La figure III.9 fournit une comparaison entre les résultats simulés et calculés dans les mêmes conditions que précédemment.

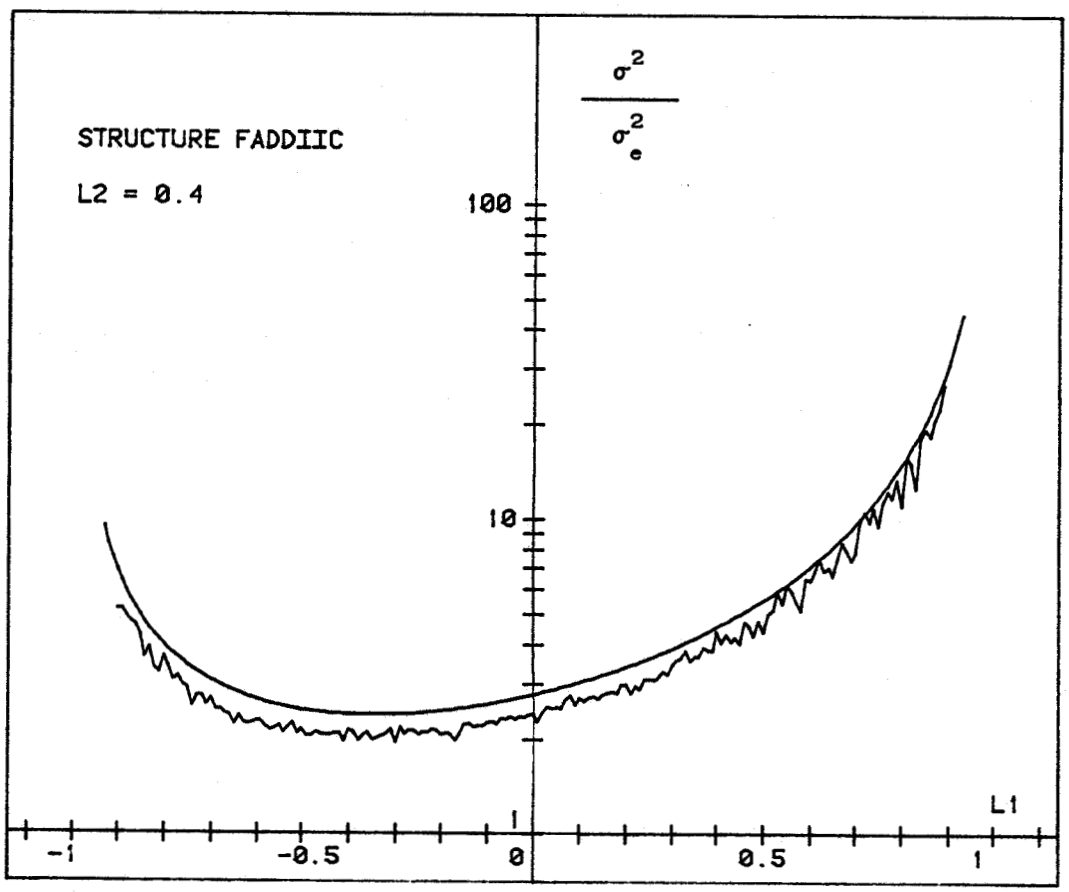


Figure III.9 - *Puissance réduite de bruit en fonction du pôle L_1 pour n = 16 et L_2 = 0.4. Valeur théorique et simulée. Cas de la représentation FADDIIC.*

III.6.4. Cas de la représentation FTTC.

De façon analogue ce type de représentation, correspondant à une réalisation de filtre classique représentée en figure II.7 (c) fournit un ensemble des matrices d'état données par :

$$A = \begin{bmatrix} -k_1 k_2 & 1-k_1^2 \\ k_2 & k_1 \end{bmatrix} \quad B = \begin{bmatrix} -k_1 \\ 1 \end{bmatrix} \quad C = \begin{bmatrix} k_2 & k_1 \end{bmatrix} \quad D = 1 \qquad (3.46)$$

La variance de l'erreur en sortie de la structure FAD correspondante est donnée par la relation suivante :

$$\sigma_e^2 = \left[\frac{2.k_2^2 + k_1^2 + 2.k_1.k_2.(L_1 + L_2) + (k_1^2 + k_2^2).L_1.L_2}{(1 - L_1^2).(1 - L_2^2).(1 - L_1.L_2)} \right] . \Delta^2 + \Delta^2 \qquad (3.47)$$

avec une relation liant les coefficients aux pôles de la fonction de transfert qui est la suivante :

$$\begin{cases} k_1 = \dfrac{L_1 + L_2}{1 + L_1.L_2} \\ k_2 = -L_2.L_1 \end{cases} \qquad (3.48)$$

La figure III.10 fournit une comparaison entre les résultats simulés et calculés dans les mêmes conditions que précédemment.

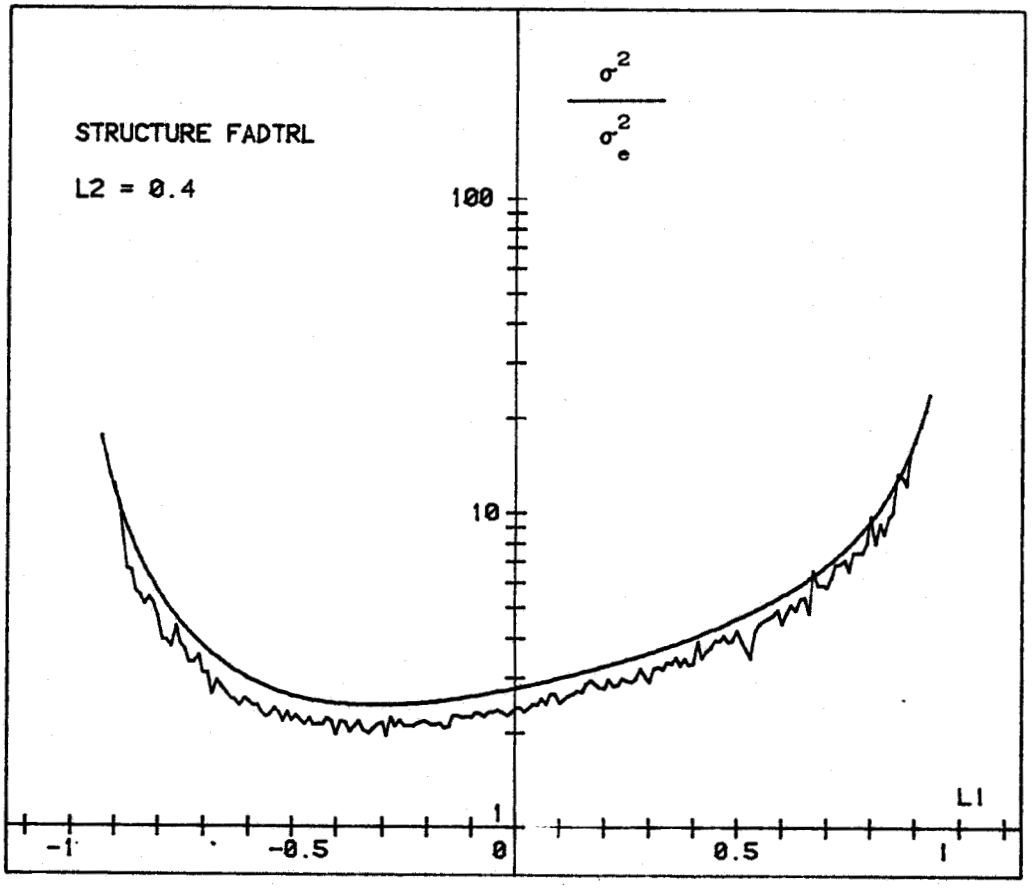


Figure III.10 - *Puissance réduite de bruit en fonction du pôle L_1 pour n = 16 et L_2 = 0.4. Valeur théorique et simulée. Cas de la représentation FADTC.*

III.7. REMARQUES ET CONCLUSION.

Les résultats précédents montrent que les valeurs des puissances réduites de bruit pour les trois structures étudiées, sont du même ordre de grandeur que celles obtenues en sortie des filtres réalisés au moyen d'une arithmétique localisée (confère figures II.9, II.10 et II.11) avec toutefois une atténuation des divergences qui existaient en certains points entre la théorie et la simulation dans le cas des filtres classiques. Ceci se comprend bien puisque ces problèmes étaient liés au passage par certaines valeurs, des coefficients des multiplieurs localisés.

Quant au problème de l'optimisation de telles structures, il conduit à deux remarques principales.

D'abord, compte tenu du fait que l'expression théorique (3.39) de l'erreur en sortie des structures FAD proposées s'exprime, comme dans le cas des filtres

classiques, en fonction des matrices d'état du système, il apparait qu'une technique d'optimisation basée sur le calcul d'une transformation T non singulière qui laisse invariante H(z), doit permettre de trouver la représentation d'état à bruit minimal. Il existe cependant une différence fondamentale avec le cas des filtres classiques. Pour ces derniers, la transformation T conduit à une nouvelle représentation d'état (A', B', C', D') qui en général complique passablement la structure à réalisée comme cela à été vu au paragraphe III.2.2. Dans le cas des structures FAD proposées, le passage à la structure optimale en terme de bruit se fait uniquement en changeant le contenu des tables mémoires générant les fonctions phi. La structure optimale n'est donc pas plus compliquée que la structure d'origine. Il est donc évident que dans le cas d'une réalisation, il convient d'adopter directement la solution d'une structure optimale en termes de bruit.

CONCLUSION

CONCLUSION.

A partir de l'étude du principe de la quantification et de ses différentes techniques de mise en oeuvre, nous nous sommes intéressé à caractériser le signal erreur résultant de l'introduction d'un quantificateur dans une chaîne de traitements numériques. Ceci nous a permis de déterminer un modèle de la source de bruit qui accompagne les multiplieurs numériques classiques ainsi que des nuances qu'il convient de prendre en compte dans le cas particulier où les coefficients multiplicatifs prennent des valeurs discrètes. Nous avons donc ensuite énuméré un certain nombre de procédés de calcul de l'erreur d'arrondi en sortie d'un filtre en développant plus spécialement la méthode qui fait appel à la représentation des filtres dans l'espace d'état. En effet ce type de représentation est bien adapté au problème de la recherche d'une structure à bruit minimal. Les valeurs théoriques obtenues ont été comparées avec des simulations pour plusieurs types de filtres. Elles fournissent des résultats tout à fait satisfaisants.

Le problème de l'optimisation en termes de bruit, des filtres à arithmétique classique, conduit cependant à constater que la réduction de la puissance de bruit en sortie d'un filtre, n'est obtenue qu'au prix d'une complexité accrue de la structure. Il nous est alors apparu intéressant de présenter la technique de l'arithmétique distribuée qui peut constituer une alternative permettant de lever la contradiction énoncée précédemment. Après avoir rappelé deux techniques de représentation possible des filtres à l'aide de cette arithmétique particulière, nous avons proposé une nouvelle structure qui permet au moyen de l'arithmétique distribuée de représenter n'importe quel filtre donné par ses équations d'état et ceci quelque soit la forme des matrices d'état. Nous avons donné une expression théorique de la puissance de bruit en sortie de cette structure et les valeurs théoriques obtenues ont été comme précédemment, confirmées par des simulations sur ordinateur.Nous avons pu constater que ce nouveau type de structures n'est pas plus bruyant que les structures classiques correspondantes. De plus, pour cette structure, des techniques d'optimisation analogues à celles utilisées dans le cas de filtres réalisés à l'aide d'une arithmétique localisée, doivent permettre d'arriver à des filtres dont le bruit en sortie est minimal. Il s'agit donc de trouver une transformation T non singulière qui laisse invariante la fonction de transfert du filtre mais qui conduit à l'obtention d'une représentation dans l'espace d'état du filtre optimal en terme de bruit. Il reste naturellement à préciser un algorithme qui permettra le calcul systématique de la transformation T comme dans le cas de l'optimisation des filtres classiques. Le passage de la structure initiale à la structure optimale en termes de bruit se faisant uniquement en changeant le contenu des tables mémoires, il n'y a donc pas complication de la structure lors du processus d'optimisation comme c'était le cas pour les filtres à arithmétique localisée.

Par ailleurs, on peut penser que compte tenu des performances technologiques actuelles, le problème des temps d'accès des mémoires n'est plus vraiment comme dans le passé, un obstacle à l'utilisation de la technique de l'arithmétique distribuée tout au moins dans le cas d'applications basses et moyennes fréquences (jusqu'à quelques dizaines de Mhz pour des tailles de bus de 16 bits par exemples). Cette technique semble par contre peu appropriée dans les cas de filtrage adaptatif ceci en raison des temps de réactualisation des tables mémoires qui peuvent devenir prépondérants vis à vis des durées propres de calculs. Cela serait toutefois très intéressant dans la mesure où la technique de l'arithmétique distribuée doit permettre de diminuer considérablement l'effet de quantification des coefficients.

ANNEXES

ANNEXE 1

ANNEXE 1.

Calcul de la puissance de bruit de saturation dans le cas d'un quantificateur par arrondi de paramètres (n, A_m) au quel on applique un signal x de densité de probabilité gaussienne, de moyenne nulle et de variance σ^2.

Compte tenu de la définition de x, nous avons la fonction densité de probabilité de ce signal qui est donnée par la relation :

$$p(x) = \frac{1}{\sigma.\sqrt{2.\pi}} e^{\frac{-x^2}{2.\sigma^2}}$$

Si A_m représente l'amplitude maximale du quantificateur, le bruit s de saturation est donné par les relations:

$$s = A_m - x \quad \text{si } x \geq A_m \quad \text{soit } s \in]-\infty, 0]$$

$$s = -A_m - x \quad \text{si } x \leq -A_m \quad \text{soit } s \in [0, +\infty[$$

Calculons la puissance de s si x est supérieur à A_m :

$$\sigma^2_{s+} = \int_{-\infty}^{0} s^2.g(s)\, ds$$

Il faut donc déterminer g(s), la densité de probabilité du bruit de saturation. Or s se présente comme une fonction de la variable aléatoire x de densité de probabilité p(x). Comme cette fonction est de plus monotone on sait que :

$$g(s) = p\left[f(s) \right].\left|\frac{dx}{ds}\right|$$

Soit donc ici :

$$g(s) = \frac{1}{\sigma.\sqrt{2.\pi}}\, e^{\frac{-(A_m - s)^2}{2.\sigma^2}}$$

et

$$\sigma^2_{s+} = \frac{1}{\sigma.\sqrt{2.\pi}} \int_{-\infty}^{0} s^2 . e^{\frac{-(A_m - s)^2}{2.\sigma^2}}\, ds$$

Soit encore en posant: $y = A_m - s$,

$$\sigma^2_{s+} = \frac{1}{\sigma.\sqrt{2.\pi}} \int_{A_m}^{+\infty} (A_m - y)^2 . e^{\frac{-y^2}{2.\sigma^2}}\, dy$$

que l'on décompose en :

$$\sigma^2_{s+} = \frac{1}{\sigma.\sqrt{2.\pi}} \int_{A_m}^{+\infty} (A_m^2 - 2.y.A_m + y^2). e^{\frac{-y^2}{2.\sigma^2}}\, dy$$

$$\sigma^2_{s+} = \frac{1}{\sigma.\sqrt{2.\pi}} (I + J + K)$$

Après calculs, nous trouvons en définissant la fonction erfc(x) par :

$$\mathrm{erfc}(x) = \frac{2}{\sqrt{\pi}} . \int_{x}^{+\infty} e^{-t^2}\, dt$$

$$I = \int_{A_m}^{+\infty} A_m^2 \cdot e^{\frac{-y^2}{2.\sigma^2}} dy = \frac{\sigma . A_m^2 . \sqrt{\pi}}{\sqrt{2}} . \operatorname{erfc}\left[\frac{A_m}{\sigma . \sqrt{2}} \right]$$

$$J = \int_{A_m}^{+\infty} -2.y.A_m . e^{\frac{-y^2}{2.\sigma^2}} dy = -2.A_m.\sigma^2 . e^{\frac{-A_m^2}{2.\sigma^2}}$$

et à l'aide d'une intégration par partie :

$$K = A_m.\sigma^2 . e^{\frac{-A_m^2}{2.\sigma^2}} + \frac{\sigma.\sqrt{\pi}}{\sqrt{2}} . \operatorname{erfc}\left[\frac{A_m}{\sigma.\sqrt{2}} \right]$$

Un calcul analogue pour $x \leq -A_m$ conduit à :

$$\sigma_{s+}^2 = \sigma_{s-}^2$$

D'où la formule donnant la puissance de bruit de saturation σ_s^2 générée par un quantificateur par arrondi de valeur maximale A_m, traversé par un signal de densité de probabilité gaussienne, de moyenne nulle et de variance σ^2 :

$$\sigma_s^2 = (\sigma^2 + A_m^2) . \operatorname{erfc}\left[\frac{A_m}{\sigma.\sqrt{2}} \right] - \frac{\sigma . A_m . \sqrt{2}}{\sqrt{\pi}} . e^{\frac{-A_m^2}{2.\sigma^2}}$$

ANNEXE 2

ANNEXE 2.

Tableau des niveaux de décision et de reconstruction des quantificateurs scalaires optimaux pour les densités uniformes, de Gauss, de Laplace et de Rayleigh. D'après [15].

	Uniform		Gaussian		Laplacian		Rayleigh	
Bits	x[i]	y[i]	x[i]	y[i]	x[i]	y[i]	x[i]	y[i]
1	−1.0000	−0.5000	−∞	−0.7979	−∞	−0.7071	0.0000	1.2657
	0.0000	0.5000	0.0000	0.7979	0.0000	0.7071	2.0985	2.9313
	1.0000		∞		∞		∞	
2	−1.0000	−0.7500	−∞	−1.5104	−∞	−1.8340	0.0000	0.8079
	−0.5000	−0.2500	−0.9816	−0.4528	−1.1269	−0.4198	1.2545	1.7010
	−0.0000	0.2500	0.0000	0.4528	0.0000	0.4198	2.1667	2.6325
	0.5000	0.7500	0.9816	1.5104	1.1269	1.8340	3.2465	3.8604
	1.0000		∞		∞		∞	
3	−1.0000	−0.8750	−∞	−2.1519	−∞	−3.0867	0.0000	0.5016
	−0.7500	−0.6250	−1.7479	−1.3439	−2.3796	−1.6725	0.7619	1.0222
	−0.5000	−0.3750	−1.0500	−0.7560	−1.2527	−0.8330	1.2594	1.4966
	−0.2500	−0.1250	−0.5005	−0.2451	−0.5332	−0.2334	1.7327	1.9688
	0.0000	0.1250	0.0000	0.2451	0.0000	0.2334	2.2182	2.4675
	0.2500	0.3750	0,5005	0.7560	0.5332	0.8330	2.7476	3.0277
	0.5000	0.6250	1.0500	1.3439	1.2527	1.6725	3.3707	3.7137
	0.7500	0.8750	1.7479	2.1519	2.3796	3.0867	4.2124	4.7111
	1.0000		∞		∞		∞	
4	−1.0000	−0.9375	−∞	−2.7326	−∞	−4.4311	0.0000	0.3057
	−0.8750	−0.8125	−2.4008	−2.0690	−3.7240	−3.0169	0.4606	0.6156
	−0.7500	−0.6875	−1.8435	−1.6180	−2.5971	−2.1773	0.7509	0.8863
	−0.6250	−0.5625	−1.4371	−1.2562	−1.8776	−1.5778	1.0130	1.1397
	−0.5000	−0.4375	−1.0993	−0.9423	−1.3444	−1.1110	1.2624	1.3850
	−0.3750	−0.3125	−0.7995	−0.6568	−0.9198	−0.7287	1.5064	1.6277
	−0.2500	−0.1875	−0.5224	−0.3880	−0.5667	−0.4048	1.7499	1.8721
	−0.1250	−0.0625	−0.2582	−0.1284	−0.2664	−0.1240	1.9970	2.1220
	0.0000	0.0625	0.0000	0.1284	0.0000	0.1240	2.2517	2.3814
	0.1250	0.1875	0.2582	0.3880	0.2644	0.4048	2.5182	2.6550
	0.2500	0.3125	0.5224	0.6568	0.5667	0.7287	2.8021	2.9492
	0.3750	0.4375	0.7995	0.9423	0.9198	1.1110	3.1110	3.2729
	0.5000	0.5625	1.0993	1.2562	1.3444	1.5778	3.4566	3.6403
	0.6250	0.6875	1.4371	1.6180	1.8776	2.1773	3.8588	4.0772
	0.7500	0.8125	1.8435	2.0690	2.5971	3.0169	4.3579	4.6385
	0.8750	0.9375	2.4008	2.7326	3.7240	4.4311	5.0649	5.4913
	1.0000		∞		∞		∞	

ANNEXE 3

ANNEXE 3.

Génération d'une variable aléatoire gaussienne discrète.

Les lois de répartition limites des variables aléatoires font partie du groupe de théorèmes appelé groupe du Théorème Central Limite. C'est Liapounof qui a montré que si $(X_1,X_2, \ldots , X_n)$ sont des variables aléatoires de même loi de répartition, d'espérance mathématique x et de variance σ^2, alors la loi de répartition Y définie par :

$$Y = \sum_{i=1}^{n} X_i$$

tend vers la loi normale quand n tend vers l'infini.

Nous avons utilisé ici pour $(X_1,X_2, \ldots , X_n)$ un ensemble de variables aléatoires uniformément réparties sur [0,1] générées de la façon suivante :

sur un calculateur binaire si on génère une séquence (i) à partir de la relation :

$$i_{n+1} = k.i_n \text{ modulo } 2^p$$

où i_n et k sont des entiers et p le nombre de bits (bit de signe exclu), on sait [38] que la périodicité maximale de la séquence est atteinte si:

i_0 est impair

et k de la forme $k = 8.m \pm 3$

et qu'elle vaut dans ce cas $2^{(p-2)}$.

Pour améliorer le caractère aléatoire de la séquence, il convient de prendre une valeur de k ni trop forte ni trop faible. Une bonne valeur est donnée par :

$$k = 2^{p/2} + 3$$

Pour obtenir une séquence de nombres dans [0,1] il suffit alors de diviser chaque terme de la séquence par 2^p.

Le calculateur VME travaillant sur des entiers définis dans une représentation en complément à deux sur 32 bits, pour obtenir une séquence de nombres entiers positifs il est nécessaire de vérifier la relation :

$$2^{p/2} \cdot 2^{p} \leq 2^{31} - 1$$

Cette relation conduit à $p < 20.6$ soit donc $p = 20$ ce qui assure une périodicité pour la variable uniformément répartie de $2^{18} = 262\,144$.

Nous avons donc l'algorithme de génération de la variable uniformément répartie qui est le suivant :

$$i_0 \text{ impair}$$

$$i_{n+1} = 1027.i_n \text{ modulo } 2^{20}$$

$$X_n = 2^{-20}.i_n$$

Il reste alors a centrér cette variable en retranchant ½ à chaque valeur de la séquence. On obtient donc une variable aléatoire discrète uniformément répartie dans l'intervalle $[-\frac{1}{2}, +\frac{1}{2}]$

Une fois cet ensemble de variables uniformément réparties généré, on peut en les ajoutant obtenir une variable Y à peu près gaussienne suivant la formule :

$$Y_n = \sum_{k=1}^{12} X_{12n-k+1} \qquad -6$$

Dans ce cas la variance du signal Y sera donné par

$$\sigma_Y^2 = \sum_{i=1}^{12} E\,[\,X_i\,] = \frac{12}{12} = 1$$

ANNEXE 4

ANNEXE 4.

Calcul de la loi de probabilité d'un signal de puissance donnée qui possède une entropie maximale [48].

Soit un signal x(t), sa puissance est définie par la relation :

$$P = \int x^2 . p(x)\, dx$$

avec p(x) la probabilité d'avoir un signal d' amplitude x. On a naturellement la relation de fermeture de la loi qui se traduit par la formule :

$$\int p(x)\, dx = 1$$

Il convient donc de chercher la forme de p(x) telle que l'entropie H soit maximale pour P fixée. C'est à dire que :

$$H = \int p(x).\ \log\left[\frac{1}{p(x)}\right] dx = -\int p(x).\log\left[p(x)\right] dx$$

soit maximale.

Pour être capable de trouver une solution non triviale à cette équation on forme la grandeur J définie par :

$$J = H + \vartheta . \int p(x)\, dx + \mu . P = H + cst.$$

en introduisant deux facteurs de Lagrange μ et Θ que l'on déterminera et qui évoluent comme H. On peut alors écrire :

$$J = \int p(x) \cdot \left[-\log\left(p(x) \right) + \vartheta + \mu x^2 \right] dx$$

$$\frac{dJ}{dp} = \int \left[-\log\left(p(x) \right) - 1 + \vartheta + \mu x^2 \right] dx$$

$$\frac{dJ}{dp} = 0 \qquad \text{pour} \qquad -\log\left(p(x) \right) - 1 + \vartheta + \mu x^2 = 0$$

ou pour

$$p(x) = e^{(\vartheta - 1)} . e^{(\mu x^2)}$$

Si on reporte cette expression donnant p(x) dans celle donnant la puissance du signal x tout en affirmant que le signal x existe, on obtient le système d'équations suivants :

$$\begin{cases} \displaystyle\int_{-\infty}^{+\infty} e^{(\vartheta-1)} . e^{\mu x^2} \, dx = 1 \\ \displaystyle\int_{-\infty}^{+\infty} x^2 . e^{(\vartheta-1)} . e^{\mu x^2} \, dx = P \end{cases}$$

or on sait que :

$$\int_{-\infty}^{+\infty} e^{-x^2}\,dx = \sqrt{\pi} \qquad \text{et} \qquad \int_{-\infty}^{+\infty} y^2\, e^{-y^2}\,dy = \frac{\sqrt{\pi}}{2}$$

d'où

$$\int_{-\infty}^{+\infty} e^{-\mu x^2}\,dx = \frac{\sqrt{\pi}}{\sqrt{-\mu}} \qquad \text{et} \qquad e^{(\vartheta-1)} \int_{-\infty}^{+\infty} e^{-\mu y^2}\,dy = 1$$

$$\frac{\sqrt{\pi} \cdot e^{(\vartheta-1)}}{\sqrt{-\mu}} = 1$$

et

$$\int_{-\infty}^{+\infty} e^{(\vartheta^2-1)}\, x^2\, e^{\mu x^2} dx = e^{(\vartheta-1)} \frac{1}{-\mu} \cdot \frac{1}{\sqrt{-\mu}} \int_{-\infty}^{+\infty} x^2\, e^{-x^2}\,dx = P$$

d'où

$$e^{(\vartheta-1)} \frac{1}{-\mu} \cdot \frac{1}{\sqrt{-\mu}} \cdot \frac{\sqrt{\pi}}{2} = P$$

Ces deux relations permettent d'obtenir :

$$\mu = \frac{-1}{2.P} \qquad \text{et} \qquad e^{(\vartheta-1)} \frac{1}{\sqrt{P}} \cdot \frac{1}{\sqrt{2\pi}} = 1$$

et si l'on pose : $P = \sigma^2$ on obtient en reportant $e^{(\Theta-1)}$ et μ dans p(x) :

$$p(x) = \frac{1}{\sigma.\sqrt{2\pi}} \; e^{\frac{-x^2}{2.\sigma^2}}$$

Cette relation montre que pour une puissance donnée la quantité d'information contenue dans un signal est maximale quand celui-ci est de type gaussien. Dans ce cas on montre que cette quantité d'information vaut :

$$H = \log \sqrt{2\pi.e.P}$$

ANNEXE 5

ANNEXE 5.

Tableau de comparaisons entre différentes organisations possibles d'une structure direct I à arithmétique distribuée. D'après [49].

p est le nombre de bits utilisés pour adresser simultanément une mémoire.

n est le nombre de groupes contenus dans chaque mot.

Dans l'exemple présent, p.n est pris toujours égale à 12.

	Configuration No.	N	P	Pkg. Count	Speed (word rate) MHz	Cost $	F_1 time - cost to compute one word	Ranking F_1
A	1	12	1	21	2.56	43.40	16.92	1
	2	6	2	22	2.86	68.04	23.81	3
	3	4	3	22	3.08	80.49	26.16	6
	4	3	4	44	3.03	260.00	85.80	15
	5	2	6	603	4.545	13.183.00	2900.00	17
	6	1	12	154×10^3	6.58	3440×10^3	522.8×10^3	19
B	7	6	2	33	2.86	120.93	36.03	12
	8	4	3	33	4	120.93	30.23	9
	9	3	4	33	5	120.93	24.19	4
	10	2	6	36	4.76	140.88	29.58	8
	11	1	12	102	6.67	716.73	107.51	16
C	12	6	2	27	2.38	73.16	30.73	10
	13	4	3	28	4.0	97.81	24.45	5
	14	3	4	28	5.0	97.81	19.56	2
	15	2	6	48	3.97	283.36	71.41	14
	16	1	12	38×10^3	8.43	840×10^3	1.008×10^6	18
	17			43	8.0	226.74	28.34	7
	18			87	7.69	388.50	50.51	13
	19			57	7.69	249.78	32.47	11

	Configuration No.	N	P	Speed Word rate (MHz)	Chip Area (mm^2)	F_2 Time - area	Rankings (F_2)	(F_1)
A	1	12	1	2.56	.84	.3276	1	1
	2	6	2	2.86	1.3125	.4594	4	3
	3	4	3	3.08	5.0925	1.655	12	6
	4	3	4	3.03	35.3325	11.66	16	15
	5	2	6	4.545	2213	486	17	17
	6	1	12	6.58	579×10^3	88×10^3	19	19
B	7	6	2	2.86	1.8785	.6575	7	12
	8	4	3	4.	1.98	.4950	6	10
	9	3	4	5.	2.1825	.4365	3	4
	10	2	6	4.76	3.3975	.7135	8	8
	11	1	12	6.67	36.3375	5.45	14	16
C	12	6	2	2.38	1.4438	.4043	2	10
	13	4	3	4.	1.8825	.4706	5	5
	14	3	4	5.	3.9525	.7905	9	2
	15	2	6	3.97	36.375	9.1665	15	14
	16	1	12	8.43	141×10^1	16.17×10^1	18	18
	17			8.	19.425	2.4281	13	7
	18			7.69	6.435	.8366	11	13
	19			7.69	6.255	.8132	10	11

A: 1 group of 3 registers
B: 3 groups of 1 register each
C: 2 groups containing 1 and 2 registers, respectively

BIBLIOGRAPHIE

BIBLIOGRAPHIE

[1] - C.T. MULLIS, R.A. ROBERTS - "Synthesis of minimum roundoff noise fixed point digital filters." IEEE Transac. on Circuit and System vol. CAS-23, pp 551-562, Septembre 1976.

[2] - C.W. BARNES - "Roundoff noise and overflow in normal digital filters." IEEE Transac. on Circuit and System vol. CAS-26, pp 154-159, Mars 1978.

[3] - D. WILLIAMSON - "Roundoff noise minimization and pole-zero sensitivity in fixed point digital filters using residue feedback." IEEE Transac. on Acoustics Speech and Signal Processing vol. ASSP-34, pp Octobre 1986.

[4] - D.V.K. RAO - "Analysis of coefficient quantisation errors in state space digital filters." IEEE Transac. on Acoustics Speech and Signal Processing vol. ASSP-34 Février 1986.

[5] - W.L. MILLS, C.T. MULLIS, R.A. ROBERTS - "Digital filters realization without overflow oscillations." IEEE Transac. on Acoustics Speech and Signal Processing vol. ASSP-26, pp 334-338, Août 1978.

[6] - L.B. JACKSON - "Limit cycles in state space structures for digital filters."IEEE Transac. on Circuit and System vol. CAS-26, pp 67-68, Janvier 1979.

[7] - P.P. VAIDYANATHAN, V. LIU - "An impoved sufficient condition for absence of limit cycles in digital filters."IEEE Transac. on Circuit and System vol. CAS-34, Mars 1987.

[8] - C.W. BARNES - "A parametric approach to the realization of second order digital filter sections." IEEE Transac. on Circuit and System vol. CAS-32, Juin 1985.

[9] - W.E. HIGGINS, D.C. MUNSON - "Noise reductionstrategie for digital filters: Error Spectrum Shaping versus the optimal linear state space formulation." IEEE Transac. on Acoustics Speech and Signal Processing vol. ASSP-30, Decembre 1982.

[10] - A. GERSHO - "Principles of quantization." IEEE Transac. on Circuit and System vol. CAS-25, pp 427-436, Juillet 1978.

[11] - B.GOLD, L.R. RABINER - "Theory and application of digital signal processing." Prentice Hall, Inc. Englewood Cliffs, New Jersey.

[12] - M. BELLANGER - "Traitement numérique du signal, théorie et pratique." 3e édition, Masson 1987.

[13] - W.R. BENNETT - "Spectrum of quantized signals." Bell Syst. Tech. J., vol. 27, pp 446-472, Juillet 1948.

[14] - CCITT - Réseaux numériques systèmes de transmission et équipement de multiplexage. Vol.III.3 . Genève 1981.

[15] - W.K. PRATT - "Digital image processing." Edition J. WILEY and SONS, 1978.

[16] - P.F. PANTER et W. DITE - "Quantizing distortion in pulse-count modulation with non uniform spacing of levels." Proced. IRE, vol. 39, pp 44-48, 1951.

[17] - B. SMITH - "Instantaneous companding of quantized signals." Bell Syst. Tech. J., vol. 27, pp 446-478, 1948.

[18] - S.P. LLOYD - "Least squares quantization in PCM." IEEE Transac. on Information theory, vol. IT-28 N° 2, pp 129-137, Mars 1982.

[19] - P. FLEITCHER - "Sufficient conditions for achieving minimum distortion in quantizer." IEEE Int. Conv. Rec., pp 104-111, 1964.

[20] - J. MAX - "Quantizing for minimum distortion." IRE Transac. on inform. Theory, vol. IT-6, mars 1960.

[21] - M.D. PAEZ et T.H. GLISSON - "Minimum mean squared-error quantization in speech PCM and DPCM systems." IEEE Transac. on Commun., vol. COM-20, pp 225-230, Avril 1972.

[22] - J. GUICHARD et D. NASSE - "Image numérique et codage." L'échos des recherches, n°126, 4° trimestre 1986.

[23] - J.P. ADOUL - "La quantification vectorielle des formes d'ondes." Annales des Télécom. 41, n° 3-4, 1986.

[24] - A. GERSHO - "Structure of vector quantizer." Transac. on Inform. theory, vol. IT-28, mars 1982.

[25] - T.R. FISCHER et R.M. DICHARRY - "

[26] - C. LAMBLIN et J.P. ADOUL - "Algorithme de quantification vectorielle sphérique à partir du réseau de Gosset d'ordre 8." Annale des Télécom., 43, n° 3-4, 1988.

[27] - C.W. BARNES, B.N. TRAN et S.H. LEUNG - "On the statistics of fixed point arithmetic." IEEE Transac. on Acoustics Speech and Signal Processing vol. ASSP-33, pp. 595-606, Juin 1985.

[28] - S.R. PARKER et P.E. GIRARD - "Corraleted noise due to roundoff in fixed point digital filters." IEEE Transac. on Circuits and Systems vol. CAS-23, pp. 204-211, Avril 1976.

[29] - I. TOKAJI et C.W. BARNES - "Roundoff error statistics for a continuous range of multiplier coefficients." IEEE Transac. on Circuits and Systems vol. CAS-1, pp.52-59, Janvier 1987.

[30] - L.B. JACKSON - "On the interaction of roundoff noise and dynamic range in digital filters." Bell Syst. Tech. Journal vol-49, pp. 159-184, Février 1970.

[31] - E.J. JURY - "Theory and application of the z transform method." John wiley, 1964.

[32] - C.T. MULLIS et R.A. ROBERTS - "Roundoff noise in digital filters: frequency transformations and invariants." IEEE Transac. on Acoustics Speech and Signal Processing vol. ASSP-24, Decembre 1976.

[33] - F.J. TAYLOR et J.W. MARSCHALL - "Computer aided design and analysis of standard IIR architectures." IEEE Circuit and System magazine vol.3, Decembre 1981.

[34] - C.S. BERGER - "A numerical solution of the matrix equation $P = P^T + S'$." IEEE Transac. on Automat. Contr. vol. AC-16, Août 1971.

[35] - J.A. HEINEN - "A technique for solving the extended discrete Liapunov matrix equation." IEEE Transac. on Automat. Contr. vol. AC-17, Fevrier 1972.

[36] - S. BARNETT - "Simplification of the Liapunov matrix equation: $A^TPA - P = -Q'$." IEEE Transac. on Automat. Contr. vol. AC-19, Août 1974.

[37] - F. WAUQUIER - "Etude et réalisation d'un simulateur pour l'aide à la conception de filtres numériques." Thèse de $3^{ième}$ cycle, Université de Lille, 1988.

[38] - M. LABARRERE, J.P. KRIEF et B. GIMONET - "Le filtrage et ses applications." Collection Sup. Aero., Capadues Editions, 1978.

[39] - S. Y. HWANG - "Minimum uncorrelated unit noise in state space digitals filtering." IEEE Transac. on Acoustics Speech and Signal Processing, vol. ASSP-25, n° 4, Aout 1977.

[40] - V. TAVSANOGLU et L.THIELE - "Optimal design of state space digital filters by simultaneous minimization of sensitivity and roundoff noise." IEEE Transac. on Circuit and System, vol. CAS-31, n° 10, Octobre 1984.

[41] - B. B. BOMAR - "Minimum roundoff noise digital filter with some power of two coefficients. IEEE Transac. on Circuit and System, vol. CAS-31, n° 10, Octobre 1984.

[42] - J.P. GERARD -"Le bruit numérique." Examen général d'ingénieur CNAM, Mars 1987.

[43] - W. E. HIGGINS et D. C. MUNSON - "Noise reduction strategie for digital filters : Error Spectrum Shaping versus the optimal linear state space formulation." IEEE transac. on Acoustics Speech and Signal Processing, vol. ASSP-30, n°6, Décembre 1982.

[44] - A. PELED et B. LIU - "A new hardware realization of digital filters." IEEE Transac. on Acoustics Speech and Signal Processing, vol. ASSP-22, Décembre 1974.

[45] - C.S. BURRUS - "Digital filter structurs described by distributed arithmetic." IEEE Transac. on Circuit and System, vol. CAS-24, Décembre 1977.

[46] - F.J. TAYLOR - "An analysis of the distributed arithmetic digital filter." IEEE Transac. on Acoustics Speech and Signal Processing, vol. ASSP-34, n° 5, Octobre 1986.

[47] - K.D. KAMMEYER - "Quantization error of the distributed arithmetic." IEEE Transac. on Circuit and System, vol. CAS-24, pp. 674-680, Décembre 1977.

[48] - J.P. DUBUS - Cours d'électronique, traitement du signal. Université de Lille Flandres Artois.

[49] - M. ARJMAND et R.A. ROBERTS - "On comparing hardware implementations of fixed point digital filters." IEEE Circuit and System Magazine, vol. 3, 1981.

[50] - V. DEVLAMINCK, F. WAUQUIER et J.P. DUBUS - "Simulation de fonctionnement de filtres numériques récursifs pour le choix de la structure à bruit d'arrondi minimal." Revue Traitement du Signal, pp. 65-71 vol.5 n°2, 1988.

Résumé.

Il est présenté dans une première partie, un rappel des différentes techniques de quantification scalaires uniformes et non-uniformes ainsi que le principe de la quantification vectorielle. Ceci conduit à caractériser le bruit généré par un multiplieur numérique et permet d'introduire certaines remarques dans le cas particulier des coefficients discrets. Ces sources de bruit étant définies, après un rappel de la méthode classique de calcul du bruit d'arrondi dû aux multiplications dans un filtre numérique RII, il est présenté une méthode faisant appel à la représentation des filtres dans l'espace d'état. Les estimations théoriques du bruit sont vérifiées par comparaison avec des valeurs expérimentales obtenues à l'aide d'un simulateur, pour trois structures de filtres tous pôles réalisant une même fonction de transfert H(z). Les résultats sont exprimés en fonction des pôles de H(z) ce qui permet une détermination immédiate de la structure la moins bruyante. La troisième partie conduit, après un rappel des méthodes classiques d'optimisation des filtres en termes de bruit, à l'étude des réalisations de structures à l'aide de l'arithmétique distribuée. Il est alors proposé une structure permettant la réalisation de filtres à partir de n'importe quel type de représentation dans l'espace d'état. Une analyse du bruit d'arrondi généré dans la structure introduite conduit à l'obtention d'une valeur théorique de la puissance de ce bruit. Ces résultats théoriques sont vérifiés au moyen de simulations pour trois types de filtres du deuxième ordre. Les résultats montrent que cette structure n'est pas plus bruyante que celles réalisées à l'aide de l'arithmétique classique. L'avantage principal de cette nouvelle structure réside dans la possibilité d'arriver à une optimisation en termes de bruit sans complication de la structure initiale et ceci au moyen de technique d'optimisation analogues à celles utilisées pour les filtres à arithmétique classique.