

Titre: Techniques de décodage séquentiel appliquées au problème de l'interférence entre symboles
Title: l'interférence entre symboles

Auteurs: Philippe E. Haurie, & David Haccoun
Authors:

Date: 1982

Type: Rapport / Report

Référence: Haurie, P. E., & Haccoun, D. (1982). Techniques de décodage séquentiel appliquées au problème de l'interférence entre symboles. (Rapport technique n° EP-R-82-05). <https://publications.polymtl.ca/9515/>
Citation:

Document en libre accès dans PolyPublie

Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/9515/>
PolyPublie URL:

Version: Version officielle de l'éditeur / Published version

Conditions d'utilisation: Tous droits réservés / All rights reserved
Terms of Use:

Document publié chez l'éditeur officiel

Document issued by the official publisher

Institution: École Polytechnique de Montréal

Numéro de rapport: EP-R-82-05
Report number:

URL officiel:
Official URL:

Mention légale:
Legal notice:

25 FEV. 1982

TECHNIQUES DE DECODAGE SEQUENTIEL APPLIQUEES AU

PROBLEME DE L'INTERFERENCE ENTRE SYMBOLES*

Philippe Haurie
Recherches Bell Northern
Ile des Soeurs, (Montréal)

David Haccoun
Professeur titulaire
Département de Génie Electrique

(1982)

* Ce rapport constitue l'essentiel du mémoire de maîtrise en Génie Electrique (communication) de Philippe Haurie exécuté sous la direction de David Haccoun.

À CONSULTER
SUR PLACE

SOMMAIRE

Le but de cette recherche est d'étudier le comportement d'un décodeur séquentiel multi-chemins modifié pour la correction de l'interférence entre symboles qui se produit lors d'une transmission numérique à travers un canal à bande passante étroite. Des simulations sur ordinateur démontrent l'avantage qu'il y a à utiliser de tels décodeurs pour cette application plutôt que le décodeur de Viterbi; ce dernier, bien qu'ayant une performance optimale, requiert un effort de calcul qui croît exponentiellement avec la longueur de contrainte décrivant l'étendue de l'interférence entre symboles. Le décodeur de Viterbi est donc limité aux cas de faibles étendues d'interférence entre symboles. Ce n'est pas le cas des décodeurs séquentiels dont l'effort bien moindre de calcul est indépendant de la longueur de contrainte. Ceci les rend très attirants pour la correction de l'interférence entre symboles à grande étendue. Pour de faibles ou moyennes étendues d'interférence entre symboles, la performance, légèrement sous-optimale, d'un décodeur séquentiel multi-chemins est appréciable compte tenu de l'effort de calcul en moyenne très léger comparativement au décodeur de Viterbi. Cet avantage sérieux du décodeur séquentiel multi-chemins en fait une option de remplacement fort intéressante pour la correction de l'interférence entre symboles en général.

ABSTRACT

The purpose of this thesis is to assess the possibility of using a modified multiple-path sequential decoder for the correction of the intersymbol interference phenomenon which can occur in data transmissions through narrow band channels. Computer simulations show that these decoders may advantageously replace the optimal Viterbi decoder for this application, giving comparable results with far less computational effort. For channels yielding wide-spread intersymbol interference the use of the optimal Viterbi decoder is not recommended because of its exponentially increasing complexity with constraint length. This is not the case for sequential decoders thus making them an interesting alternative for the correction of intersymbol interference for these channels.

TABLE DES MATIERES

	<u>Page</u>
SOMMAIRE	IV
ABSTRACT	V
LISTE DES TABLEAUX	IX
LISTE DES FIGURES	X
LISTE DES COURBES	XI
CHAPITRE 1: INTRODUCTION	1
CHAPITRE 2: DESCRIPTION THEORIQUE DE L'INTERFERENCE ENTRE SYMBOLES	4
2.1 Définition générale du problème	4
2.2 Définition théorique	5
2.3 Modélisation du canal	14
CHAPITRE 3: CORRECTION DE L'INTERFERENCE ENTRE SYMBOLES	22
3.1 Recherches précédentes	22
3.2 Techniques non linéaires	25
3.3 Similitude avec le codage convolutionnel	27
3.4 Techniques de décodage optimales	32
3.5 Décodeur séquentiel	44
CHAPITRE 4: ANALYSE DES RESULTATS	51
4.1 Bornes de la probabilité d'erreur	51
4.2 Relation entre la distance minimum et la perte provoquée par un canal dispersif	62
4.3 Simulations	67
4.4 Rapport signal sur bruit	69
4.5 Générateur de nombres aléatoires	70
4.6 Structure du programme Fortran	72

	<u>Page</u>
4.7 Choix de la métrique	76
4.8 CANAL 0.19, 0.35, 0.46, 0.5, 0.46, 0.35, 0.19.	81
4.9 CANAL 0.33, 0.67, 1, 0.67, 0.33	93
4.10 CANAL 0.38, 0.6, 0.6, 0.38	100
4.11 CANAL 0.500, 1.000, 0.500	
CANAL 0.792, 0.165, 0.034	
CANAL 0.649, 0.228, 0.080	105
CHAPITRE 5: CONCLUSION	120
BIBLIOGRAPHIE	122

LISTE DES TABLEAUX

		<u>Page</u>
Tableau 1:	Canaux normalisés à distances minimisées	66
	Tableaux donnant les détails de calculs pour chaque canal simulé.	
Tableau 2:	Canal 0.19, 0.35, 0.46, 0.5, 0.46, 0.35, 0.19	92
Tableau 3:	Canal 0.33, 0.67, 1.0, 0.67, 0.33	99
Tableau 4:	Canal 0.38, 0.6, 0.6, 0.38	104
Tableau 5:	Canal 0.5, 1, 0.5	
Tableau 6:	Canaux 0.792, 0.165, 0.034, 0.649, 0.165, 0.649, 0.165, 0.034	117
Tableau 7:	Nombres moyens de calculs avec la métrique $1-\text{SNR } d^2$	119

LISTE DES FIGURES

<u>Figure</u>			<u>Page</u>
2.1	Transmission par canaux dispersifs	6	
2.2	Chaîne de transmission classique	7	
2.3	Inclusion d'un filtre adapté après le canal	15	
2.4	Filtre transversal représentant le canal	16	
2.5	Modélisation finale du canal	21	
3.1	Egaliseur transversal linéaire	24	
3.2 (a,b)	Codage convolutionnel	29	
3.3 (a,b)	Modélisation de l'interférence entre symbole	31	
3.4	Diagramme d'état correspondant à la figure 3.3(b)	34	
3.5	Représentation en arbre	36	
3.6	Représentation en treillis	38	
3.7	Algorithme Zigangirov-Jelinek pour un décodeur L-chemins	48	
4.1	Représentation des signaux dans un espace géométrique	53	
4.2	Schéma bloc de la simulation	68	

LISTE DES COURBESPROBABILITES D'ERREUR V.S. SNR(dB)

<u>Figure</u>	<u>Page</u>
---------------	-------------

Canal 0.19, 0.35, 0.46, 0.5, 0.46, 0.35, 0.19

4.3	Décodeur séquentiel 7-chemins, métrique = $10-\text{SNR}d^2$	84
4.4	Décodeur séquentiel 7-chemins, métrique = $1-\text{SNR}d^2$	85
4.5	Décodeur séquentiel 7-chemins, métrique entière = $10(1-d^2)$	86
4.6	Décodeur séquentiel 5-chemins, métrique = $10-\text{SNR}d^2$	87
4.7	Décodeur séquentiel 5-chemins, métrique = $1-\text{SNR}d^2$	88
4.8	Décodeur séquentiel 3-chemins, métrique = $10-\text{SNR}d^2$	89
4.9	Décodeur séquentiel 3-chemins, métrique = $1-\text{SNR}d^2$	90
4.10	Décodeur séquentiel 1-chemin, métrique = $1-\text{SNR}d^2$	91

Canal 0.33, 0.67, 1, 0.67, 0.33

4.11	Décodeur séquentiel 2-chemins, métrique entière = $10(1-d^2)$	95
4.12	Décodeur séquentiel 6-chemins, métrique entière = $10(1-d^2)$	96

LISTE DES COURBES (suite)

<u>Figure</u>		<u>Page</u>
4.13	Décodeur séquentiel 1-chemin, métrique = $1-\text{SNR}d^2$	97
4.14	Décodeur séquentiel 3-chemins, métrique = $1-\text{SNR}d^2$	98
<u>Canal 0.38, 0.6, 0.6, 0.38</u>		
4.15	Décodeur séquentiel 1-chemin, métrique = $1-\text{SNR}d^2$	101
4.16	Décodeur séquentiel 3-chemins, métrique = $-1\text{SNR}d^2$	102
4.17	Décodeur séquentiel 5-chemins, métrique = $1-\text{SNR}d^2$	103
<u>Canal 0.5, 1.0, 0.5</u>		
4.18	Décodeur séquentiel 1-chemin métrique entière = $10(1-d^2)$	107
4.19	Décodeur séquentiel 2-chemins, métrique entière = $10(1-d^2)$	108
4.20	Décodeur séquentiel 1-chemin, métrique = $1-\text{SNR}d^2$	109
4.21	Décodeur séquentiel 2-chemins, métrique = $1-\text{SNR}d^2$	110
4.22	Décodeur séquentiel 3-chemins, métrique = $1-\text{SNR}d^2$	111

LISTE DES COURBES (suite)

<u>Figure</u>		<u>Page</u>
	<u>Canal 0.792, 0.165, 0.034</u>	
4.23	Décodeur séquentiel 1-chemin, métrique = $1-\text{SNRd}^2$	115
	<u>Canal 0.649, 0.228, 0.080</u>	
4.24	Décodeur séquentiel 1-chemin, métrique = $1-\text{SNRd}^2$	116

CHAPITRE 1

INTRODUCTION

Une situation classique en communications peut se décrire par une source transmettant de l'information vers un destinataire par l'entremise d'un système quelconque. Ce dernier est appelé canal en toute généralité. Pour une transmission de données numériques l'information est constituée d'une suite de symboles discrets pour lesquels un taux de transmission est établi. Il peut advenir que ce taux soit trop élevé pour le canal utilisé, celui-ci ayant une bande passante limitée. Dans un tel cas le canal est dit "dispersif" car l'information au niveau du récipiendaire est alors dispersée, ou étalée, dans le temps, créant ainsi une situation d'interférence entre symboles. A cause des erreurs de décision que peut provoquer cette situation il est nécessaire d'avoir un traitement des données à la réception. Dans cet exposé l'application de décodeurs séquentiels multi-chemins modifiés pour la correction de l'interférence entre symboles est évaluée. A l'aide de simulations par ordinateur les performances des décodeurs séquentiels sont comparées avec celles présentées dans des travaux décrivant l'utilisation de décodeurs optimaux pour corriger l'interférence entre symboles; ces décodeurs optimaux doivent effectuer un effort de calcul très supérieur à celui d'un décodeur séquentiel multi-chemins, ce qui les rend vite inabordables. L'effort de calcul très raisonnable d'un

décodeur séquentiel multi-chemins en fait une alternative très intéressante pour le traitement de données transmises par un canal dispersif.

Le chapitre 2 de cet exposé présente le problème de l'interférence entre symboles et en donne une définition théorique. A partir de cela le modèle d'un canal dispersif est établi.

Au chapitre 3 un résumé de recherches antérieures sur le sujet d'intérêt est donné. La similitude entre l'interférence entre symboles et un codage convolutionnel est montrée. Après une brève description de ce type de codage, la modélisation de l'interférence entre symboles à l'aide d'un codeur convolutionnel est décrite. Des décodeurs, incluant les décodeurs séquentiels multi-chemins ainsi que le décodeur optimal de Viterbi, sont alors présentés et un type de décodeur séquentiel multi-chemins adapté au problème de l'interférence entre symboles est décrit.

Au chapitre 4 des simulations du décodeur séquentiel modifié sont présentées. Après le calcul des bornes sur les probabilités d'erreur servant à évaluer les résultats, et la description du programme de simulation, des courbes représentant les probabilités d'erreur en fonction de rapports signal sur bruit sont présentées pour des décodeurs séquentiels multi-chemins utilisés avec divers canaux dispersifs. Des tableaux sont également donnés pour mettre en évidence les efforts de calculs des décodeurs simulés pour un

rapport signal sur bruit donné. Une comparaison peut alors se faire entre ces décodeurs séquentiels multi-chemins et le décodeur optimal de Viterbi sur les performances et les efforts de calculs requis.

CHAPITRE 2

DESCRIPTION THEORIQUE DE L'INTERFERENCE ENTRE SYMBOLES ET MODELISATION DU CANAL

2.1 Définition générale du problème

Dans une transmission de données numériques par un canal physique, deux problèmes majeurs peuvent être distingués (Forney 1972). Il s'agit du bruit aléatoire, toujours présent dans un canal, et de l'interférence entre symboles de la séquence transmise. Des erreurs de décision peuvent en résulter à la réception.

Le bruit aléatoire peut se manifester sous forme de salves plus ou moins importantes qui affectent, souvent considérablement, la transmission. Des techniques de codage sont alors employées pour ajouter de la redondance au message transmis de sorte que celui-ci, après avoir été entaché de bruit, puisse être retrouvé à la réception. Une technique ayant prouvé sa supériorité est celle du codage convolutionnel et elle a suscité de nombreuses études; (Viterbi 1967), (Forney 1973), (Haccoun 1974) etc.

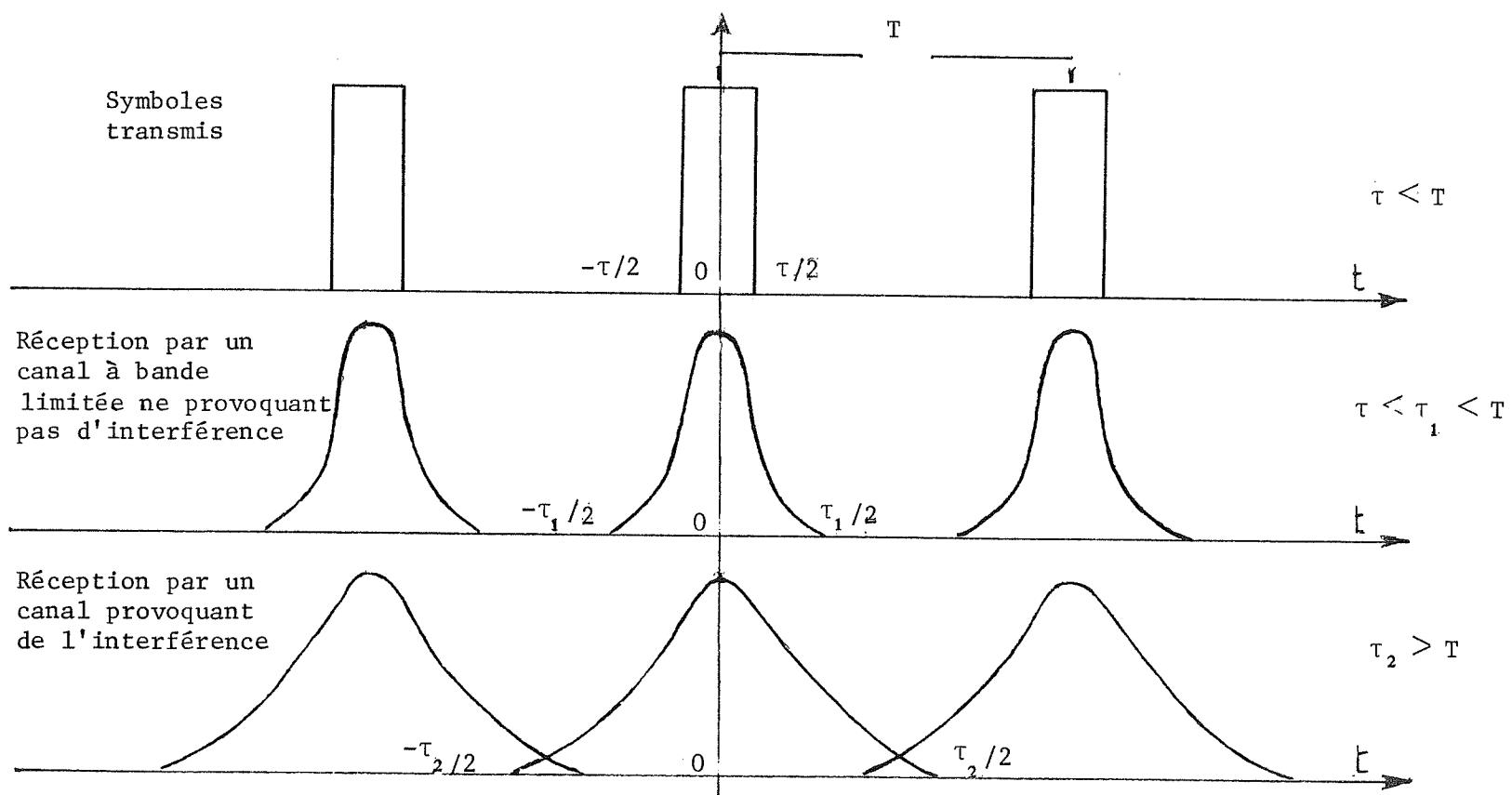
L'interférence entre symboles apparaît lorsque le taux de transmission de données est trop élevé pour la bande passante du canal. Ceci peut se produire après un codage convolutionnel car ce dernier implique une augmentation du nombre de symboles formant la séquence d'information.

Une élévation du taux de transmission dans le canal en résulte pour une communication donnée. Les symboles individuels deviennent de plus en plus étroits dans le temps ce qui produit des spectres très étendus en fréquence. Si le canal n'a pas une bande passante suffisante pour accommoder ces symboles à grand contenu fréquentiel une distorsion des signaux sera présente à la réception. Les symboles auront subi un étalement dans le temps. Advenant que cet étalement dépasse l'intervalle alloué entre des symboles adjacents il se crée une interférence de symbole à symbole pouvant entraîner une importante modification de la séquence d'information (Figure 2.1). Des erreurs de décision peuvent en résulter à la réception et le message reçu sera différent du message transmis.

L'interférence entre symboles peut être considérée comme étant un bruit additionnel venant s'ajouter au bruit aléatoire toujours présent. Ces deux bruits sont cependant de nature différente. L'interférence entre symboles, comme il le sera montré plus tard, est un phénomène déterministe. Toutefois, n'importe quelle technique visant à corriger une séquence reçue devra tenir compte de ces deux bruits simultanément.

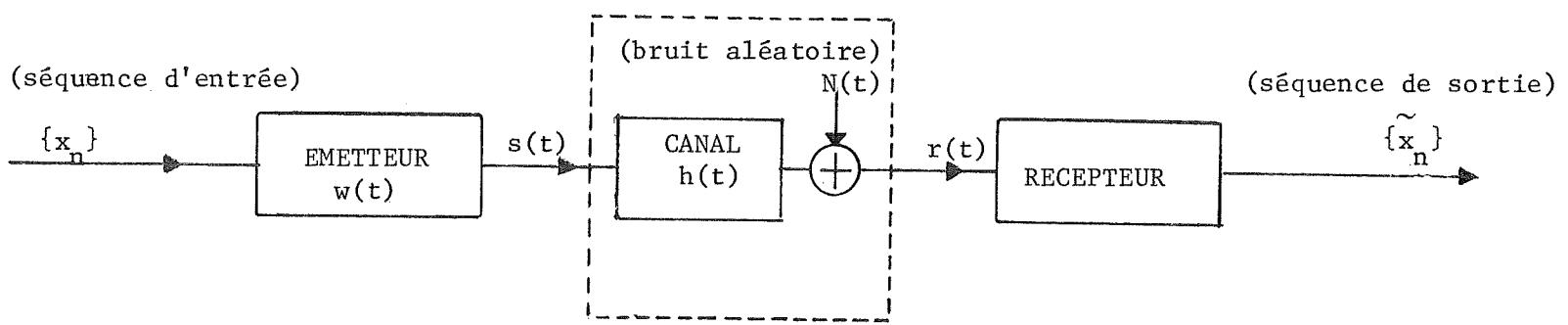
2.2 Définition théorique:

Une chaîne de transmission classique (Figure 2.2) est considérée. La modélisation du canal comprend deux parties distinctes, l'une, de réponse impulsionnelle déterministe $h(t)$, et l'autre, introduisant un bruit aléatoire $N(t)$. Les hypothèses suivantes seront faites:



TRANSMISSION PAR CANAUX DISPERSIFS

FIGURE 2.1



CHAINE DE TRANSMISSION CLASSIQUE

FIGURE 2.2

i) $N(t)$ est un bruit gaussien, blanc, stationnaire et de moyenne nulle.

ii) Le canal est causal, donc $h(t)$ est nul sur l'intervalle t négatif, ce qui revient à dire que son support est borné à gauche par le point $t = 0$.

iii) La réponse impulsionale $h(t)$ est invariante dans le temps. En réalité la fonction $h(t)$ est, en général, variante dans le temps et pourrait être notée par $h_T(t)$. Mais, en toute généralité également, les variations du canal dans le temps sont lentes comparativement au taux de transmission des données. La réponse impulsionale peut alors être considérée comme invariante dans un intervalle de temps fini (Magee 1972). Elle peut donc être notée $h(t)$.

Soit $X(t)$ le signal à transmettre, il est constitué d'une combinaison linéaire, précisée plus loin, d'un nombre fini L d'impulsions de Dirac également espacées d'un intervalle T .

Donc :

$$X(t) = \sum_{n=1}^L x_n \delta(t-nT) \quad (2.1)$$

$X(t)$ est ainsi caractérisé par une suite $\underline{x} = \{x_n\}$ où les x_n sont supposés choisis dans un alphabet M -aire. De plus, il sont supposés statistiquement indépendants, de moyenne nulle et de variance unitaire (Magee 1972).

Avant d'être transmis par le canal le signal $X(t)$ passe par un émetteur représenté à la Figure 2.2. Soit $w(t)$ la réponse impulsionale de l'émetteur et $s(t)$ sa sortie correspondant au signal $X(t)$. Il s'en suit que $s(t)$ est égal au produit de convolution, noté $*$, de $X(t)$ par $w(t)$.

Donc :

$$s(t) = X(t) * w(t) = \sum_{n=1}^L x_n \delta(t-nT) * w(t)$$

$$s(t) = \sum_{n=1}^L x_n w(t-nT) \quad (2.2)$$

$s(t)$ est constitué d'une combinaison linéaire de signaux de la forme $w(t-nT)$ et ces derniers ne se chevauchent pas. Pour cela il est supposé que le support de $w(t)$ (le plus petit intervalle en dehors duquel $w(t)$ est nul) est contenu dans l'intervalle $0 < t \leq T$.

Le canal étant défini par la réponse impulsionale $h(t)$ et le bruit aléatoire $N(t)$, le signal $r(t)$ à l'entrée du récepteur sera donné par :

$$r(t) = s(t) * h(t) + N(t)$$

$$\sum_{n=1}^L x_n w(t-nT) * h(t) + N(t)$$

$$\sum_{n=1}^L x_n \int_{-\infty}^{+\infty} h(\tau) w(t-nT-\tau) d\tau + N(t) \quad (2.3)$$

En posant,

$$g(t) = \int_{-\infty}^{+\infty} h(\tau)w(t-\tau)d\tau \quad (2.4)$$

$r(t)$ devient,

$$r(t) = \sum_{n=1}^{L} x_n g(t-nT) + N(t) \quad (2.5)$$

Pour des considérations développées ultérieurement $g(t)$ sera supposé de carré intégrable. D'autre part la réponse impulsionale $h(t)$ a une durée que l'on suppose finie en pratique. Son support, déjà borné à gauche par le point $t=0$, sera supposé borné à droite par le point $t=KT$, où K est un entier. Etant donné que le support de $w(t)$ est borné à gauche par le point $t=0$ (l'émetteur est également un système causal) et est contenu dans l'intervalle $0 < t < T$, le support de $g(t)$ sera contenu dans l'intervalle $\{0, (K+1)T\}$. Le nombre entier K sera alors défini comme étant l'étendue de $g(t)$ (Forney 1972). En conséquence, pour $K = 0$ il n'y a pas d'interférence entre symboles, et pour $K > 0$ l'interférence s'étend sur K intervalles T . En effet, T étant l'intervalle alloué entre les éléments de $\{x_n\}$, il se produit alors des interférences entre symboles lorsque l'étendue des $g(t-nT)$, résultat du produit convolutionnel de $h(t)$ par $s(t)$, dépasse cet intervalle T . Il est alors apparent que cet étalement des symboles n'a rien à voir avec le bruit aléatoire $N(t)$. Il en résulte que l'interférence entre symboles est un phénomène entièrement déterministe. Du fait de la présence d'un bruit aléatoire la correction de l'interférence entre symboles nécessite une détection optimale à la sortie

du canal afin de diminuer autant que possible, l'effet de ce bruit aléatoire. Forney (1972) a remarqué, d'après l'équation (2.5), qu'à part le bruit $N(t)$, $r(t)$ est formé par une combinaison linéaire de fonctions de base de carré intégrable $g(t-nT)$ et, par conséquent, qu'une détection optimale en présence du bruit blanc et gaussien peut être réalisée avec un ensemble de filtres adaptés à ces fonctions de base (Wozencraft et Jacobs 1965). Le nombre de ces fonctions de base, d'après l'équation (2.5), est égal au nombre d'éléments formant la suite $\{x_n\}$. Forney (1972) a décomposé la fonction $g(t)$ en une somme de K plus une fonction $g_f(t)$, nulles en dehors de l'intervalle $[0, T]$, comme suit:

$$g(t) = g_0(t) + g_1(t-T) + \dots + g_K(t-KT) \quad (2.6)$$

et à partir de cela il indiqua que les sorties " a_m " des filtres adaptés, échantillonnées à T secondes, et données par

$$a_m = \int_{-\infty}^{+\infty} r(t) g(t-mT) dt \quad m = 1, 2, \dots, L \quad (2.7)$$

formaient une statistique suffisante pour l'estimation future de la suite $\{x_n\}$. De plus, en examinant l'équation (2.7) il apparaît que les a_m ne sont en fait que les échantillons de la sortie d'un filtre de réponse impulsionnelle $g(-t)$. En effet,

$$\begin{aligned} r(t) * g(-t) & \Big|_{t=mT} = \int_{-\infty}^{+\infty} r(\tau) g(\tau-t) d\tau \Big|_{t=mT} \\ & = \int_{-\infty}^{+\infty} r(\tau) g(\tau-mT) d\tau = a_m \end{aligned} \quad (2.8)$$

Le développement de (2.8) donne,

$$\begin{aligned}
 a_m &\triangleq a(mT) = \left[\int_{-\infty}^{\infty} r(\tau) g(\tau-t) d\tau \right] \Big|_{t=mT} \\
 &= \left[\int_{-\infty}^{\infty} \left\{ \sum_{n=1}^L x_n g(\tau-nT) + N(\tau) \right\} g(\tau-t) d\tau \right] \Big|_{t=mT} \\
 &= \int_{-\infty}^{\infty} \sum_{n=1}^L x_n g(\tau-nT) g(\tau-mT) d\tau + N'_m
 \end{aligned}$$

avec $N'_m = \int_{-\infty}^{\infty} N(\tau) g(\tau-mT) d\tau$ (2.9)

D'après l'équation (2.6) la fonction d'autocorrélation de $g(t)$ sera définie par,

$$R_n = \begin{cases} \int_{-\infty}^{\infty} g(t) g(t+nT) dt & 0 \leq |n| \leq K \\ 0 & \text{ailleurs} \end{cases} \quad (2.10)$$

ce qui est faisable, $g(t)$ étant de carré intégrable. Il apparaît donc que dans l'intervalle $0 \leq |n| \leq K$ il y a $2K+1$ éléments R_n séparés de T secondes, ce qui englobe $2KT$ secondes en tout. De plus, en posant $t+nT = \tau$, $dt = d\tau$, R_n devient,

$$R_n = \begin{cases} \int_{-\infty}^{\infty} g(\tau-nT) g(\tau) d\tau & 0 \leq |n| \leq K \\ 0 & \text{ailleurs} \end{cases} \quad (2.11)$$

ce qui implique que $R_n = R_{-n}$.

Avec l'équation (2.11) il est trouvé que,

$$\begin{aligned}
& \int_{-\infty}^{\infty} g(\tau-mT) g(\tau-nT) d\tau \\
&= \int_{-\infty}^{\infty} g(t) g(t-nT+mT) dt \quad (\text{avec } t = \tau - mT, dt = d\tau) \\
&= \int_{-\infty}^{\infty} g(t) g(t+(m-n)T) dt \\
&= R_{m-n} \tag{2.12}
\end{aligned}$$

A partir de cela les sorties a_m sont données par,

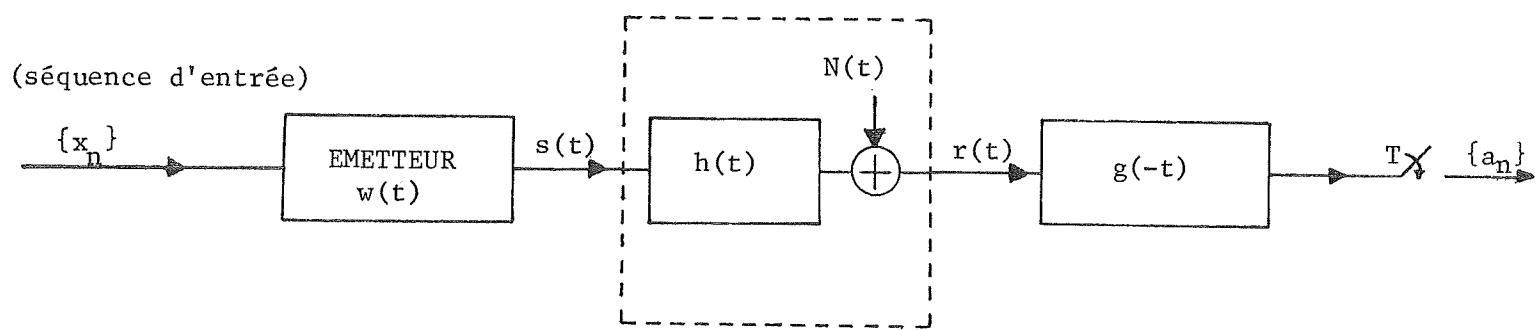
$$\begin{aligned}
a_m &= \sum_{n=1}^L x_n \int_{-\infty}^{\infty} g(\tau-mT) g(\tau-nT) d\tau + N_m' \\
&= \sum_{n=1}^L x_n R_{m-n} + N_m' \\
&= x_m R_0 + \sum_{n \neq m} x_n R_{m-n} + N_m' \tag{2.13}
\end{aligned}$$

D'après l'équation (2.10), pour $L > 2K+1$, la sommation dans (2.13) comportera des termes nuls. Le premier terme de (2.13) désigne le signal désiré à l'instant mT , N_m' désigne un échantillon du bruit aléatoire au temps mT , et, finalement, la sommation indique l'interférence provenant des signaux aux autres instants d'échantillonnage. Encore une fois, d'après l'équation (2.13), il est apparent que l'interférence entre symboles est entièrement déterminée à partir de la réponse impulsionnelle du canal, se distinguant ainsi du bruit aléatoire.

2.3 Modélisation du canal

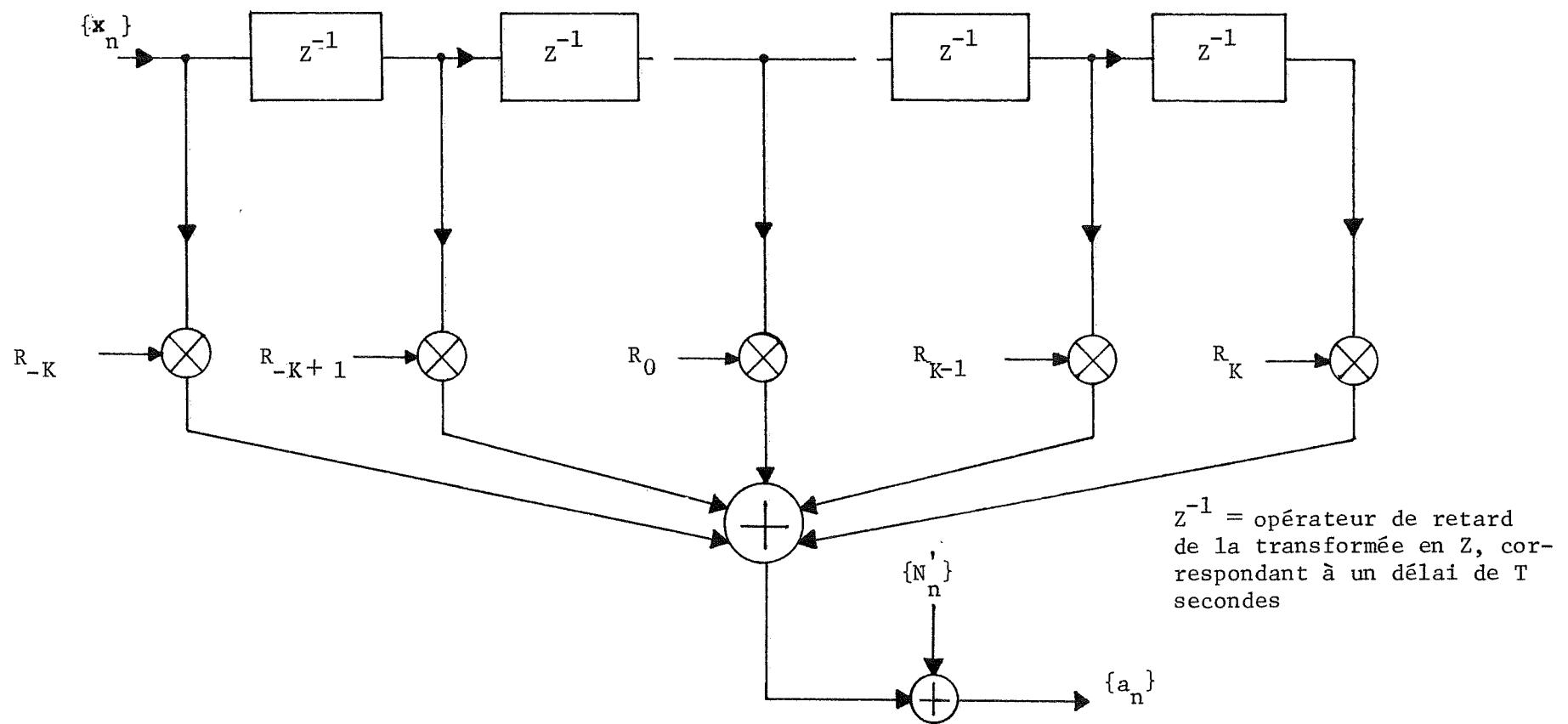
Les éléments de $\{x_n\}$ étant séparés de T secondes, l'émetteur fonctionne au rythme de $\frac{1}{T}$ symboles par seconde, comme l'échantillonneur suivant le filtre adapté. A partir de cela la cascade $w(t)$, $h(t)$ et $g(-t)$, suivie de l'échantillonneur, peut être précisée dans la chaîne de transmission (Figure 2.3) et, d'après l'équation (2.10), cette cascade englobe $2KT$ secondes. Ceci peut être représenté par un filtre transversal (Figure 2.4) défini sur $2KT$ secondes et ayant comme coefficients de pondération les $2K+1$ valeurs échantillonnées R_n de la fonction d'auto-corrélation de $g(t)$. Les éléments de $\{x_n\}$ entrent dans le filtre en série et remplissent le registre. A chaque nouvelle entrée le contenu du registre est décalé vers la droite pour faire place au nouvel élément, ce qui implique la disparition de l'élément occupant la dernière case. Pour compléter ce modèle un échantillon de bruit aléatoire est ajouté à la sortie du filtre. L'opérateur Z^{-1} de la transformée en Z se trouvant dans les cases du registre indique le délai de T secondes qui sépare deux éléments de $\{x_n\}$. Ce modèle n'est rien d'autre que la représentation schématique de l'équation (2.13).

D'après l'équation (2.9) il apparaît que le bruit aléatoire n'est plus blanc, bien que toujours gaussien. Comme il est toujours plus souhaitable de travailler avec du bruit blanc, un filtre de "blanchiment" peut être réalisé pour restituer le bruit blanc à partir du bruit "coloré".



INCLUSION D'UN FILTRE ADAPTE APRES LE CANAL

FIGURE 2.3



FILTRE TRANSVERSAL REPRESENTANT LE CANAL ET LE FILTRE ADAPTE

FIGURE 2.4

Il est bien connu que l'insertion d'un tel filtre n'augmentera nullement la probabilité d'erreur si l'il s'agit d'une opération réversible (Wozencraft et Jacobs 1965).

Soit $(\frac{N_o}{2})$ la densité spectrale du bruit blanc. Le spectre de densité de puissance du bruit coloré sera alors donné par,

$$P_N = \left(\frac{N_o}{2} \right) |G(f)|^2 \quad (2.14)$$

où $G(f)$ est la transformée de Fourier de $g(t)$. En prenant la transformée de Fourier inverse de P_N , la valeur au point mT de la fonction d'auto-corrélation du bruit coloré est obtenue comme suit

$$E \{ N'_k N'_{k-m} \} = \mathcal{F}^{-1} [P_N] \left| = \frac{N_o}{2} [g(t) * g(-t)] \right| \quad (2.15)$$

$$t = mT \qquad \qquad \qquad t = mT$$

où E représente l'espérance mathématique.

Les échantillons de bruit coloré ne seront plus indépendants l'un de l'autre et, d'après les équations (2.10) et (2.15), la corrélation entre les k -ième et j -ième échantillons de bruit sera donnée par

$$E[N'_k N'_j] = \begin{cases} \left(\frac{N_o}{2} \right) R_{k-j} & \text{pour } |k-j| \leq K \\ 0 & \text{ailleurs} \end{cases} \quad (2.16)$$

Pour trouver l'expression du filtre de blanchiment la transformée en Z de la suite des R_n dans l'intervalle $[-K, K]$ peut être considérée (Proakis 1978).

Ceci donne,

$$R(z) = \sum_{n=-K}^K R_n z^{-n} \quad (2.17)$$

Puisque $R_n = R_{-n}$, alors,

$$R(z) = R(z^{-1}) \quad (2.18)$$

Les équations (2.17) et (2.18) impliquent que $R(z)$ a $2K$ racines avec la particularité que si r est une racine, $1/r$ est aussi une racine. $R(z)$ peut alors se factoriser ainsi,

$$R(z) = F(z) F(z^{-1}) \quad (2.19)$$

où $F(z)$ est un polynôme de degré K ayant r_1, r_2, \dots, r_k comme racines, et $F(z^{-1})$ est un polynôme de même degré ayant comme racines, $1/r_1, 1/r_2, \dots, 1/r_k$.

D'après les équations (2.16) et (2.19) il apparaît que la fonction d'auto-corrélation du bruit coloré peut être décrite sous forme de transformée en Z comme suit:

$$\frac{N_o}{2} R(z) = \frac{N_o}{2} F(z) F(z^{-1}) \quad (2.20)$$

Si $N(z)$ définit le bruit blanc en transformée en Z , et $N'(z)$ le bruit coloré, ce dernier peut alors s'écrire

$$N'(z) = N(z) F(z^{-1}) \quad (2.21)$$

ce qui est compatible avec l'équation (2.20), (Forney 1972).

D'après l'équation (2.21) le filtre de blanchiment qui restituera $N(Z)$ à partir de $N'(Z)$ sera tout simplement de la forme $\frac{1}{F(Z^{-1})}$. Comme il a été montré précédemment la i -ième racine de $F(Z^{-1})$ peut être soit r_i , soit $\frac{1}{r_i}$. Le choix des racines se fait de sorte qu'elles soient toutes à l'intérieur du cercle unité produisant ainsi un système stable, (Proakis 1978). $F(Z^{-1})$ étant un polynôme de degré K il y a 2^K possibilités pour le choix de ces racines.

En conclusion, une sortie a_m définie par l'équation (2.13) passe par le filtre de blanchiment pour donner un symbole y_m . La modélisation du canal, incluant le récepteur et le filtre de blanchiment, peut alors être décrite. En exprimant la transformée en Z de la séquence d'entrée par $X(Z)$, celle de la séquence de sortie du canal par $A(Z)$, il est clair, d'après ce qui précède, que

$$\begin{aligned} A(Z) &= X(Z) R(Z) + N'(Z) \\ &= X(Z) F(Z) F(Z^{-1}) + N(Z) F(Z^{-1}) \end{aligned} \quad (2.22)$$

et en incluant le filtre de blanchiment,

$$Y(Z) = \frac{A(Z)}{F(Z^{-1})} = X(Z) F(Z) + N(Z) \quad (2.23)$$

Si

$$F(Z) = \sum_{n=0}^K f_n Z^{-n} \quad (2.24)$$

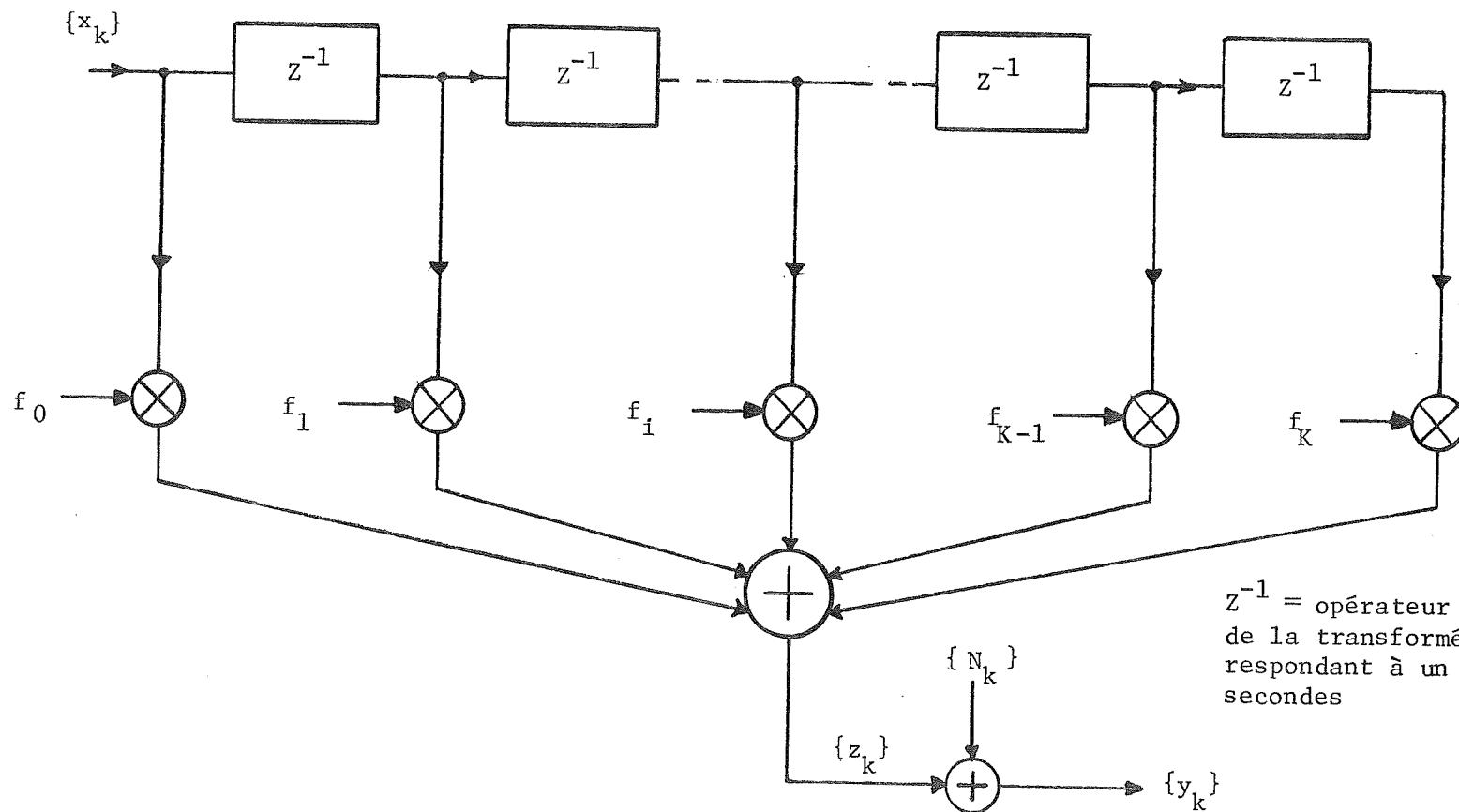
un symbole y_m sera alors donné par

$$y_m = \sum_{n=0}^K f_n x_{m-n} + n_m \quad (2.25)$$

où N_m est un échantillon de bruit blanc gaussien et les f_n représentent les coefficients de pondération d'un filtre transversal ayant $F(Z)$ comme fonction de transfert. Ce filtre n'englobe plus que KT secondes et est représenté à la figure 2.5. La modélisation est maintenant complète. y_m représente le symbole de sortie correspondant au symbole d'entrée x_m . y_m est le résultat de l'interférence entre symboles qu'a subit x_m en étant transmis par un canal dispersif et de l'addition d'un bruit aléatoire blanc et gaussien.

Comme il a déjà été mentionné le canal est effectivement variant dans le temps. Des techniques pour identifier ces variations ont été développées, (Magee 1972, Proakis 1978). Par exemple, un signal pseudo-aléatoire reconnaissable pourrait être mélangé au signal d'information et servir à l'ajustement des coefficients du filtre transversal à intervalles réguliers ou de façon continue. (Proakis 1978)

La modélisation étant complétée il est maintenant possible de tirer partie de sa structure pour trouver une méthode de correction efficace de l'interférence entre symboles et ceci fait l'objet du chapitre suivant.



MODELISATION FINALE DE LA CHAINE DE TRANSMISSION

FIGURE 2.5

CHAPITRE 3

CORRECTION DE L'INTERFERENCE ENTRE SYMBOLES

3.1 Recherches précédentes

Comme l'indique Magee (1972), le fait de laisser un intervalle suffisant entre symboles pour que leurs étalements ne provoquent pas d'interférence ne constitue pas une solution élégante au problème car le taux de transmission en est réduit considérablement. Ce sera également le cas si le modulateur émet des symboles de durée assez longue pour que l'interférence soit relativement insignifiante. De plus, le fait que la réponse impulsionale ne soit pas invariante dans le temps augmente la difficulté du traitement des données à la réception. Il faut donc traiter le signal recueilli à la sortie du canal par un filtre spécial appelé "égaliseur" qui, tout en tenant compte du bruit aléatoire, restituera au mieux l'information émise, dénuée d'interférence entre symboles. Cette interférence étant déterministe, elle est entièrement décrite par la réponse impulsionale du canal, ce qui implique que la première fonction de l'égaliseur en est une d'identification de la réponse dans le temps. La deuxième fonction est de corriger l'information reçue afin de retrouver au mieux les données originales émises.

Il existe des égaliseurs linéaires. Ceux-ci sont ajustables ce qui leur permet de suivre les variations temporelles du canal.

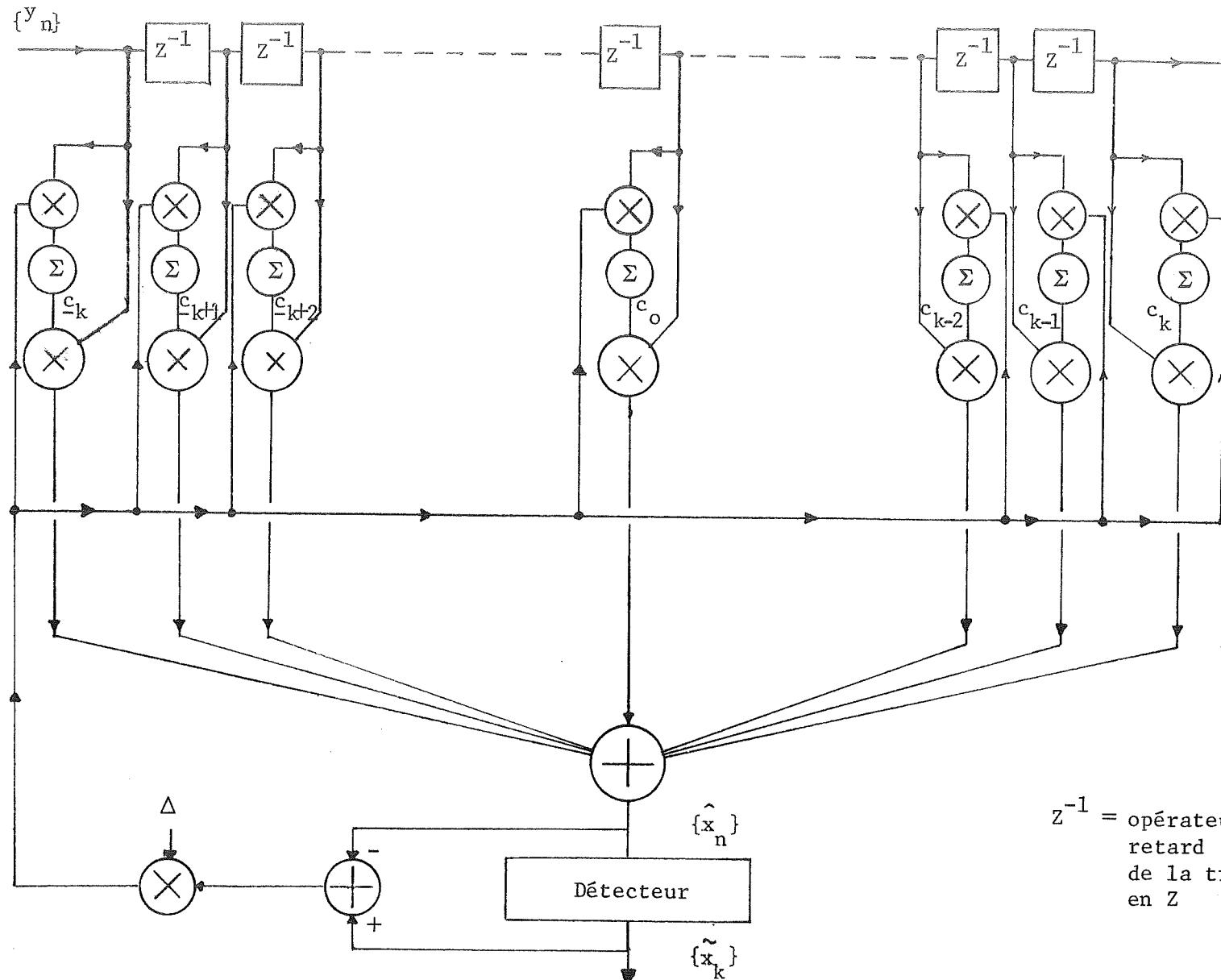
Le modèle le plus courant est celui de l'égaliseur transversal, (Figure 3.1), (Proakis 1978). C'est essentiellement un filtre transversal comme celui utilisé pour représenter le canal (Figure 2.4). Les sorties du canal remplissent en série le registre du filtre. Chaque membre du registre est multiplié par le coefficient correspondant C_i , et la somme totale forme un estimé \hat{x}_n du n ième symbole d'information. L'estimé \hat{x}_n passe alors par un détecteur qui le quantifie à la valeur du symbole \tilde{x}_n le plus proche dans l'alphabet des symboles d'information $\{x_n\}$. L'écart entre \tilde{x}_n et \hat{x}_n est multiplié par un coefficient Δ et ce résultat sert au réglage des coefficients C_i . Cette contre-réaction permet donc un réglage constant des C_i pour diminuer l'écart entre \tilde{x}_n et \hat{x}_n corrigéant ainsi l'interférence entre symboles. Une erreur de décision est commise si \tilde{x}_n diffère de x_n , le n ième symbole véritablement transmis. L'optimisation des coefficients C_i a suscité beaucoup de recherches. Divers critères sont présentés par Proakis, (Proakis 1978); par exemple, l'erreur quadratique moyenne, définie comme

$$J(\underline{C}) = E |x_n - \hat{x}_n|^2, \underline{C} = (C_{-k}, C_{-k+1}, \dots, C_{k-1}, C_k) \quad (3.1)$$

est minimisée. C'est une fonction convexe des coefficients C_i . Ces derniers sont utilisés dans le produit de convolution discret suivant,

$$\hat{x}_n = \sum_{j=-k}^k C_j y_{n-j} \quad (3.2)$$

sortie du canal



Z^{-1} = opérateur de retard de la transformée en Z

ÉGALISEUR TRANSVERSAL LINÉAIRE

Figure 3.1

Il est trouvé (Proakis 1978) que le vecteur optimum $\underline{C}_{\text{opt}}$ est donné par,

$$\underline{C}_{\text{opt}} = [\underline{A}]^{-1} \underline{\alpha} \quad (3.3)$$

où $[\underline{A}]$ est la matrice de covariance ayant pour éléments,

$$a_{ij} = E | y_{n-i}^* y_{n-j} | \quad (\text{la notation } y^* \text{ représente le complexe conjugué de } y) \quad (3.4)$$

et $\underline{\alpha}$ représente le vecteur ayant pour composantes,

$$\alpha_i = E | I_n y_{n-i}^* |, \quad i = -k, \dots, k \quad (3.5)$$

Les essais effectués avec cet égaliseur ont montré qu'il n'était pas performant avec des canaux provoquant une interférence entre symboles élevée, (Magee 1972). Le filtre de Kalman a également été envisagé (Magee 1972). Il est plus performant que les précédents, sans être toutefois transcendant, mais sa réalisation est compliquée. La conclusion qui s'est dégagée de ces essais est que les filtres linéaires ont de faibles performances avec des canaux très dispersifs, ce qui nécessite l'utilisation d'autres techniques, notamment l'égalisation non linéaire, (Magee 1972).

3.2 Techniques non linéaires

Il y a principalement deux techniques non linéaires de base. La première consiste à corriger individuellement les symboles provenant du canal, alors que la seconde travaille sur une séquence de ces symboles.

Magee (1972), cite les principaux égaliseurs fonctionnant selon la première technique, comme le détecteur d'Abend et Frichtman qui minimise la probabilité d'erreur en traitant chaque symbole séparément; si l'information est constituée de +1 et de -1 équiprobables et indépendants à la transmission, le test suivant est effectué pour minimiser la probabilité d'erreur à la réception;

Si,

$$P(y_n/x_n = -1) > P(y_n/x_n = +1) \quad (3.6)$$

la décision se prendra en faveur de -1. Si l'inégalité ne tient pas c'est +1 qui sera choisi. Cette opération est recommencée pour x_{n+1} et tous les symboles qui suivent. Avec des symboles x_i équiprobables la loi de Bayes permet de prendre les mêmes décisions que ci-dessus à partir du test suivant:

$$P(x_n = -1/y_n) > P(x_n = +1/y_n) \quad (3.7)$$

Il s'avère que ce test minimise la probabilité d'erreur même pour le cas où les symboles d'entrée x_i ne seraient pas équiprobables, (Magee 1972); de plus, sa réalisation pratique est plus simple que celle du premier test. Quand l'information n'est pas binaire ce test se généralise facilement, et la décision se fera en faveur du symbole correspondant à la probabilité conditionnelle la plus élevée. Malheureusement, l'égaliseur bâti à partir de cette technique est quand même fort complexe et ses performances bien qu'acceptables laissent encore à désirer (Magee 1972).

C'est également le cas d'autres égaliseurs analysant l'information symbole par symbole.

Vient alors la seconde classe des égaliseurs non linéaires. Ceux-ci fonctionnent en analysant une séquence de symboles reçus. La décision prise à partir d'un symbole particulier tient compte d'un certain nombre de symboles précédents. L'énorme avantage d'une telle technique est que la réalisation physique d'un système minimisant la probabilité qu'une séquence soit erronée est bien plus simple que celle d'un système minimisant le nombre moyen de symboles en erreur. De plus, les décisions prises sous cette optique sont asymptotiquement optimales. L'utilisation de telles techniques a été favorisée depuis que l'interférence entre symboles fut associée, de par sa nature déterministe, à un codage convolutionnel. Ce type de codage s'effectuant à partir d'une séquence de symboles, il est assez simple de faire le rapprochement avec l'interférence entre symboles.

3.3 Similitude avec le codage convolutionnel

Parmi les systèmes de codage visant à lutter contre le bruit aléatoire d'un canal de transmission le codage convolutionnel est reconnu comme l'un des plus performants. Il consiste à ajouter de la redondance à une séquence de données de façon uniforme et continue de sorte que chaque symbole codé contienne de l'information sur un certain nombre de ses prédécesseurs.

La représentation d'un codeur convolutionnel se fait à l'aide d'un registre à décalage (Figure 3.2a) constitué de K cases, ou cellules, reliées à v additionneurs; ces derniers sont modulo-2 pour un codage binaire s'effectuant sur des données également binaires. Chaque cellule peut être reliée à quelques additionneurs parmi les v , à tous, ou encore à aucun d'entre eux. La façon dont les cellules sont reliées aux additionneurs spécifie le code. Comme le montre la figure (3.2a), les symboles de la séquence de données entrent en série dans le codeur à partir de la première cellule. Un symbole accédant au registre décale d'une case vers la droite les symboles s'y trouvant déjà; le symbole qui occupait la première cellule passe à la deuxième et ainsi de suite jusqu'au symbole occupant la K ième cellule qui disparaît en laissant sa place à celui de la $K-1$ ième cellule. Chaque fois qu'un symbole entre dans le registre le contenu de celui-ci est additionné selon les connexions aux additionneurs (modulo-2 dans le cas binaire) pour donner v symboles de la séquence codée. Le rapport $1/v$ est défini comme étant le taux de codage. Le nombre K de cellules représente la longueur de contrainte du codeur. L'état du codeur est défini comme étant le contenu des $K-1$ premières cellules. La connaissance de l'état du codeur et du prochain symbole à entrer dans le registre permet de déterminer la valeur des v symboles de sortie qui en résulteront, ainsi que le prochain état. Donc, pour chaque nouveau symbole d'entrée le codeur tient compte des $K-1$ symboles précédents pour déterminer les symboles codés.

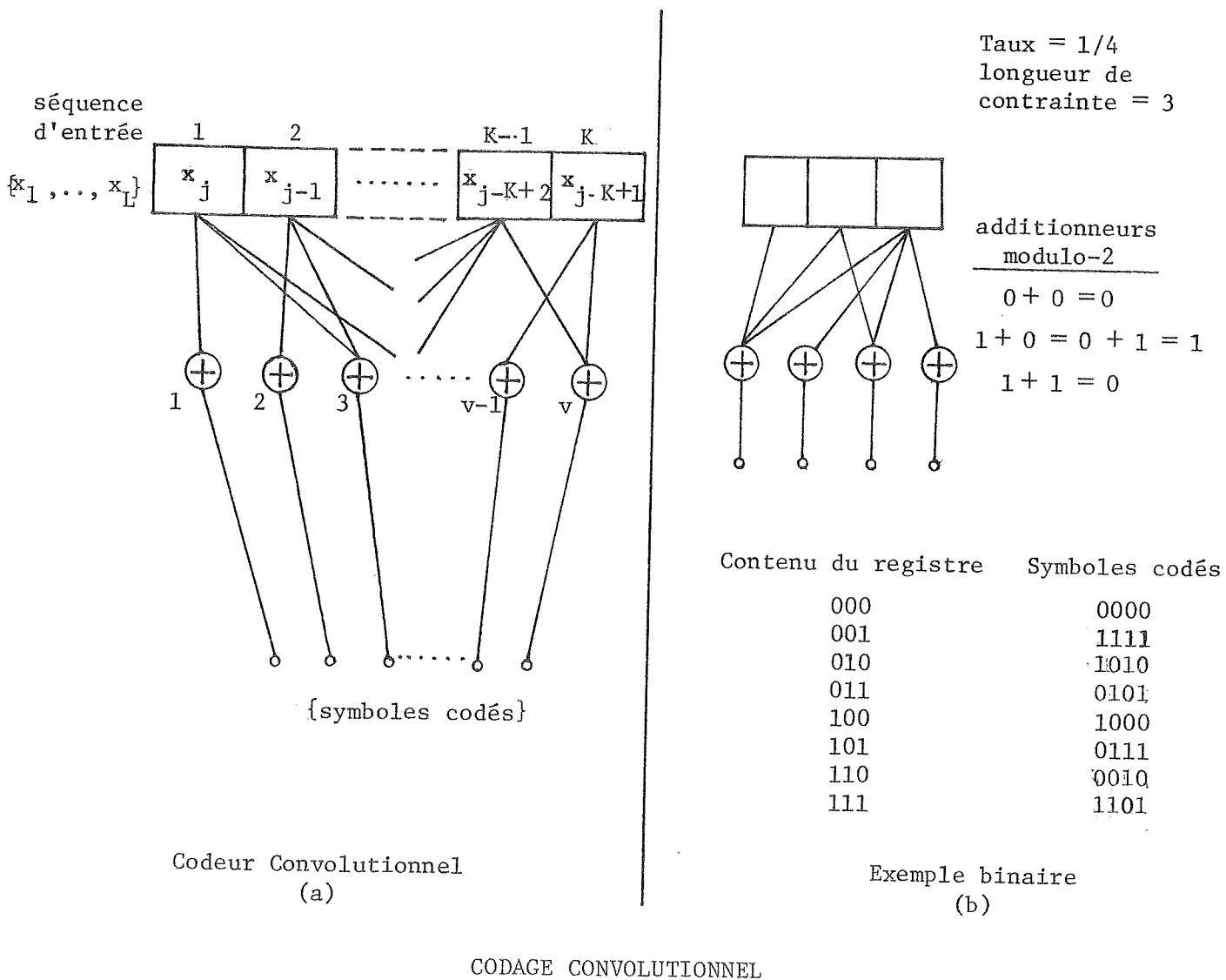


Figure 3.2

Un exemple simple d'un tel codeur est représenté à la figure.

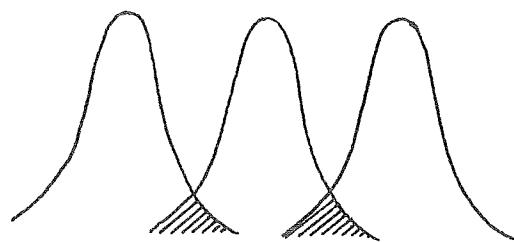
3.2b. Le code généré est de taux 1/4 et la longueur de contrainte est de 3. La séquence d'entrée est binaire, formée de 0 et de 1, et les additionneurs sont modulo-2. Si le contenu du registre est 011, l'état est 01. Si le prochain symbole qui entre est un 0 la sortie codée sera 1111, correspondant au contenu 001; l'état du codeur passe de 01 à 00. Par contre si c'est un 1 qui entre dans le registre la sortie sera 0111, correspondant au contenu 101; l'état passe alors de 01 à 10.

En comparant le modèle du canal développé au premier chapitre (Figure 2.5), avec le codeur convolutionnel classique, (Figure 3.2a), la similitude apparaît; un canal dispersif peut être représenté par un codeur convolutionnel de longueur de contrainte égale à l'étendue plus un de l'interférence entre symboles, avec un seul additionneur (non-modulo) dont les entrées sont pondérées par les coefficients f_i .

La figure 3.3a illustre cette similitude à l'aide d'un exemple simple. Soit un canal dispersif provoquant une interférence entre un symbole transmis et ses deux voisins adjacents. Le canal est identifié et sa réponse impulsionale discrète est définie par,

$$f_0 = 0.5, f_1 = 1.0, f_2 = 0.5 \quad (3.8)$$

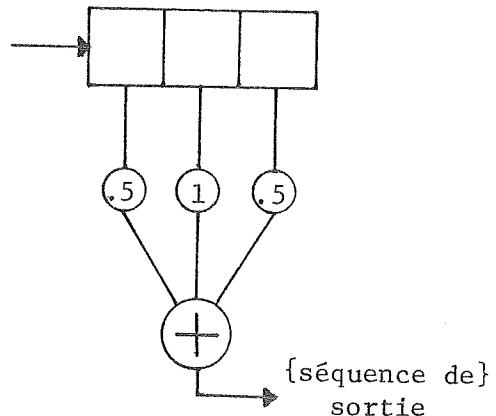
Le canal est alors représenté par le codeur convolutionnel équivalent de la figure 3.3b.



//// zones d'interférence

Interférence d'un symbole sur ses deux voisins adjacents
(a)

{séquence}
d'entrée



{séquence de}
sortie

Contenu du
registre

-1 -1 -1	-2
-1 -1 1	-1
-1 1 -1	0
-1 1 1	+1
1 -1 -1	-1
1 -1 1	0
1 1 -1	+1
1 1 1	+2

Symboles de
sortie

Modélisation par codage
convolutionnel
(b)

MODELISATION DE L'INTERFERENCE
ENTRE SYMBOLES

Figure 3.3

Si une séquence d'entrée binaire est transmise à l'aide de ce canal, une séquence différente de même longueur en résultera à la sortie. L'additionneur n'étant pas modulo-2, cette séquence sera formée de symboles qui ne seront plus forcément binaires. Soit l'alphabet binaire composé de $(+1, -1)$ pour l'entrée; si la séquence $\{+1, +1, +1, -1, +1, -1, -1, 1, -1, -1\}$ est transmise, la séquence de sortie sera alors, (avec le registre initialement rempli de -1), $\{-1, +1, +2, +1, 0, 0, -1, -1, 0, -1\}$.

L'interférence entre symboles a donc été représentée comme un codage. Evidemment ce codage peut ne pas être reluisant. En effet, comme le montre la figure 3.3b, des contenus de registre différents peuvent donner une même sortie. Des erreurs de décision peuvent en résulter. De plus, comme le montre la figure 2.4, l'information subit une altération supplémentaire due au bruit aléatoire.

Avec cette modélisation un canal ne provoquant aucune interférence aura une réponse impulsionnelle discrète formée d'un seul coefficient, $f_0 = 1$, et un symbole d'information n'influencera aucunement ses voisins.

3.4 Techniques de décodage optimales

L'interférence entre symboles ayant été représentée comme un codage convolutionnel, les techniques de décodage appropriées devront maintenant être étudiées.

Pour mieux assimiler ces techniques il est utile de schématiser le codeur par un diagramme d'état. Ce dernier montre de façon simple et précise toutes les transitions possibles d'états ainsi que les symboles codés correspondants. Pour un codeur de longueur de contrainte K , il y a 2^{K-1} états possibles. Avec l'exemple de la figure 3.3b le diagramme d'état obtenu sera celui de la figure 3.4. Avec le codeur en question il y a quatre états différents représentés par les quatre noeuds du diagramme. Les flèches représentent les transitions et le chiffre entre parenthèses sous une flèche indique le symbole d'entrée entraînant cette transition. De plus, les sorties correspondantes du codeur sont indiquées au dessus des flèches. Les données étant binaires il y a deux transitions partant de chaque noeud, soit une pour +1 et une pour -1. Le diagramme d'état distingue les transitions permisibles de celles qui ne le sont pas.

En effet, la figure 3.4 montre qu'il est impossible, par exemple, de passer directement de l'état -1 -1 à l'état 11, ce qui implique qu'une séquence détectée comportant un -2 suivi d'un +2 à la réception est erronée. Mais le bruit aléatoire du canal peut également provoquer des erreurs moins évidentes en produisant une séquence de sortie différente de la séquence émise mais avec une succession de symboles permisibles, en accord avec le diagramme d'état.

Avec cette représentation il est possible de trouver la succession d'états provoqués par une séquence d'entrée déterminée.

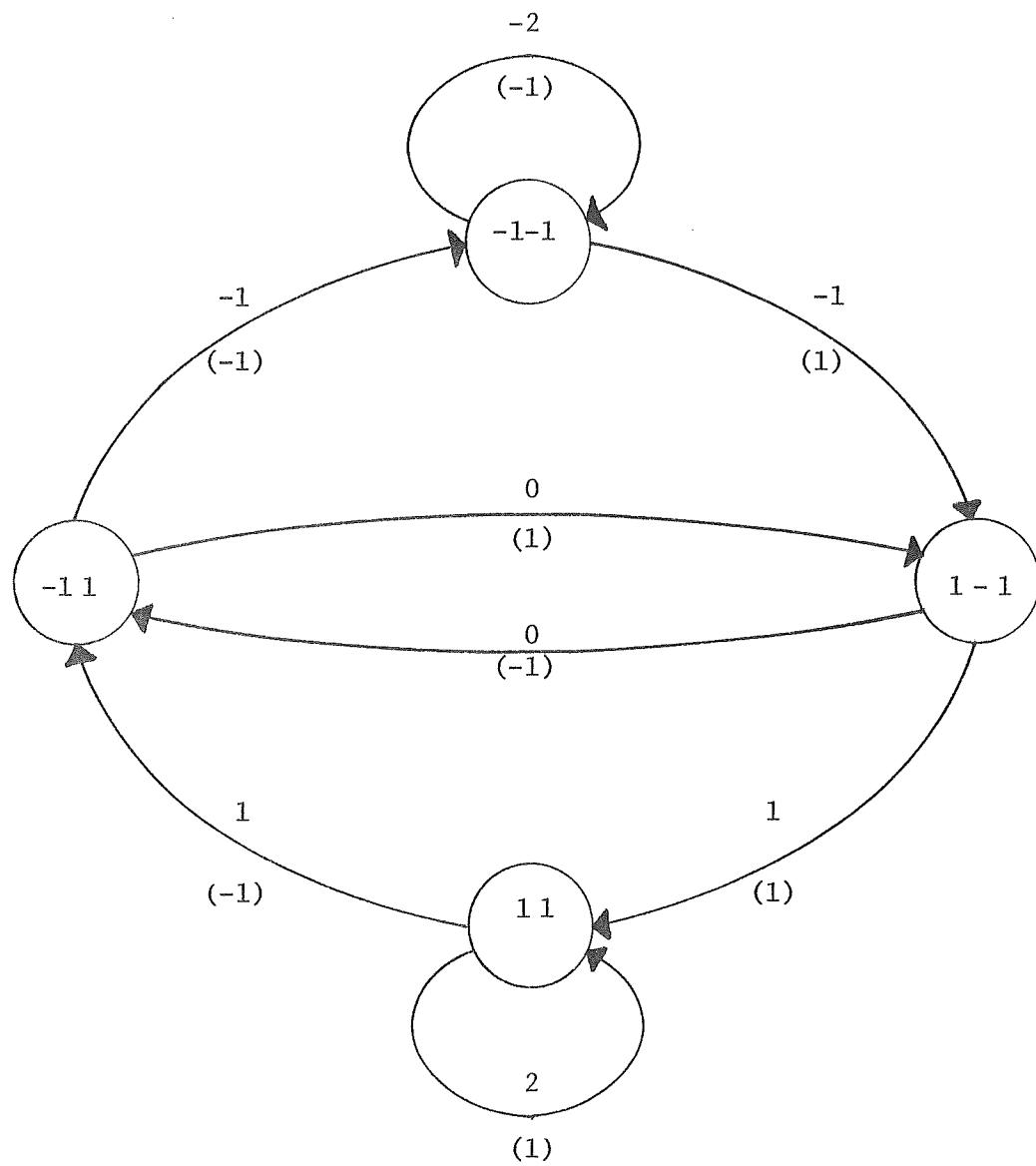


DIAGRAMME D' ETAT

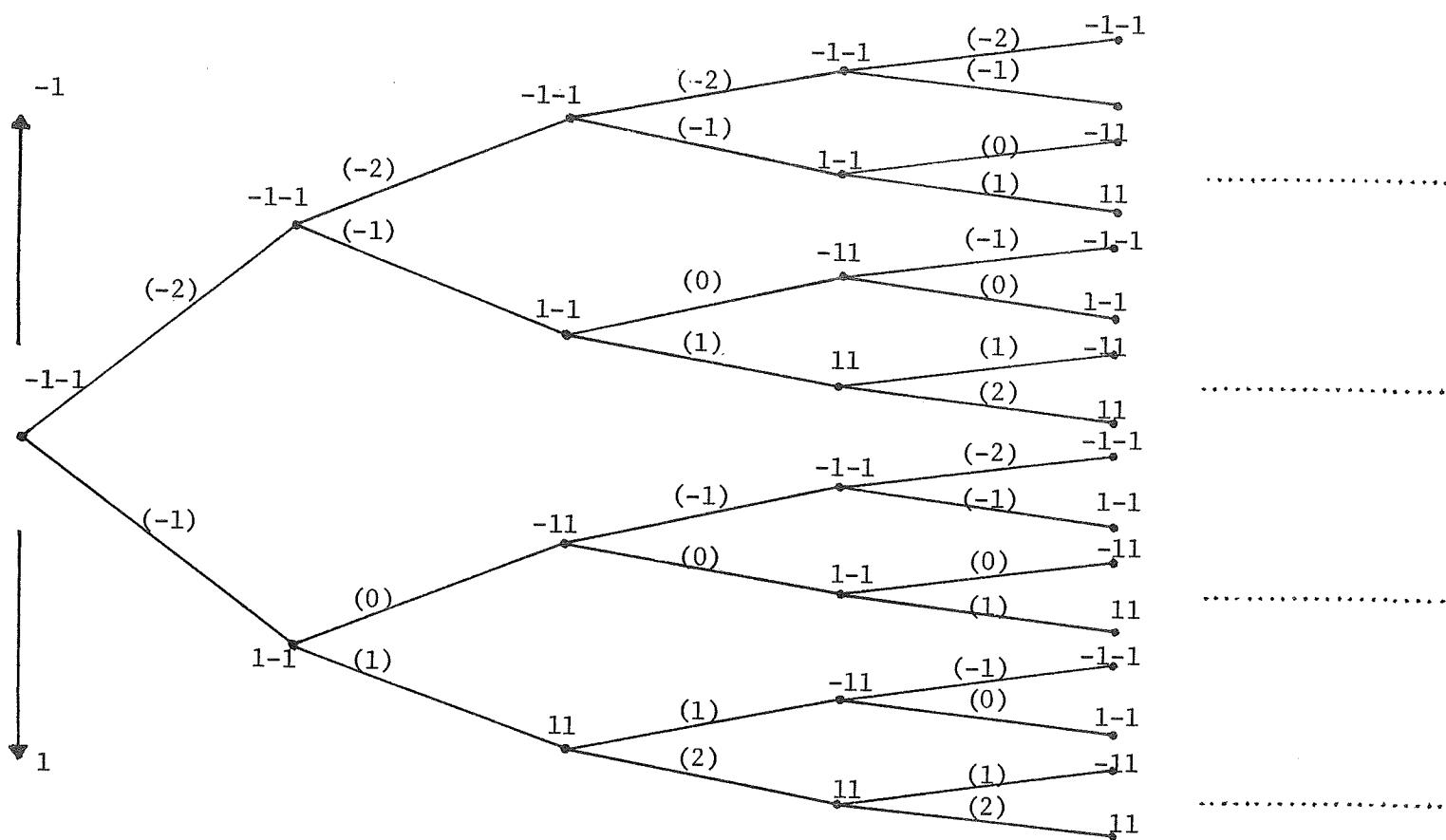
CORRESPONDANT A

LA FIGURE 3.3(b)

Figure 3.4

Si L symboles binaires forment une séquence d'entrée, il y aura au plus 2^L séquences possibles différentes. Chacune de ces séquences définira une succession spécifique d'états. Une représentation en arbre peut aussi être considérée (Figure 3.5). Le noeud de départ désigne l'état -1 -1. A partir de ce noeud les deux branches définissant les transitions provoquées par l'entrée d'un -1 ou d'un +1 dans le codeur sont tracées (une transition vers le haut correspondant à -1). Elles mènent aux deux noeuds spécifiant les états possibles après qu'une transition se soit faite. Encore une fois les symboles codés sont donnés au-dessus des transitions. Pour chaque nouveau noeud les deux branches de transition sont déterminées ce qui donnera en tout un arbre à L niveaux définissant 2^L chemins différents. Un décodeur devra pouvoir déterminer, à partir de cet arbre et de ce que le récepteur a détecté comme séquence codée, le chemin le plus vraisemblable. Le décodeur peut alors donner la séquence binaire correspondante de L symboles comme résultat. Idéalement cette séquence sera identique à la séquence d'entrée originale.

Comme il a déjà été mentionné, les noeuds dans la représentation en arbre sont associés aux états du codeur. De ce fait cette représentation comporte de la redondance; en effet, au niveau L il y a 2^L noeuds représentés alors qu'en fait, pour une longueur de contrainte de K , il n'y a que 2^{K-1} états du codeur différents ($K \ll L$).



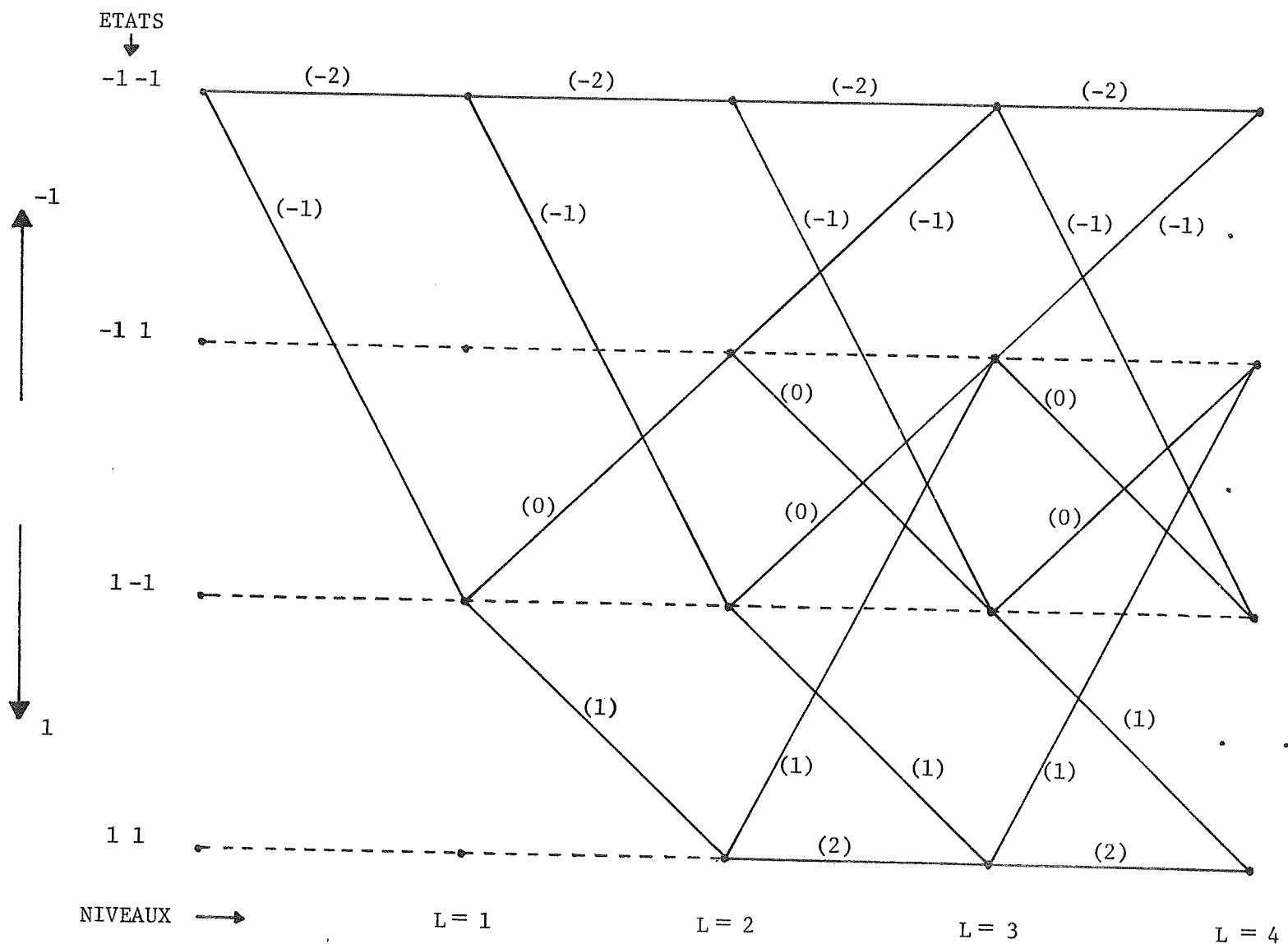
REPRÉSENTATION EN ARBRE

FIGURE 3.5

Une autre représentation décrivant les 2^L chemins possibles mais évitant d'introduire de la redondance est celle du treillis (Figure 3.6). Cette structure est dérivée de celle de l'arbre en reliant ensemble tous les noeuds désignant un même état à un niveau donné. Cette structure permet de limiter, pour un décodeur, le champ de recherche du chemin correct car, à chaque niveau, il n'y a que 2^{K-1} noeuds différents à considérer, soit un pour chacun des 2^{K-1} états différents.

Donc, selon ce que le décodeur utilise, une séquence d'entrée devra être décodée à partir du chemin le plus probable dans l'arbre ou le treillis défini par le code. Comme le mentionne Haccoun (1974) ce type de décodage dit "probabiliste", par opposition aux décodeurs "algébriques" qui se basent sur des propriétés de codes particuliers, est en général bien plus performant.

L'algorithme optimum de décodage est celui de Viterbi, (Viterbi 1967), (Forney 1973). Le décodeur correspondant analyse tous les chemins du treillis systématiquement et rend ainsi une décision optimale. Un chemin est éliminé le plus tôt possible quand il apparaît qu'il ne sera pas choisi lors de la décision finale. Pour cela, dans le cas binaire, le décodeur ne garde qu'un seul chemin survivant quand deux chemins convergent vers un même noeud. Pour déterminer lequel de ces deux chemins est à conserver une mesure de vraisemblance appelée "métrique" est utilisée. Le chemin le moins vraisemblable des deux peut être éliminé sans danger de nuire à l'optimalité du décodage.



REPRESENTATION EN TREILLIS

FIGURE 3.6

En effet, si deux chemins, \underline{U}_1 et \underline{U}_2 , convergent vers un même noeud à un certain niveau du treillis, et que \underline{U}_1 est plus vraisemblable que \underline{U}_2 (ayant une métrique supérieure), il est inutile de prolonger \underline{U}_2 au-delà de ce noeud de convergence car ses prolongations seront forcément identiques à celles de \underline{U}_1 ; \underline{U}_2 étant moins vraisemblable que \underline{U}_1 au noeud de convergence, il le sera plus loin dans le treillis à fortiori. Donc, à un niveau donné, 2^{K-1} comparaisons sont effectuées car, comme le montre la Figure 3.6, il y a deux chemins convergeant vers chacun des 2^{K-1} noeuds, et de ces comparaisons 2^{K-1} chemins survivants résultent. Cette opération se répète à chaque niveau jusqu'à la fin du treillis délivrant ainsi le chemin le plus vraisemblable de tous. La détermination de la métrique pour un chemin donné suit la règle de la vraisemblance maximale. Soit un message \tilde{m} pris dans un ensemble $\{M\}$ de messages équiprobables. Ce message peut être produit en une séquence de N symboles alimentant un codeur. Soit $1/v$ le taux de ce dernier; il en résulte donc une séquence codée $\underline{X}(\tilde{m})$ de Nv symboles, et celle-ci est transmise dans un canal avec du bruit. A la réception c'est une séquence \underline{Y} , comportant l'effet du bruit dans le canal, qui apparaît. En considérant une structure en treillis ou en arbre pour définir le codage, les symboles de $\underline{X}(\tilde{m})$ produits au I ième niveau, correspondant au I ième symbole de la séquence d'entrée au codeur, sont définis par,

$$\underline{x}_I(\tilde{m}) = \{ x_I^1(\underline{U}), x_I^2(\underline{U}) \dots x_I^v(\underline{U}) \} \quad (3.9)$$

où \underline{U} est le chemin qui a été suivi dans l'arbre ou le treillis pour arriver à ce niveau I . Les symboles correspondants de \underline{Y} sont donnés par,

$$\underline{y}_I = \{ y_I^1, y_I^2, \dots, y_I^v \} \quad (3.10)$$

Avec $P_{\underline{Y}}$ dénotant une densité de probabilité conditionnelle le décodeur produira comme résultat le message \hat{m} qui vérifie la condition suivante,

$$P(\hat{m}) P_{\underline{Y}}(\underline{Y}/\underline{X}(\hat{m})) \geq P(m) P_{\underline{Y}}(\underline{Y}/\underline{X}(m)) \quad \forall m \in \{ M \}, m \neq \hat{m} \quad (3.11)$$

$\{ M \}$ étant un ensemble de messages équiprobables, (3.11) se réduit à

$$P_{\underline{Y}}(\underline{Y}/\underline{X}(\hat{m})) \geq P_{\underline{Y}}(\underline{Y}/\underline{X}(m)) \quad \forall m \in \{ M \}, m \neq \hat{m} \quad (3.12)$$

Un récepteur maximisant $P_{\underline{Y}}(\underline{Y}/\underline{X}(\hat{m}))$ est appelé récepteur à vraisemblance maximale, et pour des m équiprobables, il minimisera la probabilité d'erreur (Wozencraft, Jacobs 1967). Pour un canal discret sans mémoire (le bruit discret affectant chaque symbole d'information indépendamment des autres), (3.12) devient, (avec P_y dénotant une densité de probabilité conditionnelle)

$$\prod_{j=1}^N P_y(y_j/x_j(\hat{m})) \geq \prod_{j=1}^N P_y(y_j/x_j(m)) \quad \forall m \in \{ M \} \quad (3.13)$$

ou encore,

$$\log \prod_{j=1}^N P(y_j/x_j(\hat{m})) \geq \log \prod_{j=1}^N P(y_j/x_j(m)) \quad \forall m \in \{ M \} \quad (3.14)$$

le logarithme étant monotone croissant. Donc, (3.14) devient finalement,

$$\sum_{j=1}^N \log P_y(y_j/x_j(\hat{m})) \geq \sum_{j=1}^N \log P_y(y_j/x_j(m)) \quad \forall m \in \{M\} \quad (3.15)$$

A partir de (3.15) la fonction de vraisemblance qu'est la métrique peut être définie; la métrique $\gamma_\ell(\hat{U})$ correspondant au ℓ ième niveau du chemin \hat{U} sera,

$$\gamma_\ell(\hat{U}) = \sum_{i=1}^v \log P_y(y_\ell^i/x_\ell^i(\hat{U})) \quad (3.16)$$

Au total, au niveau N , la métrique accumulée $\Gamma_N(\hat{U})$ sera,

$$\Gamma_N(\hat{U}) = \sum_{j=1}^N \gamma_j = \sum_{j=1}^N \sum_{i=1}^v \log P_y(y_j^i/x_j^i(\hat{U})) \quad (3.17)$$

Le décodeur cherche à maximiser $\Gamma_N(\hat{U})$ pour retrouver le chemin \hat{U} qui, le plus vraisemblablement possible, était celui défini par le message \tilde{m} qui a été initialement codé. Si le message \hat{m} est donné comme résultat par le décodeur est égal à \tilde{m} les altérations dues au bruit dans le canal auront été entièrement corrigées. Pour le cas de l'interférence entre symboles il a été montré dans la section 3.3 qu'un canal dispersif pouvait se modéliser comme un codeur convolutionnel de taux unitaire avec coefficients de pondération (Figure 3.3). La métrique accumulée totale devient alors,

$$\Gamma_N(\hat{U}) = \sum_{j=1}^N \log P_y(y_j/x_j(\hat{U})) \quad (3.18)$$

En se référant à la figure 2.5, il est apparent que

$$P_y(y_j/x_j(\hat{U})) = P_y(z_j + n_j/x_j(\hat{U})) \quad (3.19)$$

z_j étant entièrement déterministe il en résulte que,

$$P_y(y_j/x_j(\hat{U})) = P_n(n_j/x_j(\hat{U})) \quad (3.20)$$

où P_n est la densité de probabilité des n_j , les éléments discrets du bruit blanc et gaussien. Comme il n'y a aucune raison que le bruit aléatoire dépende de la séquence déterministe qu'est x , (3.20) devient,

$$P_y(y_j/x_j(\hat{U})) = P_n(n_j) = P_n(y_j - z_j) \quad (3.21)$$

ce qui donne comme métrique accumulée totale

$$\Gamma_N(\hat{U}) = \sum_{j=1}^N \log P_n(y_j - z_j) \quad (3.22)$$

Si la variance du bruit blanc et gaussien n_j est définie par $N_0/2$ et si, sans nuire à la validité du développement, sa moyenne est nulle, alors

$$P_n(y_j - z_j) = \frac{1}{\sqrt{N_0 \pi}} e^{-(y_j - z_j)^2 / N_0} \quad (3.23)$$

Et si des logarithmes népériens sont considérés (3.22) devient,

$$\Gamma_N(\hat{U}) = \sum_{j=1}^N \left\{ \log \left(\frac{1}{\sqrt{N_0 \pi}} \right) - \left(\frac{(y_j - z_j)^2}{N_0} \right) \right\} \quad (3.24)$$

N_o étant constant, maximiser $\hat{\Gamma}_N(\underline{U})$ revient à minimiser les termes $(y_j - z_j)^2$, c'est-à-dire, minimiser les distances euclidiennes entre les y_j et z_j correspondants.

Pour des codeurs convolutionnels normaux avec des additionneurs modulo-2 et des taux non unitaires, comme $1/v$, la conclusion ci-dessus est valable sauf que la sommation de (3.24) se fera sur Nv termes comme dans (3.17). De plus, les additionneurs modulo-2 assurant que les z_j restent binaires, les distances euclidiennes $(y_j - z_j)^2$ peuvent subir une quantification dure et devenir des distances de Hamming. Ceci est impossible dans la modélisation du canal dispersif, les z_j étant des valeurs réelles, et il faut s'en tenir aux distances euclidiennes.

Donc, en résumé, l'algorithme de Viterbi prend une décision optimale en étudiant tous les chemins possibles définis dans le treillis. A chaque niveau les reconvergences sont étudiées. Pour chaque état, une des deux branches y menant est éliminée; la transition associée correspond à un symbole z_{j1} plus distant du symbole y_j reçu à ce niveau que ne l'est le symbole z_{j2} correspondant à l'autre transition. Pour une longueur de contrainte de K il y a donc 2^{K-1} survivants qui sont retenus à chaque niveau et, pour une séquence d'entrée de N symboles le décodeur de Viterbi devra faire $N2^{K-1}$ calculs. L'effort de calcul croît exponentiellement avec la longueur de contrainte et c'est là que réside le désavantage majeur d'un tel décodeur.

3.5 Décodeur séquentiel

Une autre technique de décodage, non optimale mais n'ayant pas une croissance exponentielle de l'effort de calcul, est celle du décodage séquentiel (Jelinek 1969). Un décodeur séquentiel recherche le chemin correct dans l'arbre (Figure 3.5) en ne considérant que les chemins qui, au départ, semblent être les plus vraisemblables.

A partir du noeud d'origine dans l'arbre, le décodeur étudie les deux prolongations menant aux deux noeuds du niveau 1. Les métriques correspondantes sont comparées et le noeud de plus grande métrique est prolongé pour donner deux noeuds au niveau 2. Continuant alors jusqu'au niveau final de l'arbre le décodeur prolonge à chaque cycle le chemin le plus vraisemblable découlant des comparaisons avec tous les chemins qu'il a explorés aux niveaux précédents.

Contrairement à l'algorithme de Viterbi dont les comparaisons, pour des séquences binaires, n'impliquaient que deux chemins par niveau, le décodeur séquentiel doit constamment tenir compte de tous les chemins déjà explorés. Il pourra alors repérer les situations où le chemin qu'il prolonge s'avère être mauvais, sa métrique cumulative chutant brusquement, et ceci nécessitera un retour en arrière dans l'arbre pour retrouver le bon chemin. Le fait qu'un décodeur puisse s'engager sur un mauvais chemin pendant un certain temps peut être dû à des salves importantes de bruit dans le canal. Une fois ces salves dissipées le chemin étudié n'aura plus une métrique croissante mais,

au contraire, fortement décroissante.

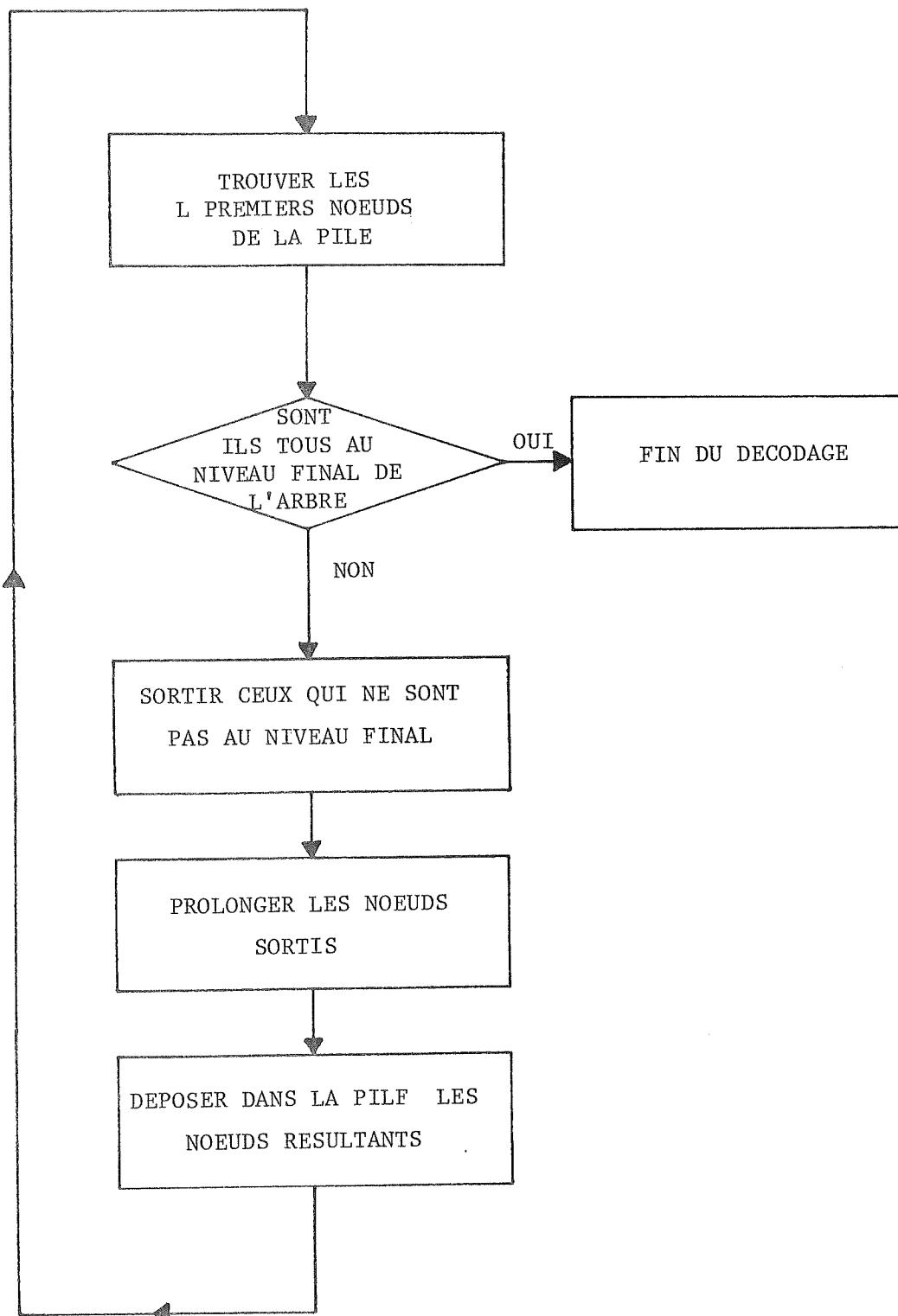
Les comparaisons du décodeur séquentiel se faisant pour des noeuds de profondeurs différentes, alors que pour le Viterbi les comparaisons étaient faites à la même profondeur, elles peuvent être rendues plus équitables en modifiant la métrique. Cette dernière est biaisée de telle sorte que le chemin correct aura une métrique qui, en général, augmentera avec la profondeur dans l'arbre. Des études (Fano 1963, Massey 1972) ont montré qu'en ajoutant à la métrique ordinaire développée dans la section précédente un terme correspondant au taux de codage l'opération du décodeur séquentiel était optimisée en minimisant les calculs de chemins incorrects. Pour les cas spéciaux des décodeurs combattant l'interférence entre symboles le biaisage de la métrique a été étudié dans ce travail de façon empirique lors des essais sur ordinateur présentés plus loin.

Il apparaît donc qu'un décodeur séquentiel, en moyenne, fournit un effort de calcul très inférieur à celui du Viterbi. En effet le décodeur séquentiel ne prolonge qu'un seul noeud à chaque niveau au lieu de 2^{K-1} dans le cas du Viterbi. Le désavantage du décodeur séquentiel est que cet effort de calcul n'est pas constant à cause des éventuels retours en arrière. Cette variabilité de l'effort de calcul nécessite en pratique l'utilisation d'un tampon d'entrée pour emmagasiner les symboles reçus du canal pendant que le décodeur effectue de longs retours en arrière. Occasionnellement, si cette recherche est

trop longue, le tampon déborde et de l'information est perdue. Pour parer à ce problème, des décodeurs se situant à mi-chemin entre le décodeur séquentiel et celui de Viterbi sont utilisés; ce sont les décodeurs séquentiels multi-chemins (Haccoun 1974). Ceux-ci prolongent plusieurs noeuds à chaque niveau au lieu d'un seul; à la limite ils peuvent prolonger jusqu'à 2^{K-1} chemins différents rejoignant ainsi le décodeur de Viterbi, mais généralement ce n'est pas nécessaire. Par exemple, pour un code de longueur de contrainte de six, un décodeur pourrait prolonger quatre chemins, au lieu de trente deux comme dans le cas du Viterbi. Les décodeurs multi-chemins peuvent éviter de faire de longs retours en arrière. En effet comme il a été déjà dit, un décodeur à chemin unique réalisant à un certain moment que le chemin qu'il prolonge n'est pas correct n'aura d'autres recours que de retourner en arrière; un décodeur quatre-chemins, par exemple, se trouvant devant la même situation examinera les trois autres chemins qu'il a prolongés et pourrait fort bien y retrouver le chemin correct sans avoir à effectuer de retours en arrière. Le bruit qui le forcera à rebrousser chemin lors de ses recherches devra être bien plus important que dans le cas du décodeur à chemin unique. La variabilité de calcul en est diminuée et les performances se rapprochent considérablement de celles du décodeur optimal. Des décodeurs multi-chemins variables ont été étudiés (Janelle et Haccoun 1978). Ceux-ci sont plus complexes, et leur fonctionnement consiste à ne prolonger qu'un petit nombre de chemins tant que le bruit est faible et à augmenter ce nombre dès que des tests témoins indiquent que le bruit est plus important.

Le décodeur séquentiel peut être décrit de façon claire, et facilement simulable sur ordinateur, par l'algorithme de Zigangirov-Jelinek, (Zigangirov 1966), (Jelinek 1969). Chaque noeud étudié par le décodeur est déposé dans une pile où l'information relative à la métrique, la profondeur, l'état associé et à d'autres paramètres nécessaires au décodage est conservée. Les noeuds sont disposés par ordre croissant de métrique dans la pile de sorte que le sommet contienne le noeud correspondant au chemin le plus vraisemblable. Pour prolonger un chemin, le noeud au sommet de la pile est retiré pour être remplacé par les deux noeuds qui résultent de son extension. Ces deux nouveaux noeuds sont ordonnés dans la pile selon leurs métriques. Le chemin s'est donc divisé en deux chemins supplémentaires. Le décodage est terminé lorsque le noeud du sommet de la pile se situe au niveau final; le noeud en question désigne le chemin qui a été jugé le plus vraisemblable par le décodeur.

Un décodeur multi-chemins peut être décrit également par cet algorithme. Pour un décodeur ℓ -chemins les ℓ premiers noeuds, en partant du sommet de la pile, sont retirés et remplacés par leurs prolongations à chaque cycle d'extension. Pour représenter l'algorithme de Viterbi il faut prolonger 2^{K-1} noeuds à partir du sommet en repérant les reconvergences à un même niveau. Des algorithmes prolongeant moins de 2^{K-1} noeuds mais repérant quand même les reconvergences peuvent également être représentés. L'organigramme d'un algorithme à pile pour des algorithmes multi-chemins, sans détection de reconvergence est montré à la figure 3.7.



ALGORITHME ZIGANGIROV-JELINEK
POUR UN DECODEUR L-CHEMINS

Figure 3.7

En pratique, pour simplifier l'ordonnance d'une pile, il est possible d'utiliser des classes d'équivalence de métrique appelées "sous-pile". Avec des sous-piles de taille H les noeuds dont la métrique se situe entre γ et $\gamma+H$ seront groupés dans une même sous-pile de façon arbitraire (dernier entré, premier sorti). Donc, pour inclure un nouveau noeud dans la pile il suffit de trouver la sous-pile à laquelle il appartient et de l'y insérer sans se soucier de son ordre de grandeur dans la sous-pile.

Des études complètes ont été réalisées pour l'adaptation du décodeur de Viterbi aux canaux dispersifs, (Magee (1972), Forney (1972a)). Ce ne fut pas le cas pour les décodeurs séquentiels; pourtant, les décodeurs séquentiels multi-chemins donnent d'excellentes performances, en usage normal, qui sont très proches de celles du décodeur de Viterbi, (Haccoun (1979), Janelle et Haccoun (1978)). Son application au problème de l'interférence entre symboles peut donc être envisagée avec optimisme, ce que confirment les essais au chapitre suivant.

L'avantage du décodeur séquentiel étant que son effort de calcul n'augmente pas avec la longueur de contrainte, il peut s'attaquer à des canaux très dispersifs causant une interférence entre symboles très étendue. Pour de tels canaux le décodeur de Viterbi devient vite inutilisable de par sa complexité. L'utilisation du décodeur séquentiel est donc toute indiquée. Celui-ci peut même remplacer le décodeur de Viterbi pour les cas à faibles longueurs de contraintes et obtenir des

performances très compétitives en faisant beaucoup moins de calculs.

Dans ce qui suit, des simulations de décodeurs séquentiels adaptés aux canaux dispersifs, avec des longueurs de contrainte allant jusqu'à sept, sont présentées. Un algorithme à pile multi-chemins, sans détection de reconvergence, est utilisé.

CHAPITRE 4

ANALYSE DES RESULTATS

4.1 Bornes de la probabilité d'erreur

Pour évaluer les résultats obtenus il est nécessaire d'étudier les bornes, inférieure et supérieure, de la probabilité d'erreur avec un canal dispersif. C'est essentiellement Forney (1972) puis Magee et Proakis (1973b) qui ont décrit les procédures à suivre pour l'obtention de ces bornes. Le travail se faisant avec des séquences finies il est de pratique courante de les représenter à l'aide d'une transformée polynomiale. La transformée en D explicite une séquence finie sous forme de polynôme en D , où D représente un élément de retard. Ce n'est en fait que la transformée en Z avec $D = Z^{-1}$. Avec cette notation une séquence $\underline{x} = \{x_0, x_1, \dots, x_n\}$ aura pour transformée en D le polynôme suivant:

$$X(D) = x_0 + x_1 D + x_2 D^2 + \dots + x_n D^n \quad (4.1)$$

Cette représentation est fréquemment utilisée dans les travaux traitant des probabilités d'erreur pour des transmissions avec des séquences discrètes et pour les relations entrée-sortie de codeurs.

Pour une transmission de signaux discrets il est bien connu, (Wozencraft et Jacobs 1965), que ces signaux peuvent être représentés comme des vecteurs dans un espace géométrique dont les axes mutuellement perpendiculaires correspondent à des fonctions de base φ_i .

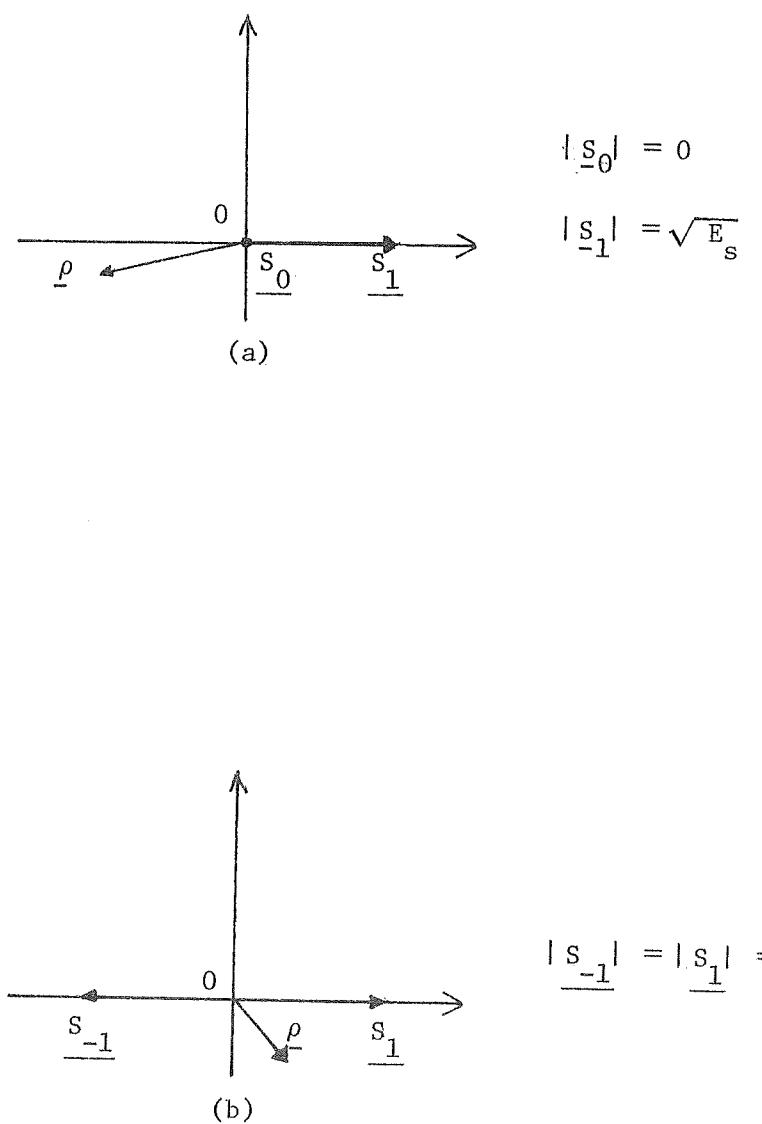
orthonormées. L'énergie d'un signal représenté par un vecteur \underline{s}_i sera alors donnée par $|\underline{s}_i|^2$. Pour une transmission binaire de deux bits équiprobables deux représentations possibles peuvent être étudiées. Si les bits sont choisis dans l'alphabet {0,1} une représentation adéquate est celle de la figure 4.1a. Le vecteur nul \underline{s}_0 correspond à la transmission d'un 0 tandis que \underline{s}_1 , de module $\sqrt{E_s}$, correspond à la transmission d'un 1. Donc de l'énergie est dépensée uniquement lors de la transmission d'un 1. La distance entre les deux signaux est alors définie par,

$$d_{10} = |\underline{s}_1 - \underline{s}_0| = \sqrt{E_s} \quad (4.2)$$

Les bits auraient pu être choisis dans l'alphabet {-1,1}. Ceci est décrit par les signaux antipodaux de la figure 4.1b. \underline{s}_{-1} et \underline{s}_1 définissent la transmission d'un -1 et d'un 1 respectivement et sont deux vecteurs opposés par rapport à l'origine. Cette fois-ci les deux bits sont transmis avec une énergie $\sqrt{E_s}$ et la distance entre les deux est donnée par

$$d_{1-1} = |\underline{s}_1 - \underline{s}_{-1}| = 2\sqrt{E_s} \quad (4.3)$$

Cette distance est donc deux fois plus grande que dans le cas précédent. A la réception un symbole reçu peut également être représenté dans l'espace des signaux; c'est le vecteur $\underline{\rho}$ représenté à la figure 4.1. Le symbole reçu est entâché de bruit et peut alors différer grandement des symboles émis.



REPRESENTATION DES SIGNAUX DANS UN
ESPACE GEOMETRIQUE

FIGURE 4.1

Pour des signaux équiprobables en présence de bruit blanc la règle de décision est la minimisation de $|\underline{\rho} - \underline{s_i}|^2$. Donc pour le cas de la figure 4.1b une décision en faveur de 1 se fera à la réception si

$$|\underline{\rho} - \underline{s_{-1}}|^2 > |\underline{\rho} - \underline{s_1}|^2 \quad (4.4)$$

En règle générale plus grande est la distance entre les deux signaux à l'émission, plus faible sera la probabilité d'erreur à la réception. Le prix à payer se traduit par l'énergie dépensée lors de la transmission des signaux.

Quand une séquence de bits est transmise deux probabilités d'erreur peuvent être considérées; la probabilité d'erreur de séquence ou la probabilité d'erreur de bit. Une séquence de bits est en erreur si tous les bits la formant ne sont pas détectés correctement. La probabilité d'erreur de séquence est donnée par,

$$P_S(\ominus) = 1 - P[C] \quad (4.5)$$

où $P[C]$ est la probabilité que toute la séquence soit détectée correctement. Donc si une séquence de 10,000 bits est transmise il suffit qu'un seul de ces bits soit mal détecté pour que la séquence reçue soit erronée. Par contre la probabilité d'erreur de bit, pour des séquences suffisamment longues, est donnée par le rapport de bits en erreur avec le nombre total de bits formant la séquence. Donc si 500 bits sont en erreur dans une séquence de 10,000 bits la probabilité

d'erreur de bit sera égale à 0.05. Pour une séquence de données constituée de n bits, la probabilité d'erreur de bit se trouve à partir de celle de séquence comme suit; (Lindsey et Simon 1973)

$$P_B(\epsilon) = \frac{2^{n-1}}{2^n - 1} P_S(\epsilon) \quad (4.6)$$

Soit la séquence $X(D)$, formée de symboles binaires, transmise à travers un canal dispersif, et soit $\hat{X}(D)$ la séquence décodée par le récepteur; si $\hat{X}(D)$ diffère de $X(D)$ des erreurs auront été commises et une séquence d'erreur ramenée au niveau de l'entrée, est donnée par

$$\epsilon_{\hat{X}}(D) = X(D) - \hat{X}(D) \quad (4.7)$$

Si l'alphabet binaire de $X(D)$ et $\hat{X}(D)$ est constitué de 0 et de 1 une séquence d'erreur non-nulle sera formée de +1 et de -1. Par contre avec l'alphabet {-1, 1} la séquence d'erreur sera formée de +2 et de -2. Par exemple:

$$X(D) = -1 + D - D^2 + D^3$$

$$\hat{X}(D) = -1 + D + D^2 - D^3$$

$$\rightarrow \epsilon_{\hat{X}}(D) = -2D^2 + 2D^3 \quad (4.8)$$

Ces 2 et -2, (1 et -1 dans l'alphabet {0 1}) représentent des unités de distance séparant une séquence erronée d'une séquence correcte.

Le canal de transmission, linéaire et dispersif, représenté à la figure 2.1 peut également être décrit par la transformée en D comme suit;

$$F(D) = f_0 + f_1 D + f_2 D^2 + \dots \quad (4.9)$$

Par abus de langage l'énergie d'un tel canal est définie par $\sum f_i^2$. Les relations entrée-sortie pour le canal $F(D)$ peuvent s'écrire,

$$Z(D) = X(D) F(D) \quad (4.10)$$

et la séquence d'erreur à la sortie du canal sera

$$\begin{aligned} \epsilon_Z(D) &= \epsilon_X(D) F(D) \\ &= e_{Z0} + e_{Z1} D + e_{Z2} D^2 + \dots \end{aligned} \quad (4.11)$$

Si cette séquence est de longueur n sa distance euclidienne sera donnée par,

$$d(\epsilon_Z) = \| \epsilon_Z \| = \sum_{i=0}^{n-1} e_{Zi}^2 \quad (4.12)$$

En calculant la distance minimum,

$$d_{\min} = \min_{\epsilon_Z} \{ d(\epsilon_Z) \} \quad (4.13)$$

les bornes de la probabilité d'erreur de séquence avec un bruit gaussien d'écart type σ , s'obtiennent comme suit, (Forney 1972),

$$P_S(\epsilon) \geq K_1 Q \left[\frac{d_{\min}}{2\sigma} \right] \quad K_1 = \text{constante} \quad (4.14)$$

$$P_S(\epsilon) \leq K_2 Q \left[\frac{d_{\min}}{2\sigma} \right] \quad K_2 = \text{constante} \quad (4.15)$$

$$\text{avec } Q(y) = \sqrt{\frac{1}{2\pi}} \int_y^{\infty} e^{-z^2/2} dz$$

En règle générale la probabilité qu'un signal \underline{s}_i soit confondu, à la réception, avec un signal \underline{s}_k équiprobable est donnée par $Q \left[\frac{d_{ik}}{2\sigma} \right]$ où d_{ik} est la distance entre ces deux signaux (Wozencraft et Jacobs 1965). Pour le calcul des bornes c'est la plus petite distance qu'il puisse y avoir entre deux signaux qui compte.

Ces bornes servent à évaluer les performances des récepteurs optimaux se basant, par exemple, sur des algorithmes de décodage séquentiel ou de Viterbi; les bornes, supérieure et inférieure, étant proportionnelles, ceci implique l'optimalité de telles techniques de décodage. En effet si la borne inférieure tend vers 0 la borne supérieure tendra également vers 0.

Le problème réside dans l'obtention de d_{\min} et dans la détermination des constantes K_1 et K_2 .

Si le canal n'a pas d'interférence entre symboles sa fonction de transfert sera

$$F(D) = f_0 \quad (4.16)$$

La distance minimum au carré sera alors obligatoirement proportionnelle à l'énergie du canal; si les symboles émis sont d'énergie unitaire, ($E_S = 1$), d_{\min}^2 sera donc donné par,

$$d_{\min}^2 = kf_0^2 \quad (4.17)$$

Pour l'alphabet $\{0,1\}$ $k=1$, car les unités de distance à l'entrée du canal se traduisent par des ± 1 . Pour l'alphabet $\{-1, 1\}$ $k=4$, les unités de distance étant données par ± 2 . Ceci est compatible avec les équations (4.2) et (4.3).

Pour un canal ayant une fonction de transfert de longueur 2, soit,

$$F(D) = f_0 + f_1 D \quad (4.18)$$

la distance minimum sera encore une fois déterminée à partir de l'énergie du canal;

$$d_{\min}^2 = k \{ f_0^2 + f_1^2 \} \quad (4.19)$$

($k = 1$ ou 4 selon l'alphabet)

En effet pour n'importe quelle séquence d'erreur $\epsilon_X(D)$ non nulle à l'entrée, la séquence $\epsilon_Z(D)$ de sortie débutera forcément par $\sqrt{k}f_0$ et finira par $\sqrt{k}f_1$. La distance minimum se trouvera alors pour le cas où les composantes intermédiaires de $\epsilon_Z(D)$ sont nulles.

Pour des canaux de longueur 3 ou plus, l'obtention de d_{\min}^2 est moins immédiate. Il a été montré (Magee 1972) qu'en utilisant des séquences d'erreur $\in_X(D)$ de la forme $\pm(1-D)$, d_{\min} pouvait être obtenu, et qu'il était inutile de considérer des séquences plus longues ou de plus haut degré, comme $(1-D+D^2)$ ou $(1-D^2)$.

Soit un canal de longueur 3, décrit par

$$F(D) = f_0 + f_1 D + f_2 D^2 \quad (4.20)$$

avec

$$\in_X(D) = \sqrt{k} (1-D) \quad (4.21)$$

il est obtenu

$$\begin{aligned} \in_Z(D) &= \in_X(D)F(D) = \sqrt{k} (1-D)(f_0 + f_1 D + f_2 D^2) \\ &= \sqrt{k} [f_0 + (f_1 - f_0)D + (f_2 - f_1)D^2 - f_2 D^3] \end{aligned} \quad (4.22)$$

et donc

$$d^2 = k [f_0^2 + (f_1 - f_0)^2 + (f_2 - f_1)^2 + f_2^2] \quad (4.23)$$

Il en va de même pour les canaux de longueurs supérieures à 3, les distances étant obtenues à l'aide de la séquence d'erreur $\pm(1-D)$ comme dans les équations (4.22) et (4.23).

Pour compléter le calcul de la borne les constantes K_1 et K_2 restent à évaluer. Forney (1972b) a démontré que K_1 se traduit par la probabilité qu'une séquence d'entrée et une séquence d'erreur $\in_X(D)$ en

s'ajoutant donnent, ramenées à l'entrée du canal, une séquence $X'(D)$ qui peut être prise pour une séquence d'entrée valable. En effet, même avec un détecteur parfait qui ramènerait toute séquence entachée de bruit à la séquence d'entrée valable correspondante, il y aura toujours la possibilité d'une confusion entre $X'(D)$ et $X(D)$, les deux étant des séquences valables. En considérant des séquences d'erreur du type $\pm (1-D)$, $X'(D)$ et $X(D)$ seront séparés, au minimum, par une distance d_{\min} . La probabilité K_1 qu'un tel cas se produise, donnant au maximum une probabilité $Q(\frac{d_{\min}}{2\sigma})$ de confusion, se trouve facilement en considérant toutes les séquences d'entrée de longueur 2 ajoutées à des séquences d'erreur de type $\pm (1-D)\sqrt{k}$. Par exemple, soit l'alphabet d'entrée $\{+1, -1\}$; alors, $\sqrt{k}=2$ et,

$$X(D) = 1 + D \Rightarrow \begin{cases} X'(D) = X(D) + 2-2D = 3-D \\ X'(D) = X(D) - 2+2D = -1+3D \end{cases} \quad (4.24)$$

$$X(D) = 1 - D \Rightarrow \begin{cases} X'(D) = X(D) + 2-2D = 3-3D \\ X'(D) = X(D) - 2+2D = -1+D \end{cases} \quad (4.26)$$

$$X(D) = -1 + D \Rightarrow \begin{cases} X'(D) = X(D) + 2-2D = 1-D \\ X'(D) = X(D) - 2+2D = -3+3D \end{cases} \quad (4.28)$$

$$X(D) = -1 - D \Rightarrow \begin{cases} X'(D) = X(D) + 2-2D = 1-3D \\ X'(D) = X(D) - 2+2D = -3+D \end{cases} \quad (4.30)$$

Il apparaît donc que les cas de confusion, compatibles avec l'alphabet $\{+1, -1\}$ sont donnés par les équations (4.27) et (4.28), donc pour deux séquences d'entrée parmi quatre possibles. En effet, seules ces séquences peuvent être considérées comme étant des séquences transmises.

Alors,

$$K_1 = \frac{1}{2} \quad (4.32)$$

Pour évaluer K_2 il faut considérer la pire situation possible qui se reflète par un canal générant une chaîne continue d'erreurs sans augmenter la distance minimum. Pour ce type de canal Forney (1972) a montré que

$$K_2 = 2m(m-1) \quad (4.33)$$

où m est le nombre de niveaux dans l'alphabet. Pour un alphabet binaire $m=2$ et,

$$K_2 = 4 \quad (4.34)$$

Les bornes peuvent maintenant être calculées à partir des équations (4.14) et (4.15) ce qui donne pour le cas binaire,

$$\frac{1}{2} Q \left(\frac{d_{\min}}{2\sigma} \right) < P_S < 4 Q \left(\frac{d_{\min}}{2\sigma} \right) \quad (4.35)$$

Cette paire de bornes est très serrée, surtout si $\left(\frac{d_{\min}}{2\sigma} \right) \gg 1$. De plus, ces bornes ayant été calculées pour des séquences d'erreur de longueur 2 il est visible, d'après l'équation (4.6) avec $n=2$, qu'un rapport de $2/3$ est à considérer pour passer de la probabilité d'erreur de séquence à celle de bit.

4.2 Relation entre la distance minimum et la perte provoquée par un canal dispersif

En règle générale l'interférence entre symboles peut provoquer une perte de performance, par rapport au cas sans interférences, en diminuant d_{\min} . Pour évaluer cette perte il est utile de considérer des canaux de même énergie. Soit un canal d'énergie unitaire ne provoquant pas d'interférence entre symboles,

$$F(D) = f_0^2 \text{ avec } f_0^2 = 1 \quad (4.36)$$

et un canal d'énergie unitaire à interférence de longueur 3;

$$F_i(D) = f_{0_i}^2 + f_{1_i}^2 + f_{2_i}^2 \quad (4.37)$$

$$\text{avec } f_{0_i}^2 + f_{1_i}^2 + f_{2_i}^2 = 1 \quad (4.38)$$

Avec des bits d'énergie unitaire, $F(D)$ produit, d'après l'équation (4.17), une distance minimum de \sqrt{k} , tandis que $F_i(D)$, d'après l'équation (4.23), produit une distance minimum $d_{i\min}$, avec,

$$d_{i\min}^2 = k [f_{0_i}^2 + (f_{1_i} - f_{0_i})^2 + (f_{2_i} - f_{1_i})^2 + f_{2_i}^2] \quad (4.39)$$

(4.39) est une équation quadratique qui peut se mettre sous forme matricielle comme suit,

$$d_{i\min}^2 = k \underline{f_i^T} A \underline{f_i} \quad (4.40)$$

où $\underline{f_i}$ est un vecteur colonne ayant pour composantes les coefficients f_{j_i} et A est une matrice (3x3) définie positive qui dépend de la séquence d'erreur considérée. Avec l'équation (4.39), déterminée à partir d'une séquence \sqrt{k} (1-D), A est trouvé comme suit,

$$\begin{bmatrix} f_{0_i} & f_{1_i} & f_{2_i} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} f_{0_i} \\ f_{1_i} \\ f_{2_i} \end{bmatrix} = f_{0_i}^2 + (f_{1_i} - f_{0_i})^2 + (f_{2_i} - f_{1_i})^2 + f_{2_i}^2 \quad (4.41)$$

Par identification il est trouvé que A est la matrice symétrique suivante,

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix} \quad (4.42)$$

A partir de cela il est possible de trouver le canal de longueur 3 provoquant la perte maximum par rapport au canal d'énergie unitaire sans interférence. Il faut trouver le vecteur $\underline{f_i}$ qui minimise l'équation (4.40) sous la contrainte de l'équation (4.38). Il est montré (Wylie 1966) que $\underline{f_i}$ est alors un vecteur propre correspondant à la valeur propre minimum de A. De plus cette valeur propre minimum n'est autre que d_i^2 min à k près. Les valeurs propres γ_i de la matrice A, ainsi que les vecteurs propres correspondants, se trouvent comme suit

$$\underline{A}\underline{X} = \lambda \underline{X} \quad (4.43)$$

où λ , la valeur propre, est un scalaire et \underline{X} le vecteur propre correspondant

$$|\underline{A} - \lambda \underline{I}| = 0 \quad (4.44)$$

donc

$$\begin{vmatrix} (2-\lambda) & -1 & 0 \\ -1 & (2-\lambda) & -1 \\ 0 & -1 & (2-\lambda) \end{vmatrix} = 0$$

$$\Rightarrow \lambda_1 = 2 \quad \lambda_2 = 3.414 \quad \lambda_3 = 0.5858 \quad (4.45)$$

et alors

$$d_i^2 \min = k(0.5858)$$

et le vecteur propre correspondant est donné par

$$\begin{bmatrix} (2-\lambda_{\min}) & -1 & 0 \\ -1 & (2-\lambda_{\min}) & -1 \\ 0 & -1 & (2-\lambda_{\min}) \end{bmatrix} \begin{bmatrix} f_{0i} \\ f_{1i} \\ f_{2i} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (4.46)$$

ce qui donne deux équations indépendantes,

$$1.414 f_{0i} - f_{1i} = 0 \quad (4.47)$$

$$- f_{1i} + 1.414 f_{2i} = 0$$

En considérant également l'équation (4.38) il est obtenu que

$$f_{0_i} = 0.5, \quad f_{1_i} = 0.71, \quad f_{2_i} = 0.5 \quad (4.48)$$

La perte est alors définie comme étant le rapport au carré des distances résultant du canal sans interférence et de celui avec interférence. Pour des canaux à énergie unitaire et avec $k = 1$,

$$\begin{aligned} \text{PERTE(dB)} &= 10 \log \left[\frac{k}{d_{\min}^2} \right] \\ &= -10 \log [0.5858] \\ &= 2.3 \text{ dB} \end{aligned} \quad (4.49)$$

Magee et Proakis (1973 b) ont répété cette procédure pour des canaux allant jusqu'à des longueurs de 10 et les résultats obtenus sont montrés au tableau 1, avec $k = 1$. Ces canaux normalisés sont ceux provoquant la perte maximum par rapport à un canal normalisé de longueur 1. Il est à remarquer que pour un canal d'énergie unitaire de longueur 2 la perte est nulle, car, comme il a déjà été montré, d_{\min}^2 est alors directement proportionnelle à l'énergie du canal.

Pour la simulation les canaux peuvent être choisis parmi ceux du tableau 1 et les d_{\min} correspondants dans les expressions des pertes servent au calcul des bornes, en tenant compte de k . Pour des canaux d'énergie non unitaire d_{\min} est trouvé à partir des équations (4.11) et (4.12) et le calcul des bornes peut alors s'effectuer.

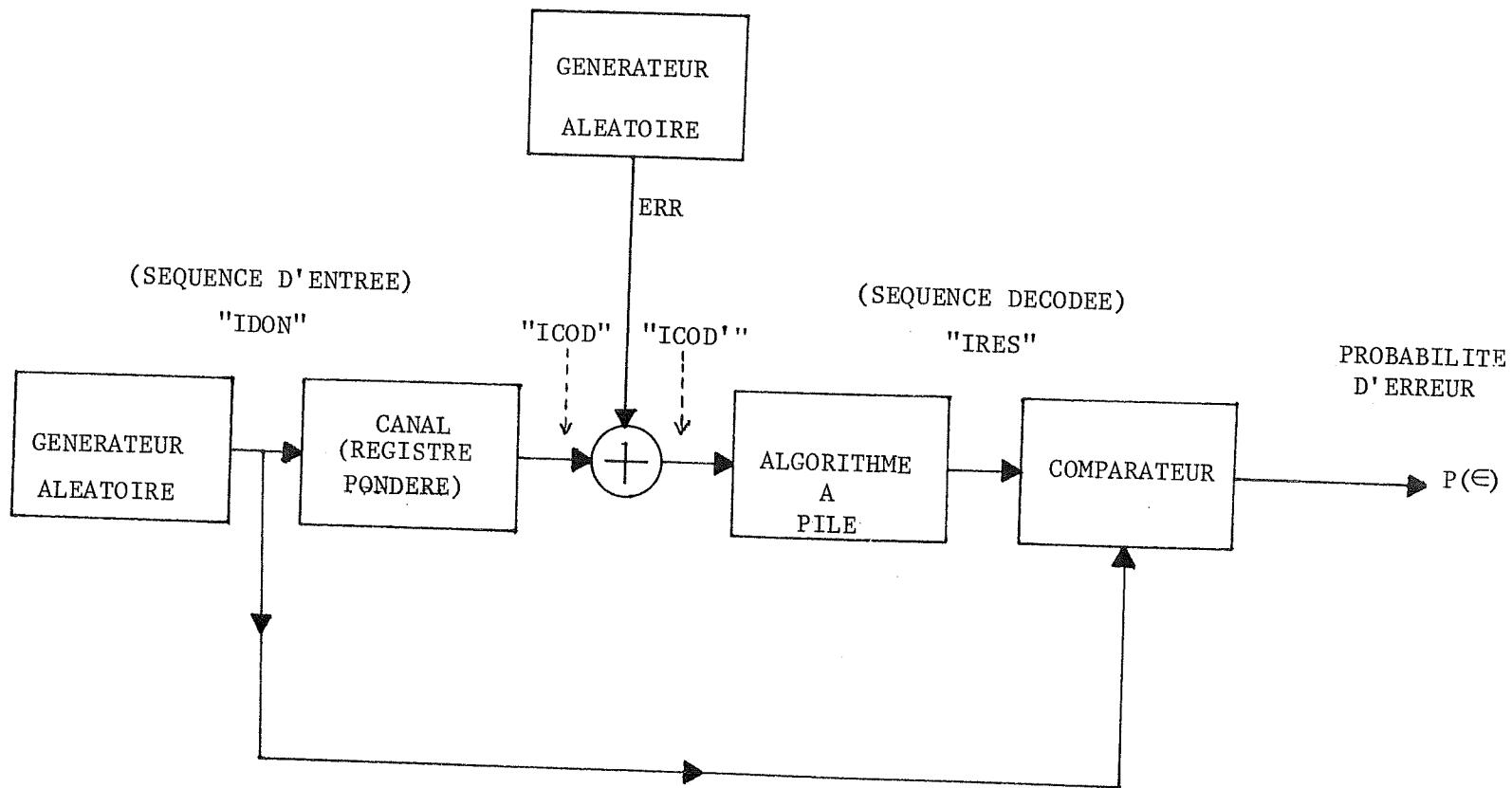
Longueur du canal	Pertes (dB) ($-10 \log_{10} d_i^2 \min$)	Canal à distance minimisée (avec $k = 1$)
2	0	
3	2.3	(0.5, 0.71, 0.5)
4	4.2	(0.38, 0.60, 0.60, 0.38)
5	5.7	(0.29, 0.50, 0.58, 0.50, 0.29)
6	7.0	(0.23, 0.42, 0.52, 0.52, 0.42, 0.23)
7	8.2	(0.19, 0.35, 0.46, 0.50, 0.46, 0.35, 0.19)
8	9.2	(0.16, 0.30, 0.41, 0.46, 0.46, 0.41, 0.30, 0.16)
9	10.1	(0.14, 0.26, 0.36, 0.43, 0.45, 0.43, 0.36, 0.26, 0.14)
10	10.9	(0.12, 0.23, 0.32, 0.39, 0.42, 0.42, 0.39, 0.32, 0.23, 0.12)

TABLEAU 1

4.3 Simulations

Les simulations de canaux dispersifs ont été réalisées à l'aide d'un programme Fortran. Le principe de ce dernier est décrit par le schéma bloc de la figure 4.2.

Une séquence IDON dont les éléments proviennent de l'alphabet $\{-1, +1\}$ est créée à partir d'un générateur de nombres aléatoires. Le canal dispersif est simulé à partir du modèle développé précédemment. Un registre à décalage est défini. Ses cellules sont initialement remplies de -1 et elles peuvent être pondérées par des coefficients. Avec IDON comme entrée le canal produit une séquence ICOD de même longueur qui représente l'effet de l'interférence entre symboles sur IDON. Un second générateur de nombres aléatoires produit alors une séquence toujours de même longueur, ERR, représentant un bruit blanc gaussien et qui dépend du rapport signal sur bruit considéré. Cette séquence s'ajoute à la séquence ICOD; le n -ième élément de ERR est un échantillon discret de bruit gaussien, indépendant de ses voisins, et il s'ajoute au n -ième élément de ICOD. Le résultat est une séquence ICOD' qui tient compte de l'interférence entre symboles et du bruit aléatoire. Le canal ayant été complètement simulé le programme passe dans la phase de récupération de la séquence d'entrée. La séquence ICOD' entre dans un détecteur défini par l'algorithme à piles d'un décodeur séquentiel multi-chemins. La séquence décodée IRES est comparée avec la séquence originale IDON qui a été gardée en mémoire.



SCHEMA BLOC DE LA SIMULATION

FIGURE 4.2

Une probabilité d'erreur de bit est calculée en faisant le rapport entre le nombre de symboles d'IRES qui diffèrent de leurs correspondants dans IDON et le nombre total de symboles composant ces séquences. Plusieurs essais sont effectués en variant le rapport signal sur bruit pour un canal donné. Une courbe représentant la probabilité d'erreur de bit en fonction du rapport signal sur bruit est alors tracée. Celle-ci est comparée avec les bornes calculées au début du chapitre et qui sont également tracées sur le même graphique. La performance du détecteur est ainsi évaluée.

4.4 Rapport signal sur bruit

Le rapport signal sur bruit, SNR, considéré pour la simulation est le même que celui utilisé par Magee (1972) pour ses essais avec le décodeur de Viterbi. Les résultats peuvent alors être facilement comparés avec ceux obtenus ici. Pour un bruit blanc de variance $\frac{N_o}{2}$ le rapport SNR est défini par

$$\text{SNR} = \frac{E_b}{N_o} \quad (4.50)$$

où E_b est l'énergie utilisée pour transmettre un bit d'information. Cette énergie peut être associée à la variance α_x^2 des symboles binaires équiprobables produits pour la séquence IDON par le premier générateur de nombres aléatoires (Magee 1972). Si α_N^2 dénote la variance du bruit produit par le deuxième générateur aléatoire l'équation (4.50) devient alors

$$\text{SNR} = \frac{\sigma_X^2}{2\sigma_N^2} \quad (4.51)$$

L'alphabet $\{-1, +1\}$ pour la séquence d'entrée IDON définit des symboles antipodaux d'énergie unitaire. En effet, ces symboles ont une moyenne nulle et une variance donnée par

$$\sigma_X^2 = (-1)^2 \frac{1}{2} + (1)^2 \frac{1}{2} = 1 \quad (4.52)$$

l'équation (4.51) devient donc

$$\text{SNR} = \frac{1}{2} \quad (4.53)$$

la valeur en décibels de SNR est calculée,

$$(\text{SNR})_{\text{dB}} = 10 \log_{10} \text{SNR} \quad (4.54)$$

et c'est cette valeur qui est portée en abscisse pour les graphiques donnant les probabilités d'erreur de bit.

4.5 Générateur de nombres aléatoires

Le générateur aléatoire Fortran suivant (IBM STAFF 1959) est utilisé. Une valeur entière et impaire IX est donnée.

```

5 IY = IX * 65539
     IF (IY) 10,20,20
10 IY = IY + 2147483647+1
20 YFL = IY
30 YFL = YFL * 0.4656613E-9
40 IX = IY

```

Au niveau de l'énoncé 30, YFL est une valeur réelle uniformément distribuée sur l'intervalle (0,1). Une valeur de l'X à l'énoncé 40 peut être renvoyée à l'énoncé 5 ce qui donnera une autre valeur entre 0 et 1 pour YFL; le générateur peut donc être employé de façon récursive. Pour produire la séquence IDON il suffit d'actionner ce générateur autant de fois que le nombre total de symboles désirés dans IDON et de produire un -1 chaque fois que YFL se trouve dans l'intervalle (0, 0.5) et un +1 quand YFL se trouve dans l'intervalle (0.5, 1).

Un générateur identique est utilisé pour produire ce bruit aléatoire. Pour rendre le bruit approximativement gaussien le générateur est actionné plusieurs fois et les YFL produits sont ajoutés l'un à l'autre. Pour cette simulation, le générateur est actionné douze fois pour chaque symbole de bruit. Les valeurs de YFL étant uniformément distribuées entre 0 et 1 leur moyenne est de 0.5. Pour la somme de douze valeurs de YFL la moyenne sera donc de 6. Un symbole ERR_i de bruit gaussien est alors produit comme suit,

$$ERR_i = MOY + \sigma_N (SOM - 6.0) \quad (4.55)$$

où MOY est la valeur moyenne désirée pour un symbole de bruit et SOM est la somme des douze valeurs de YFL. Cette opération est répétée pour chaque symbole de la séquence ERR définissant ainsi un bruit blanc gaussien discret de moyenne MOY et d'écart type σ_N . Cette procédure pour la génération du bruit gaussien est encore une fois identique à celle qu'utilise Magee (1972) ce qui permet de comparer ses

résultats avec ceux obtenus dans ce travail. Pour les simulations le bruit est supposé de moyenne nulle, ce qui fixe MOY à 0.

4.6 Structure du programme Fortran

A partir d'un programme original conçu par Haccoun (1974), et modifié par Dufour (1980), décrivant un algorithme à piles pour décodeur séquentiel, de nombreux changements ont été réalisés pour l'adapter au problème de l'interférence entre symboles. Il a fallu, entre autre, créer une sous-routine qui simule l'effet d'un canal dispersif sur une séquence discrète, qui établit les nouvelles valeurs possibles des symboles de cette séquence suite à l'interférence et qui définit un diagramme d'état approprié du genre de celui représenté à la figure 3.4 : pour un canal de longueur 3. Les seules routines qui n'ont pas été modifiées dans le programme original sont celles décrivant l'extraction ou l'ajout d'un noeud dans une pile. Le programme ainsi modifié peut simuler des canaux dispersifs ayant des longueurs de contrainte comprises entre 3 et 7. Le nombre de symboles pour la séquence d'entrée est au choix de l'utilisateur. Pour alléger le fardeau de calculs le nombre total de symboles requis est divisé en blocs plus petits qui sont transmis l'un à la suite de l'autre. Une réinitialisation des variables, autres que celles concernant les générateurs aléatoires, s'effectue avant la transmission de chaque bloc. Dans les essais présentés ici la séquence d'entrée comprend 10,000 symboles sous la forme d'une transmission de vingt blocs de 500

symboles ou de vingt-cinq blocs de 400 symboles.

Programme Principal

Le programme principal est essentiellement le fil conducteur de la simulation suivant le schéma bloc de la figure 4.2. Après la déclaration des variables le programme va lire les données nécessaires à la simulation. Celles-ci sont fournies par l'utilisateur dans l'ordre suivant:

1ère carte de donnée: (3 éléments)

Le nombre de symboles par blocs de la séquence de donnée, la dimension de la pile, et le nombre maximum de sous-piles, sont donnés.

Ces trois valeurs sont données en format I6.

2e carte de donnée: (2 éléments)

La mémoire du codeur ainsi que celle du décodeur sont spécifiées en format I2. La mémoire est définie comme étant la longueur de contrainte moins un. Habituellement les deux valeurs données sont identiques.

3e carte de donnée: (7 éléments)

Les coefficients du canal dispersif sont spécifiés en format

F8.4. Pour des canaux de longueurs de contrainte inférieures à 7 les coefficients sont ajustés le plus à droite possible et les espaces de gauche non utilisés sont remplis de zéros.

4e et 5e cartes de données: (1 élément chacune)

Les deux valeurs de départ des générateurs aléatoires sont spécifiées, une sur chaque carte. Ces valeurs doivent être impaires. Le format utilisé est I15.

6e carte de donnée: (1 élément)

On spécifie le nombre total de blocs formant la séquence de données en format I3.

7e carte de donnée: (1 élément)

Le nombre de chemins explorés par le décodeur séquentiel est spécifié en format I3.

Cartes subséquentes: (2 éléments chacune)

Chacune de ces cartes spécifie un bruit additif gaussien à l'aide de l'écart type et la moyenne de ce dernier. Ces valeurs sont données en Format F5.2. Il y a autant de cartes que de simulations requises.

La séquence de données est alors produite. A la fin de celle-ci une suite de symboles, tous fixés à -1, sont ajoutés. Cette suite est appelée la queue de la séquence. Le nombre de symboles formant cette queue est égal à la mémoire du codeur. Cette queue permet au décodeur de prendre des décisions exactes à la fin d'une séquence reçue ce qui lui donne plus de chance de retrouver le bon chemin dans l'arbre de la figure 3.5. Le programme fait alors appel à la sous-routine REGIS qui simule l'interférence entre symboles et définit un diagramme d'état. De retour au programme principal un bruit aléatoire s'ajoute complétant ainsi la simulation du canal. La phase de décodage d'un bloc commence. A partir de la séquence reçue un décodage suivant l'algorithme à piles multi-chemins s'effectue. Le chemin le plus vraisemblable est recherché dans l'arbre. Chaque fois qu'un noeud est déposé dans une sous-pile l'information concernant sa métrique, son niveau dans l'arbre, l'état correspondant, ainsi que le noeud du niveau précédent dont il a été issu est retenue. De cette façon, arrivé au niveau final et après avoir considéré la queue de symboles fixés, le décodeur peut, à partir du noeud de métrique maximum, retrouver tous les noeuds aux niveaux précédents qui ont mené jusqu'à lui. Cette succession de noeuds définit le chemin dans l'arbre qui a été jugé le plus vraisemblable. Ce chemin définit à son tour une suite de transitions dans le diagramme d'états établi par la sous-routine REGIS, ce qui permet au décodeur de donner une séquence de +1 et de -1 correspondant à ces transitions. Une comparaison avec la séquence d'entrée originale s'effectue alors pour

déterminer le nombre de symboles en erreur subsistant après le décodage et une probabilité d'erreur de bit est calculée. Lors du décodage le programme principal fait appel à la sous-routine DEVLOP qui détermine à partir d'un noeud à un niveau donné ses deux successeurs au prochain niveau et leur assigne à chacun une métrique. Le programme principal fait aussi appel à la sous-routine AJOUT qui ajoute un noeud dans la sous-pile appropriée, à la sous-routine CHERSP qui cherche la plus haute sous-pile non-vide qui contient les noeuds de plus hautes métriques, ainsi que la sous-routine CHERNO qui extrait le meilleur noeud d'une sous-pile. Finalement, le décodage terminé, le programme principal imprime les résultats. Des exemples de ceux-ci sont donnés à l'appendice.

4.7 Choix de la métrique

La métrique utilisée correspond à la métrique théorique développée au chapitre 3. Avec un noeud au niveau $i-1$ le décodeur passe au niveau i en considérant les deux transitions possibles d'après le diagramme d'état. A chacune de ces transitions correspond un symbole 'codé' qui tient compte de l'interférence entre symboles. Le décodeur calcule alors les distances euclidiennes au carré entre le i -ième symbole qu'il a reçu et chacun de ces deux symboles codés. Ces distances fixent les métriques cumulatives des deux nouveaux noeuds au niveau i . Comme cela a été montré dans le chapitre 3, maximiser une métrique cumulative d'un noeud à un niveau i revient à choisir les

transitions successives aux niveaux précédents menant à ce noeud auxquelles correspondent des plus petites distances entre les symboles codés et les symboles reçus. Pour les simulations, diverses métriques ont été essayées. L'algorithme à pile original fonctionnant avec des métriques entières, une métrique entière de la forme $10(1-d^2)$ fut essayée; d^2 indique le carré de la distance euclidienne entre un symbole reçu et un symbole codé pour une transition donnée. La valeur réelle $(1-d^2)$ est multipliée par 10 pour être ensuite tronquée à une valeur entière. Cette troncation nuisant aux performances du décodeur le programme a été modifié pour accepter des métriques réelles. Les essais ont montré que pour un rapport signal sur bruit élevé il était souhaitable de faire chuter très fortement les métriques des mauvais chemins dans l'arbre pour favoriser au maximum le bon chemin. Par contre, pour un faible rapport signal à bruit une telle procédure peut être néfaste. En effet, le bon chemin peut avoir une métrique qui chute exagérément lors d'une rafale de bruit particulièrement forte et qui n'arrive plus à remonter pour se distinguer de celles des mauvais chemins. Une métrique de la forme $(A-\beta d^2)$ fut alors considérée. A désigne une constante arbitraire et β un coefficient variant selon le rapport signal sur bruit. Avec une telle forme les montées de métriques, quand βd^2 est petit, sont réglées par A .

Les chutes, par contre, dépendent de la grandeur de βd^2 par rapport à A . Pour un rapport signal à bruit élevé une grande valeur pour

β par rapport à A assure que les métriques des mauvais chemins chutent très rapidement. Pour un faible rapport β doit avoir une valeur plus voisine de A. Cette dépendance de la métrique pour le rapport signal sur bruit à incité l'utilisation de la valeur SNR du rapport signal sur bruit, définie à l'équation (4.53), comme valeur de β . Ceci donne, par exemple, un β égal à 2 pour un rapport signal sur bruit de 3dB, et un β égal à 50 pour un rapport de 17dB. Deux valeurs de A furent alors considérées soit 1 et 10. Avec un A égal à 1 le décodeur effectue plus de retours en arrière, donc plus de calculs, lors de sa progression dans l'arbre que pour le cas où A est égal à 10. En effet, avec A = 1, les montées de métriques sont en moyenne plus faibles que les chutes et ceci peut occasionner des retours en arrière fréquents. Ces derniers sont moins fréquents pour A égal à 10 et le décodeur fait moins de calculs. Ces deux formes de métriques furent alors retenues pour les essais définitifs présentés dans ce rapport.

$$M_1 = 1 - SNRd^2 \quad (4.56)$$

$$M_{10} = 10 - SNRd^2 \quad (4.57)$$

Pour fin de comparaison quelques essais avec la métrique entière

$$M = 10(1-d^2) \quad (4.58)$$

sont également présentés.

Pour tous les essais qui suivent une courbe représentant la probabilité d'erreur de bit en fonction du rapport signal à bruit est tracée. Les bornes, supérieure et inférieure, pour la probabilité d'erreur sont indiquées en pointillés et ceci permet d'évaluer le rendement des décodeurs séquentiels multi-chemins adaptés aux canaux dispersifs. Pour deux des canaux étudiés des comparaisons sont faites avec les essais de Magee (1972) avec un décodeur de Viterbi. Finalement, une comparaison avec un essai plus récent de Crozier (1979) avec un Viterbi également est présentée. A la suite des essais pour un canal donné un tableau regroupant l'information relative aux nombres de calculs effectués par les divers décodeurs est présenté. Comme il a été mentionné précédemment, les essais sur 10,000 symboles se divisent en 25 blocs de 400 symboles ou en 20 blocs de 500 symboles. De plus, les décodeurs séquentiels multi-chemins étudiés fonctionnent comme décodeurs à chemin-unique tant que leurs piles ne contiennent pas suffisamment d'éléments pour qu'ils fonctionnent comme multi-chemins. Pour tenir compte de ceci, le nombre théorique de calculs qu'effectue un décodeur N-chemins qui ne fait jamais de retours en arrière est établi comme suit:

Si 25 blocs de 400 symboles sont transmis, alors,

$$\text{Nombre de calculs} = N(25)(400) - 25(N-1)^2 \quad (4.59)$$

Si 20 blocs de 500 symboles sont transmis, alors,

$$\text{Nombre de calculs} = N(20)(500) - 20(N-1)^2 \quad (4.60)$$

Ces équations se vérifient en considérant que lors des simulations Fortran un décodeur N-chemins fonctionne comme décodeur à chemin unique pour les N-1 premiers symboles de chaque bloc.

A partir de ceci il sera possible d'évaluer le pourcentage de calculs supplémentaires dus aux retours en arrière que devra effectuer un décodeur utilisant une métrique particulière.

Les résultats présentés dans les tableaux qui suivent les différents essais ont été obtenus pour un rapport signal sur bruit de 3dB. Cette valeur a été choisie comme référence car, dans la majorité des essais, elle correspond au plus petit rapport signal sur bruit utilisé; c'est donc le cas où les décodeurs séquentiels multi-chemins effectuent le plus grand nombre de retours en arrière lors de la recherche du chemin correct dans l'arbre. Ceci permet de faire une comparaison équitable avec l'effort de calcul constant d'un décodeur de Viterbi.

4.8 CANAL 0.19, 0.35, 0.46, 0.5, 0.46, 0.35, 0.19

Ce canal dispersif avec une réponse impulsionale s'étendant sur 7 symboles provient du tableau 1. C'est donc un canal normalisé de longueur 7 provoquant la perte maximum par rapport à un canal normalisé de longueur 1. Le codeur utilisé pour représenter convenablement ce canal a une mémoire de 6. Un décodeur de Viterbi adapté à ce canal devrait effectuer 64 calculs à chaque niveau du treillis représentant le codage. Pour 10,000 symboles de séquence d'entrée cela impliquerait 640,000 calculs. C'est pour de tels canaux que l'utilisation d'un décodeur séquentiel est particulièrement avantageuse. De bons résultats peuvent être obtenus en faisant beaucoup moins que 640,000 calculs.

Les figures 4.3, 4.4 et 4.5 montrent les performances de trois décodeurs séquentiels 7-chemins utilisant trois métriques différentes. A la figure 4.3, c'est la métrique réelle de l'équation (4.57) qui est considérée. Cette métrique ne favorise pas les retours en arrière et ceci est vérifié dans le tableau 2; le décodeur l'utilisant n'effectue pratiquement aucun retour en arrière, son supplément de calcul étant de 0.01%; de plus ce décodeur n'effectue que 10.8% des calculs d'un décodeur de Viterbi. Par contre, la borne supérieure de la probabilité d'erreur est légèrement dépassée vers 11dB de rapport signal sur bruit.

A la figure 4.4, le décodeur 7-chemins utilise la métrique réelle

de l'équation (4.56). Cette métrique favorise les retours en arrière et, d'après le tableau 2, 11.3% de calculs supplémentaires en résultent. Ce décodeur effectue 12% des calculs qu'effectuerait un Viterbi. L'effort de calcul est donc un peu plus grand que dans le cas précédent mais la borne supérieure n'est jamais dépassée.

La performance d'un décodeur 7-chemins utilisant la métrique entière de l'équation (4.58) est montrée à la figure 4.5. Il va sans dire que cette métrique est totalement inappropriée et qu'il est inutile de la considérer pour les autres essais avec ce canal.

Les figures 4.6 et 4.7 comparent l'utilisation des deux métriques réelles considérées précédemment pour des décodeurs 5-chemins; ceux-ci effectuant moins de calculs que les décodeurs 7-chemins, les performances s'en ressentent. Toutefois, comme le montre la figure 4.7, le décodeur utilisant la métrique qui favorise les retours en arrière parvient à ne pas dépasser la borne supérieure de la probabilité d'erreur.

L'effort de calcul est encore réduit en utilisant des décodeurs 3-chemins. Pour ce décodeur, le gain de performance obtenu en effectuant des retours en arrière est assez frappant comme en témoigne la figure 4.9 comparativement à la figure 4.8.

Finalement l'essai d'un décodeur à chemin unique utilisant la

métrique favorisant les retours en arrière est présentée à la figure 4.10. Ce décodeur n'étudiant qu'un seul chemin, il y a un énorme supplément de calculs de 136%, bien supérieur à ceux associés aux décodeurs multi-chemins précédents. Par contre ce décodeur ne fait que 3.69% du nombre de calculs qu'effectuerait un Viterbi. De plus sa performance est meilleure que celle du décodeur 3-chemins utilisant la métrique de l'équation (4.57) (Figure 4.8).

Les essais avec ce canal ont montré l'avantage d'utiliser la métrique réelle favorisant les retours en arrière. Les suppléments de calculs sont plus forts mais ceci n'augmente pas beaucoup l'effort de calcul par rapport à celui du décodeur de Viterbi. Pour cette raison seule la métrique de l'équation (4.56) est considérée lors des essais subséquents avec une métrique réelle. Par contre la métrique entière, (4.58), est essayée avec d'autres canaux, de longueurs inférieures à 7, pour voir si sa performance est toujours aussi mauvaise.

PROBABILITE D'ERREUR V.S. SNR(dB)

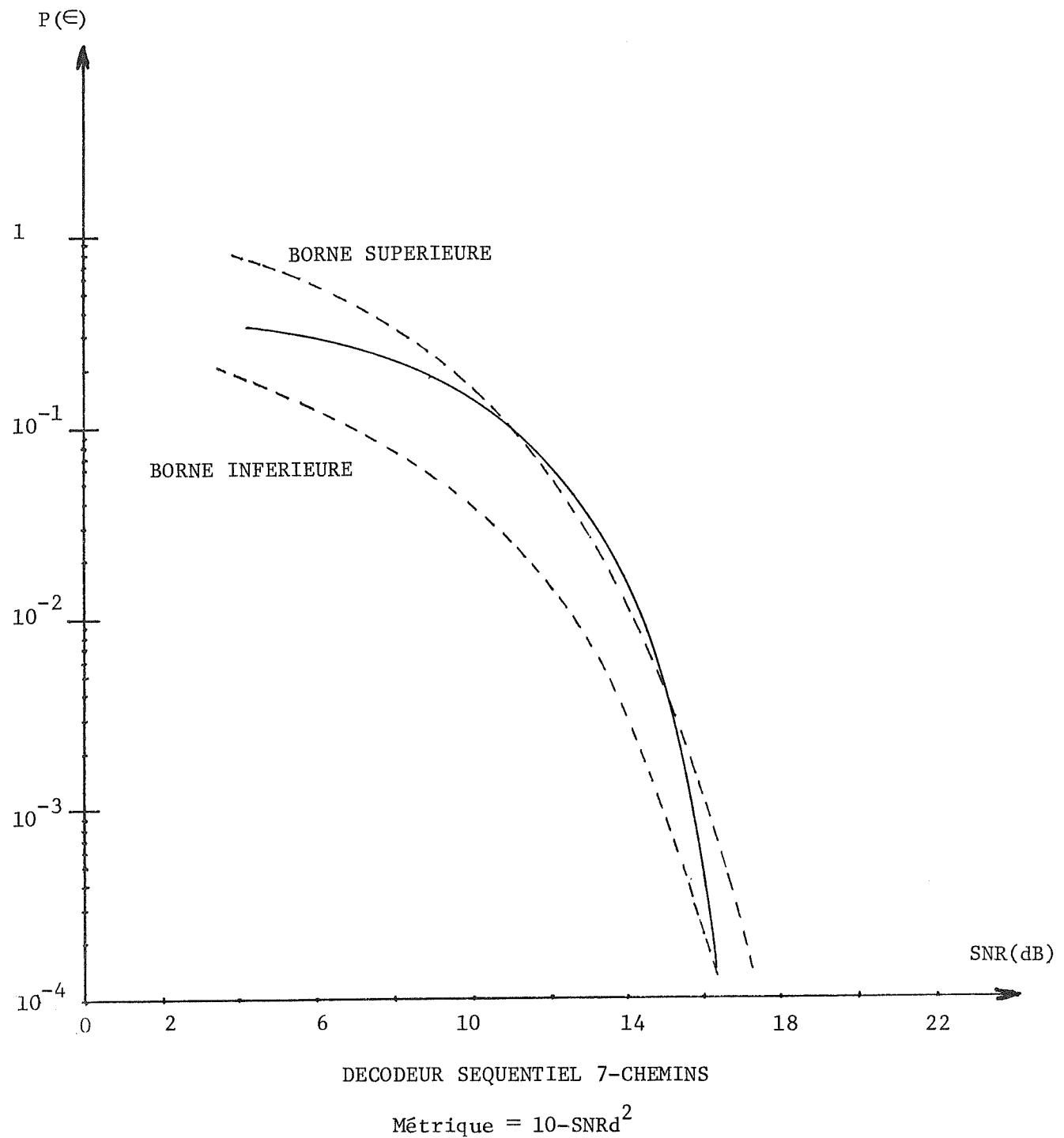


FIGURE 4.3

PROBABILITE D'ERREUR V.S. SNR(dB)

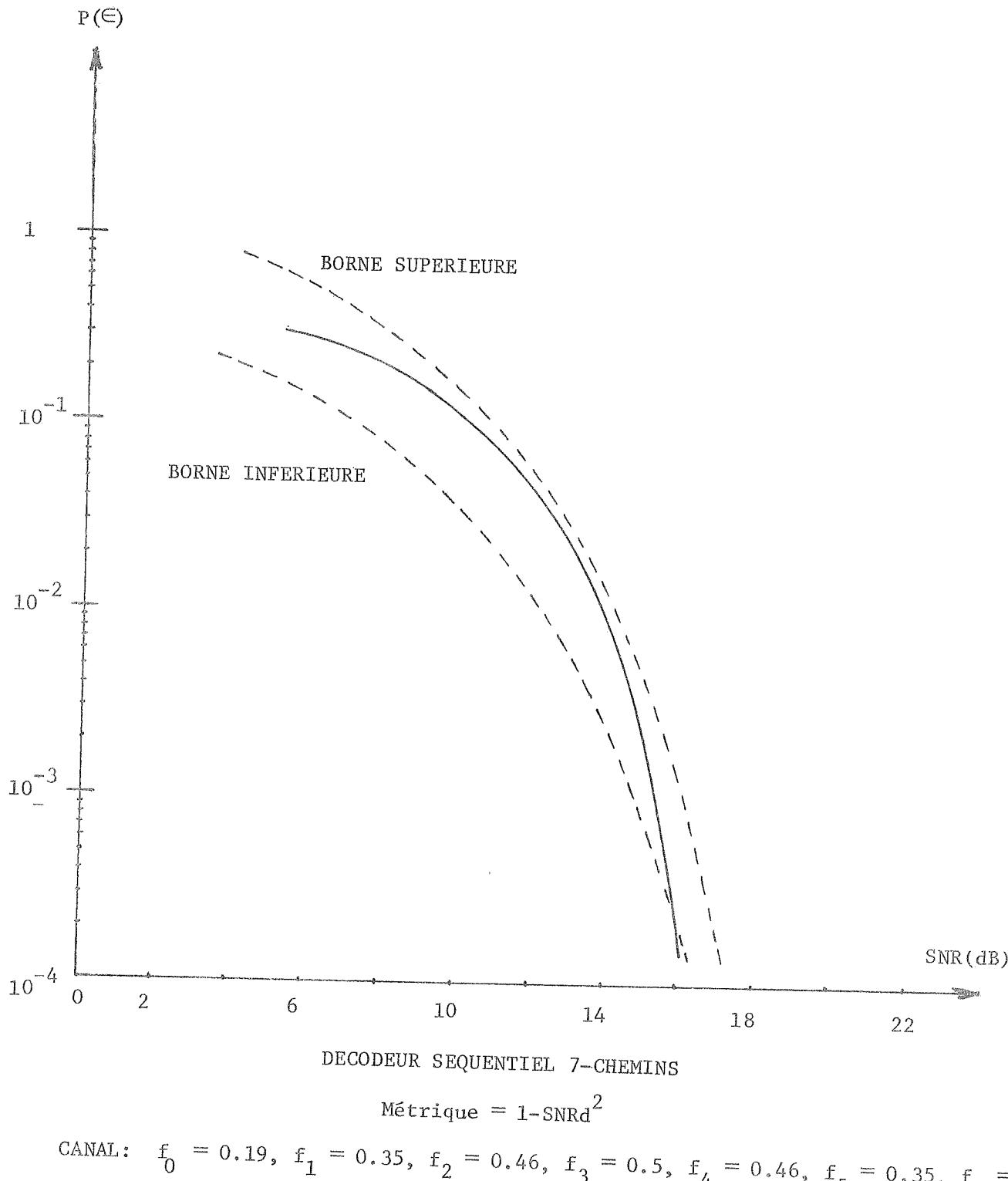
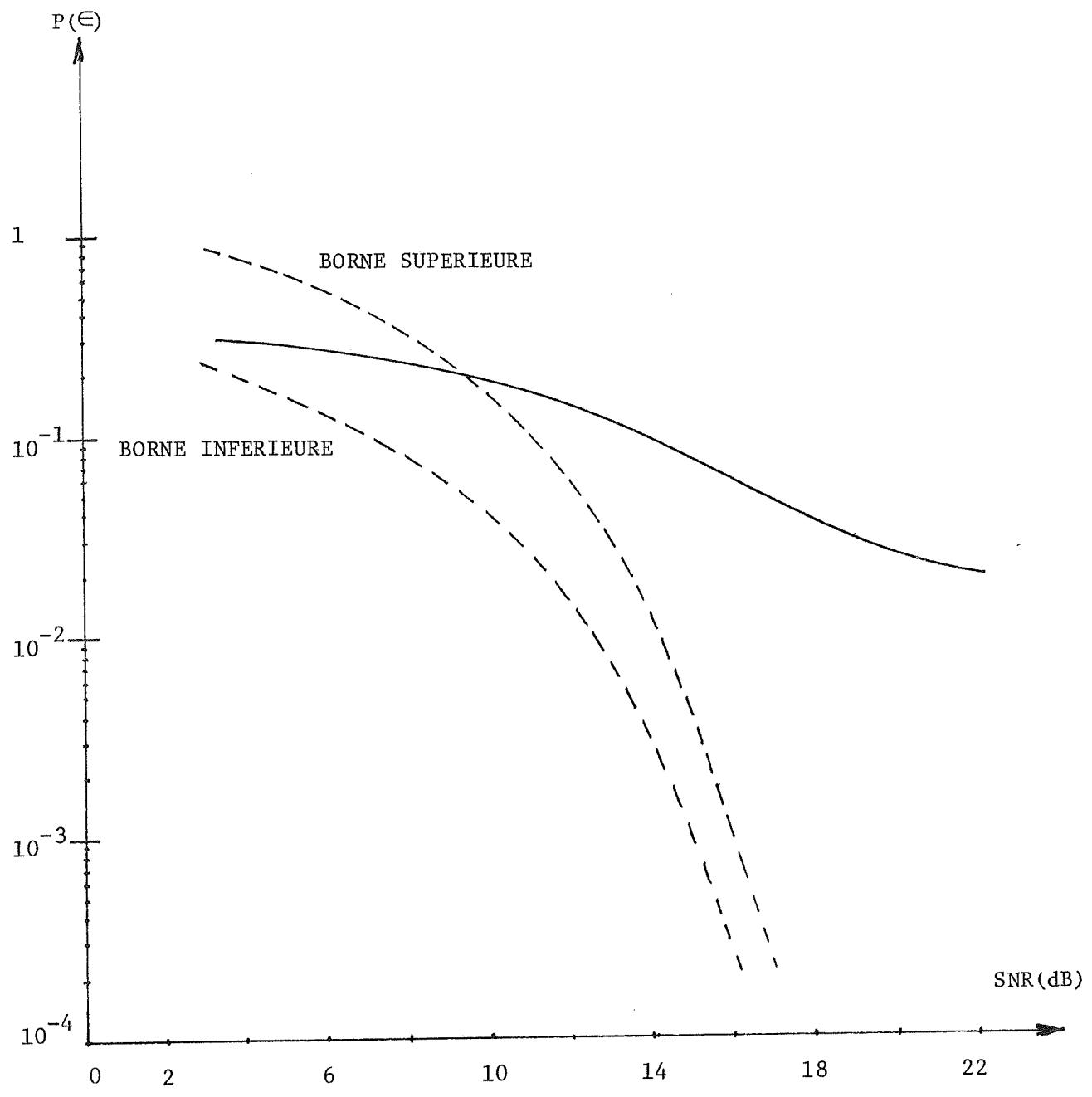


FIGURE 4.4

PROBABILITE D'ERREUR V.S. SNR(dB)



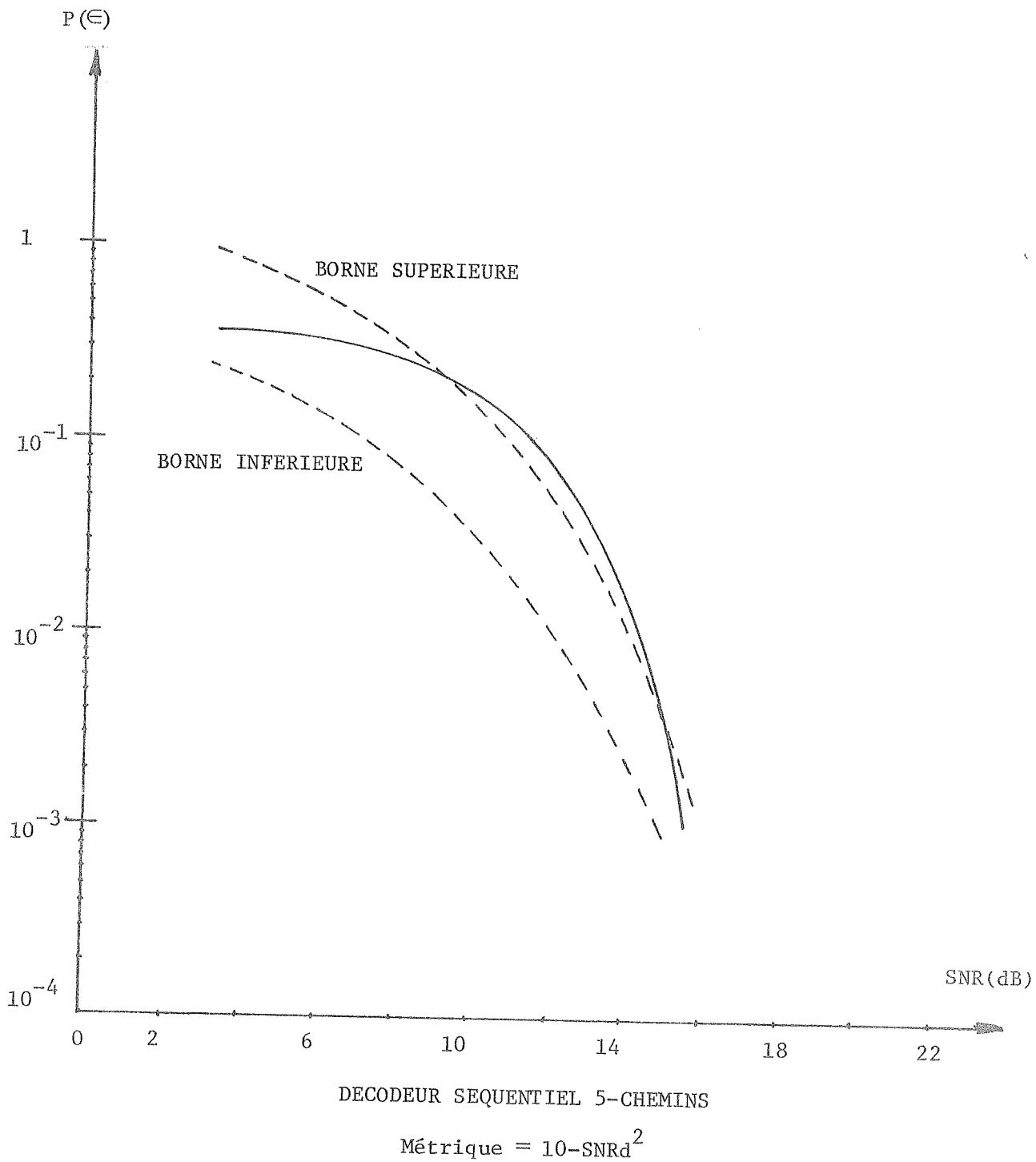
DECODEUR SEQUENTIEL 7-CHEMINS

$$\text{Métrique entière} = 10(1-d)^2$$

CANAL: $f_0 = 0.19, f_1 = 0.35, f_2 = 0.46, f_3 = 0.5, f_4 = 0.46, f_5 = 0.35, f_6 = 0.19$

FIGURE 4.5

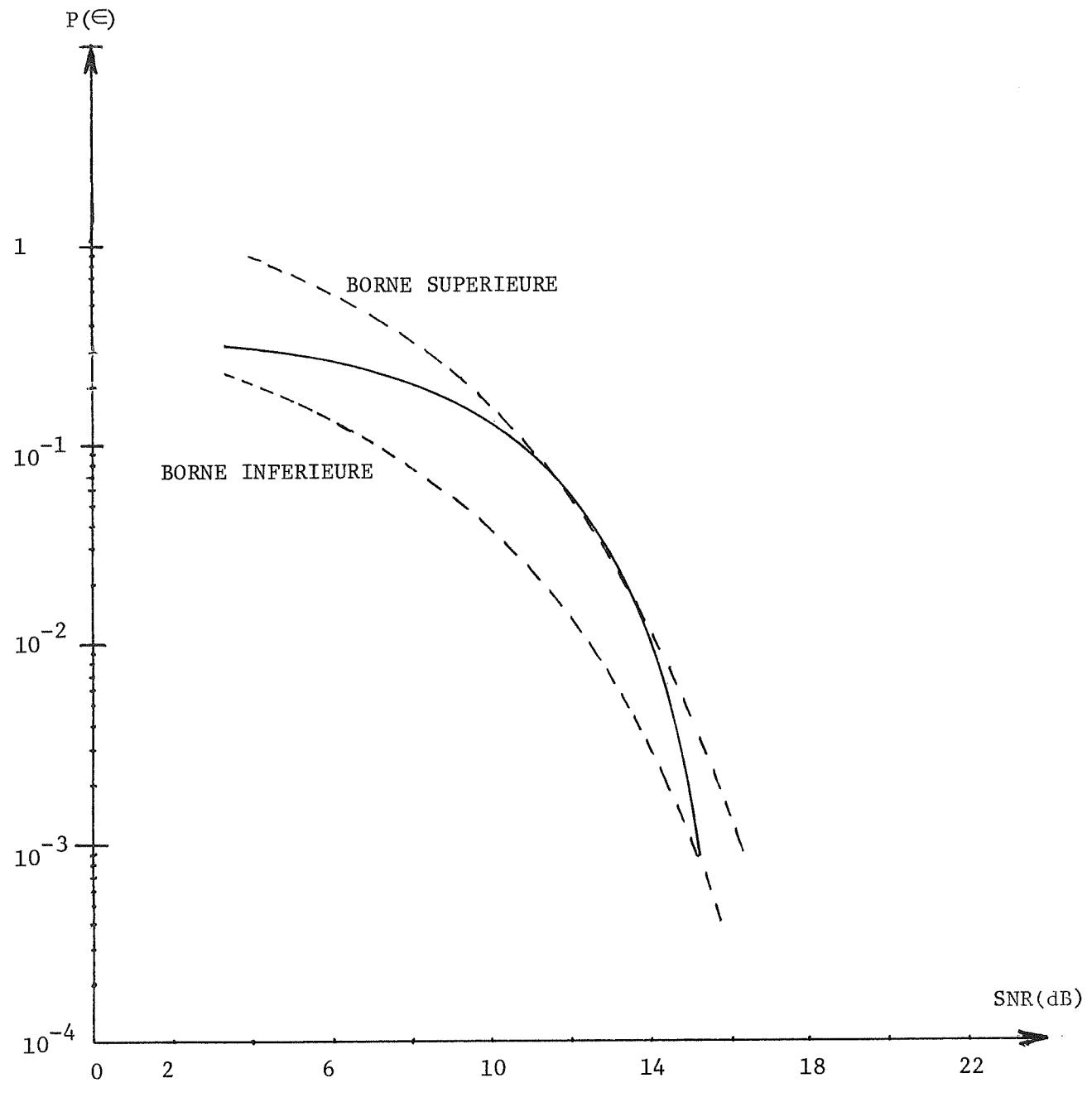
PROBABILITE D'ERREUR V.S. SNR(dB)



CANAL: $f_0 = 0.19, f_1 = 0.35, f_2 = 0.46, f_3 = 0.5, f_4 = 0.46, f_5 = 0.35, f_6 = 0.19$

FIGURE 4.6

PROBABILITE D'ERREUR V.S. SNR(dB)

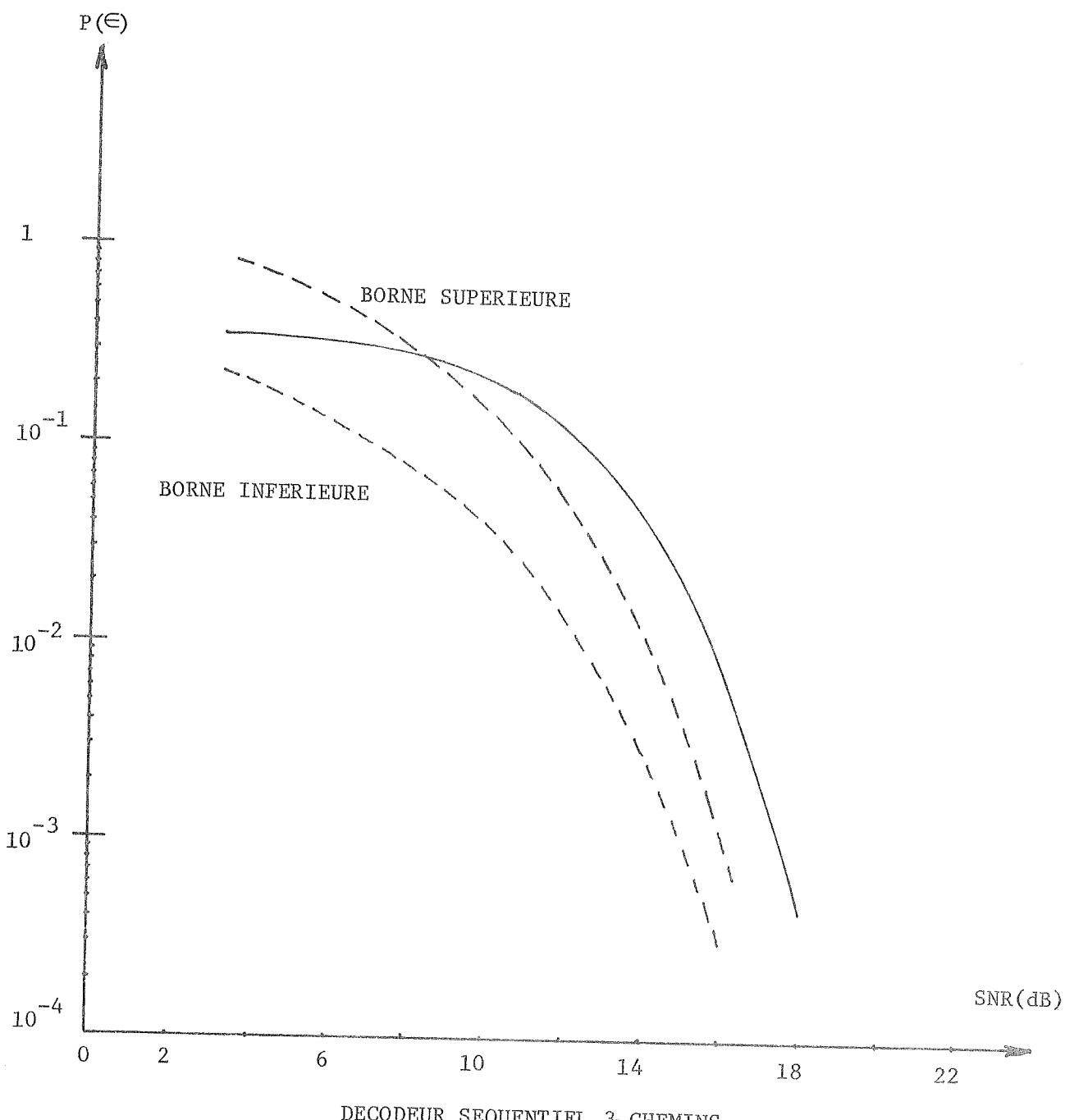


$$\text{Métrique} = 1 - \text{SNR}_d^2$$

CANAL: $f_0 = 0.19, f_1 = 0.35, f_2 = 0.46, f_3 = 0.5, f_4 = 0.46, f_5 = 0.35, f_6 = 0.19$

FIGURE 4.7

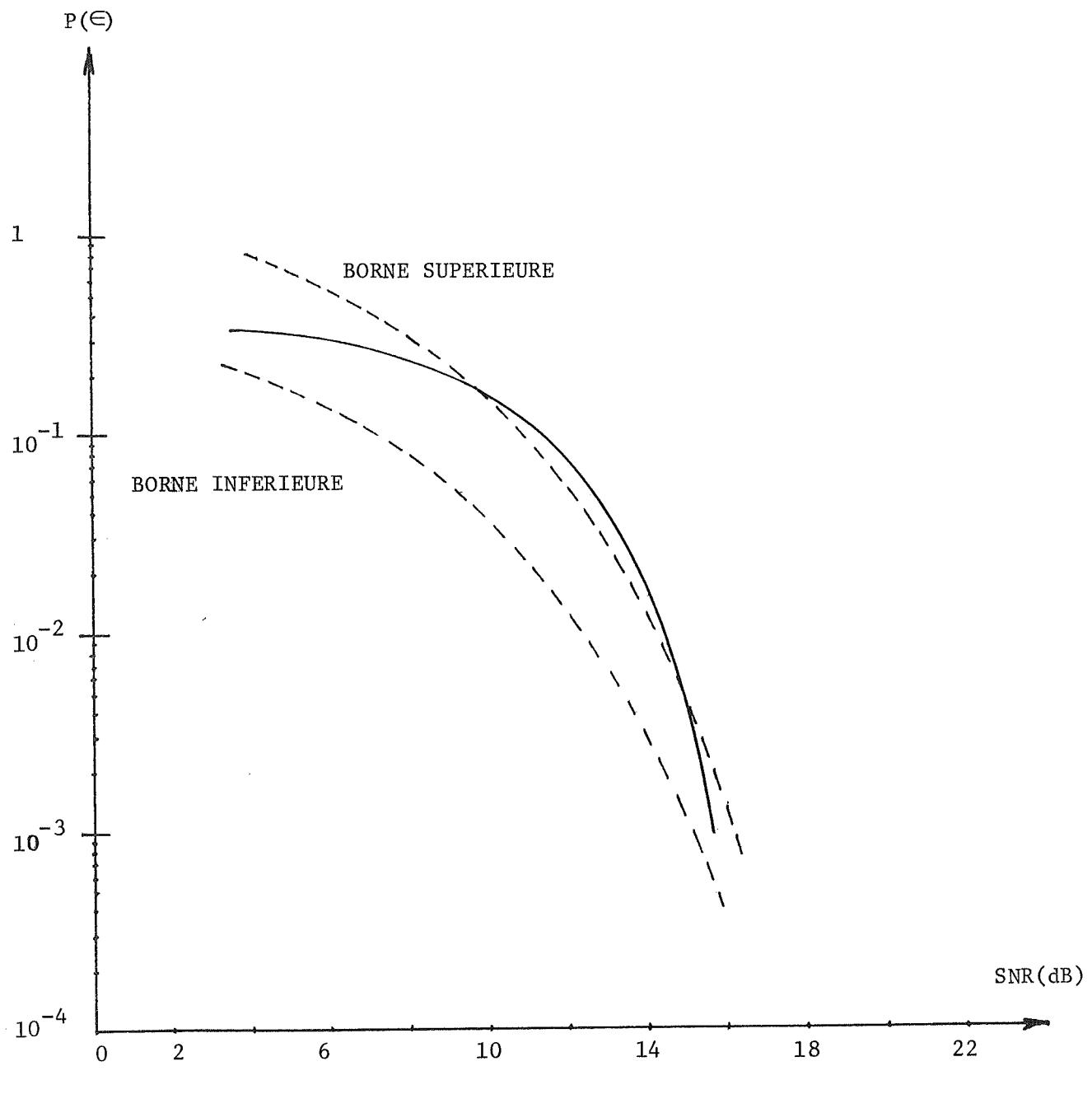
PROBABILITE D'ERREUR V.S. SNR(dB)



CANAL: $f_0 = 0.19, f_1 = 0.35, f_2 = 0.46, f_3 = 0.5, f_4 = 0.46, f_5 = 0.35, f_6 = 0.19$

FIGURE 4.8

PROBABILITE D'ERREUR V.S. SNR(dB)



DECODEUR SEQUENTIEL 3-CHEMINS

$$\text{Métrique} = 1 - \text{SNR}_d^2$$

CANAL: $f_0 = 0.19, f_1 = 0.35, f_2 = 0.46, f_3 = 0.5, f_4 = 0.46, f_5 = 0.35, f_6 = 0.19$

FIGURE 4.9

PROBABILITE D'ERREUR V.S. SNR(dB)

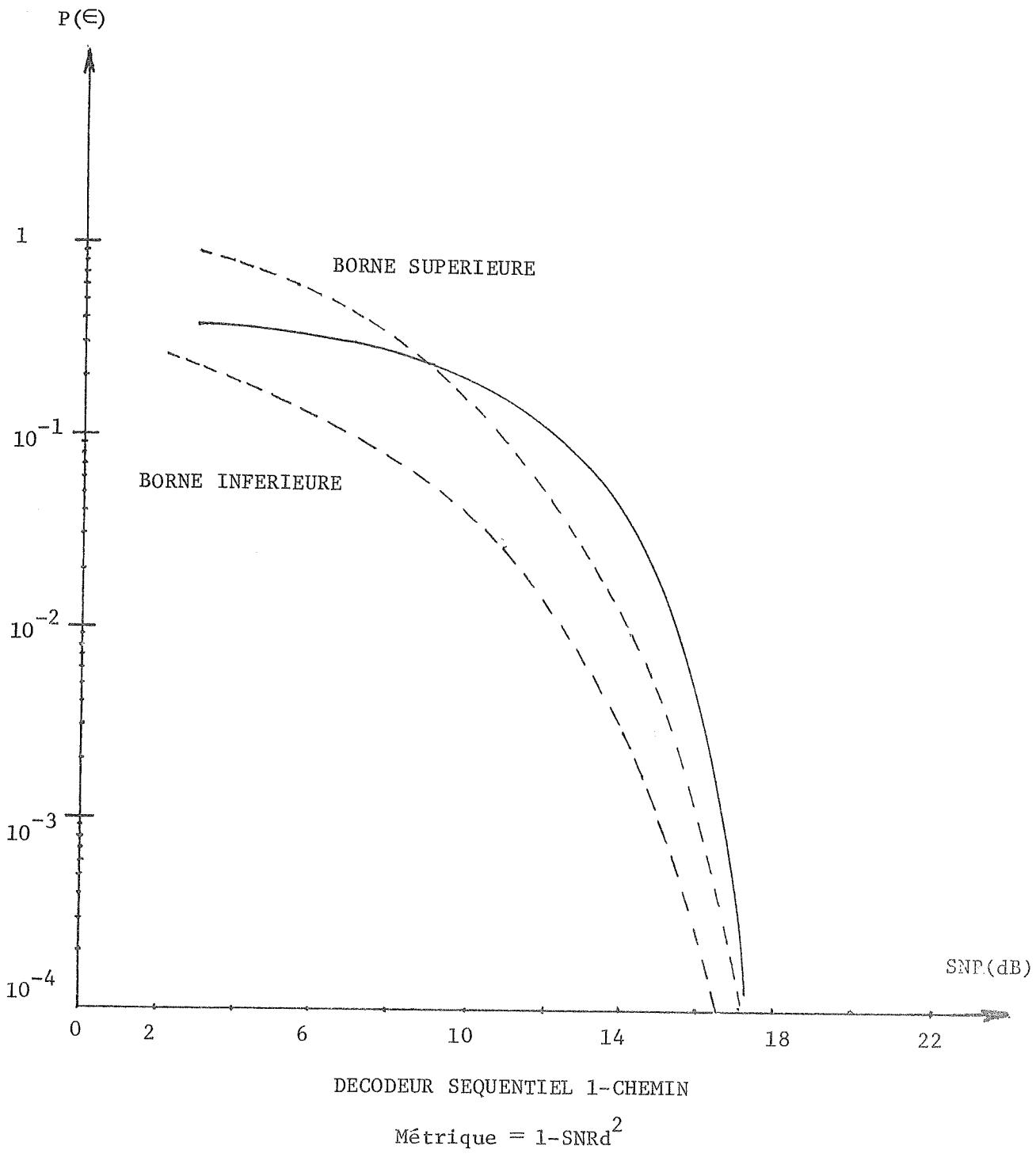


FIGURE 4.10

CANAL 0.19, 0.35, 0.46, 0.5, 0.46, 0.35, 0.19

- 10,000 symboles transmis
- Nombre de calculs effectués par un décodeur de Viterbi = 640,000
- $(\text{SNR})_{\text{dB}} = 3\text{dB}$

Décodeurs Séquentiels Multi-Chemins

$\text{Métrique entière} = 10(1-d^2)$

	Nombre de calculs théorique (sans retours en arrière)	Nombre de calculs réel (avec retours en arrière)	Supplément causé par les retours en arrière	Rapport du nombre de calculs réel avec celui du Viterbi
7-chemins	69,280	69,739	0.663%	10.9%
	$\boxed{\text{Métrique réelle} = 10-\text{SNR}d^2}$			
	22,900	29,903	0.01%	4.67%
	49,600	49,605	0.01%	7.75%
7-chemins	69,100	69,107	0.01%	10.8%
	$\boxed{\text{Métrique réelle} = 1-\text{SNR}d^2}$			
	10,000	23,638	136%	3.69%
	29,900	38,947	30.3%	4.67%
	49,600	57,596	16.1%	9.00%
7-chemins	69,100	76,894	11.3%	12.0%

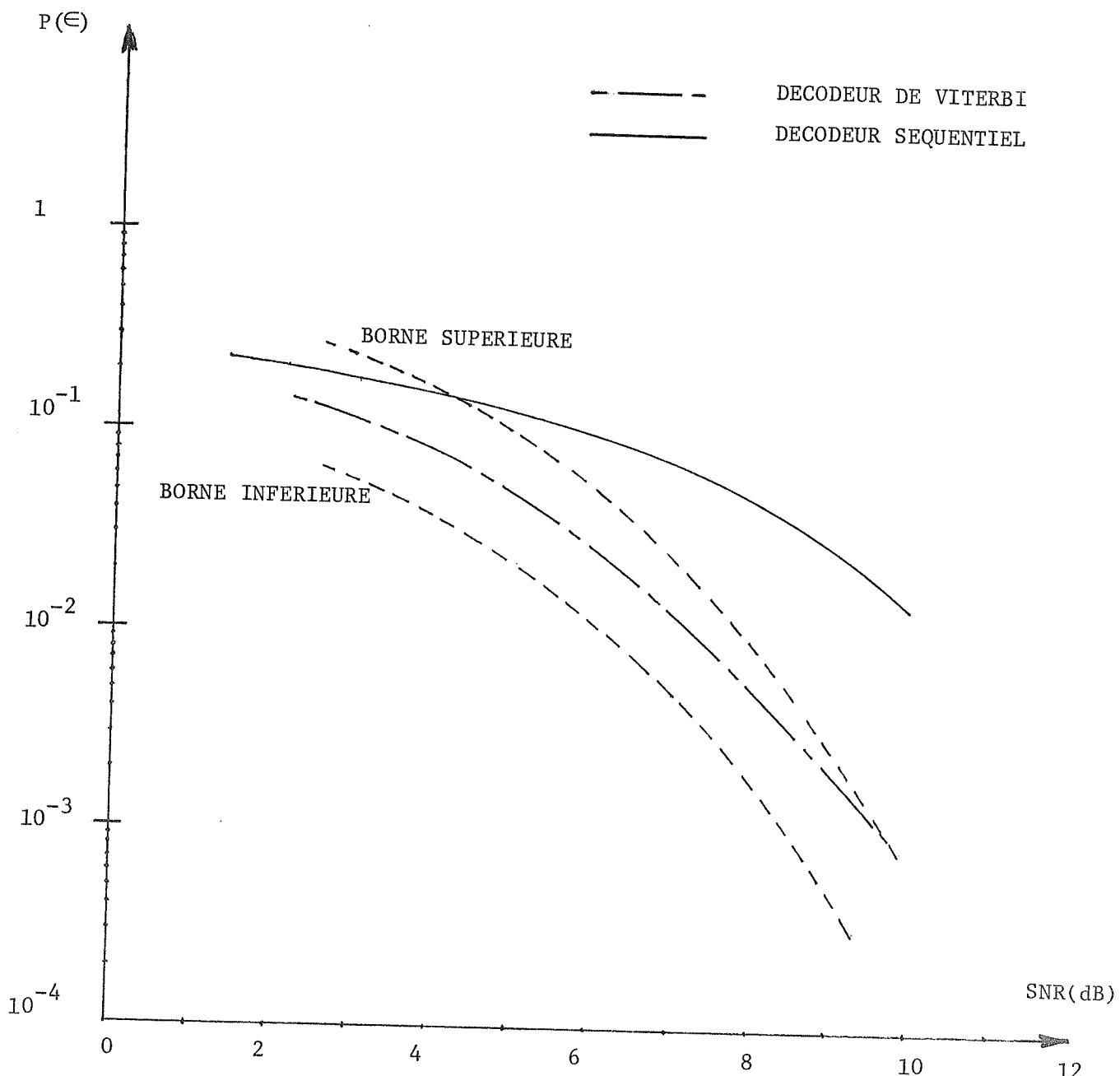
TABLEAU 2

4.9 CANAL 0.33, 0.67, 1, 0.67, 0.33

Pour ce canal non normalisé de longueur 5 les performances de décodeurs séquentiels multi-chemins sont comparées avec celle du décodeur de Viterbi utilisé par Magee (1972). Les essais de Magee s'effectuaient également sur 10,000 symboles mais avec un générateur aléatoire différent de celui utilisé pour ce projet. La métrique utilisée par Magee dépendait uniquement des distances euclidiennes entre les symboles reçus et ceux résultant du codage, sans aucun biaisage. Pour les décodeurs séquentiels la métrique entière de l'équation(4.58) a d'abord été essayée. Cette métrique, ayant un terme constant de 10 qui est en général plus élevé que d^2 dans l'équation(4.58) ne favorise pas les retours en arrière. En effet les métriques cumulatives de tous les chemins étudiés ont plus ou moins tendance à augmenter. Des essais avec la métrique entière pour des décodeurs 2-chemins et 6-chemins sont donnés aux figures 4.11 et 4.12 respectivement. Le décodeur 2-chemins a une performance bien inférieure à celle du Viterbi et dépasse la borne supérieure à partir d'un rapport signal sur bruit de 4.2 dB . Une performance plus compétitive avec celle du Viterbi est réalisée avec un décodeur 6-chemins. La borne supérieure est dépassée à partir de 7.8 dB de rapport signal sur bruit. Par contre, les essais avec la métrique réelle de l'équation(4.56)donnent d'excellentes performances pour le décodeur séquentiel; tout en étant plus précise que la métrique entière, elle favorise les retours en arrière lors du décodage, comme cela a déjà été mentionné. Un décodeur à chemin unique utilisant cette métrique donne de fort bons résultats, (Figure 4.13), se.

rapprochant de ceux du Viterbi. La borne supérieure n'est pas dépassée. Un décodeur 3-chemins donne d'excellents résultats,(Figure 4.14), et surpassé même le décodeur optimal de Viterbi; ceci est probablement dû au générateur aléatoire différent qu'a utilisé Magee. Toutefois cela donne une idée des possibilités du décodeur séquentiel multi-chemins. Ce dernier rivalise avec le décodeur optimal en effectuant beaucoup moins de calculs comme en témoigne le tableau 3. Le décodeur séquentiel 3-chemins avec la métrique réelle donne une performance comparable à celle du Viterbi en ne faisant que 20.5% des calculs du Viterbi. Le décodeur à chemin unique ne fait que 10.4% des calculs du Viterbi; par contre, du fait qu'il n'étudie qu'un seul chemin à la fois il est plus sujet aux retours en arrière ce qui augmente son nombre de calculs de 67.1% comparativement à 9.69% pour le décodeur 3-chemins.

PROBABILITE D'ERREUR V.S. SNR(dB)



DECODEUR SEQUENTIEL 2-CHEMINS

$$\text{Métrique entière} = 10(1-d^2)$$

CANAL: $f_0 = 0.33, f_1 = 0.67, f_2 = 1.00, f_3 = 0.67, f_4 = 0.33$

FIGURE 4.11

PROBABILITE D'ERREUR V.S. SNR(dB)

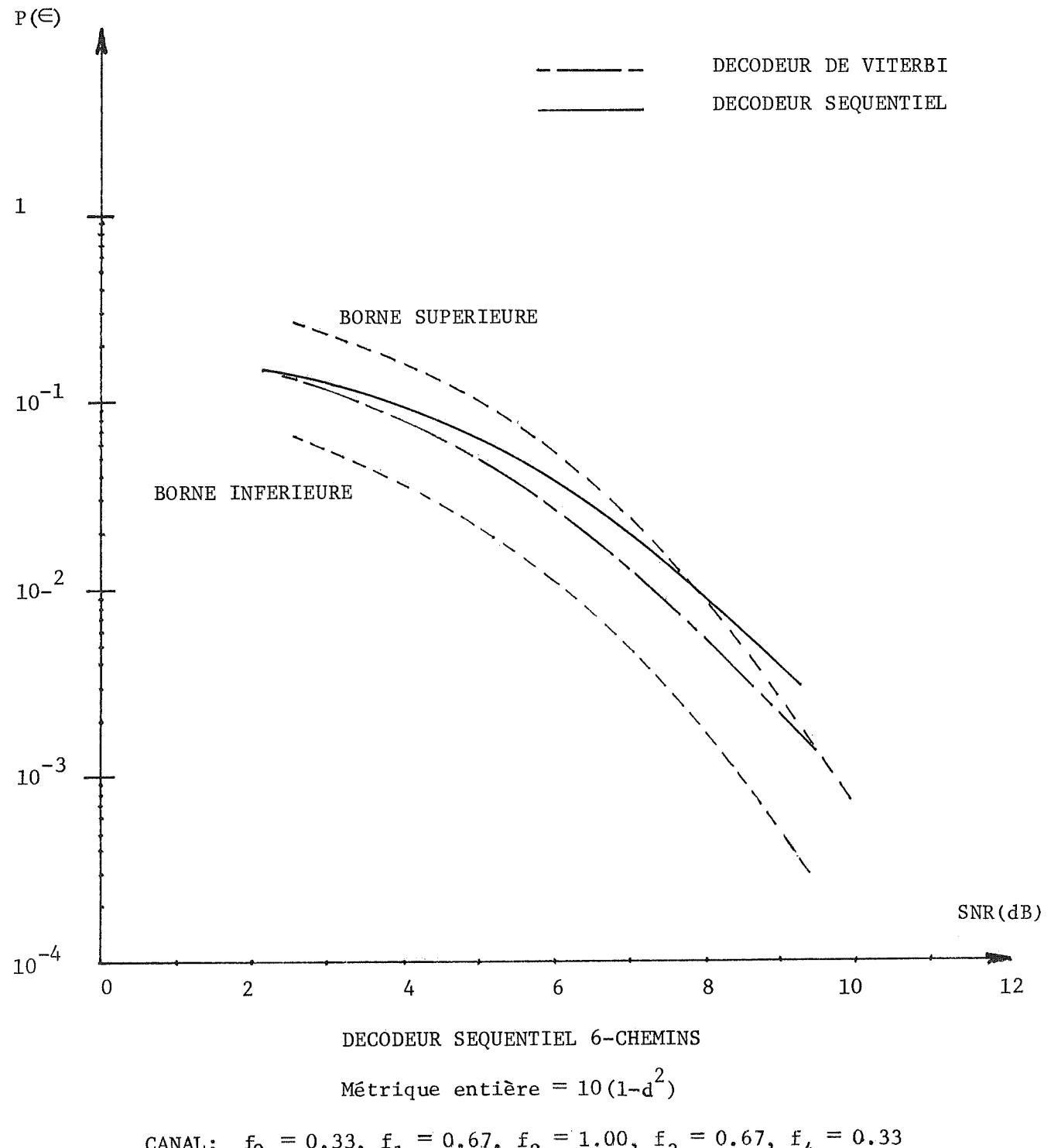


FIGURE 4.12

PROBABILITE D'ERREUR V.S. SNR(dB)

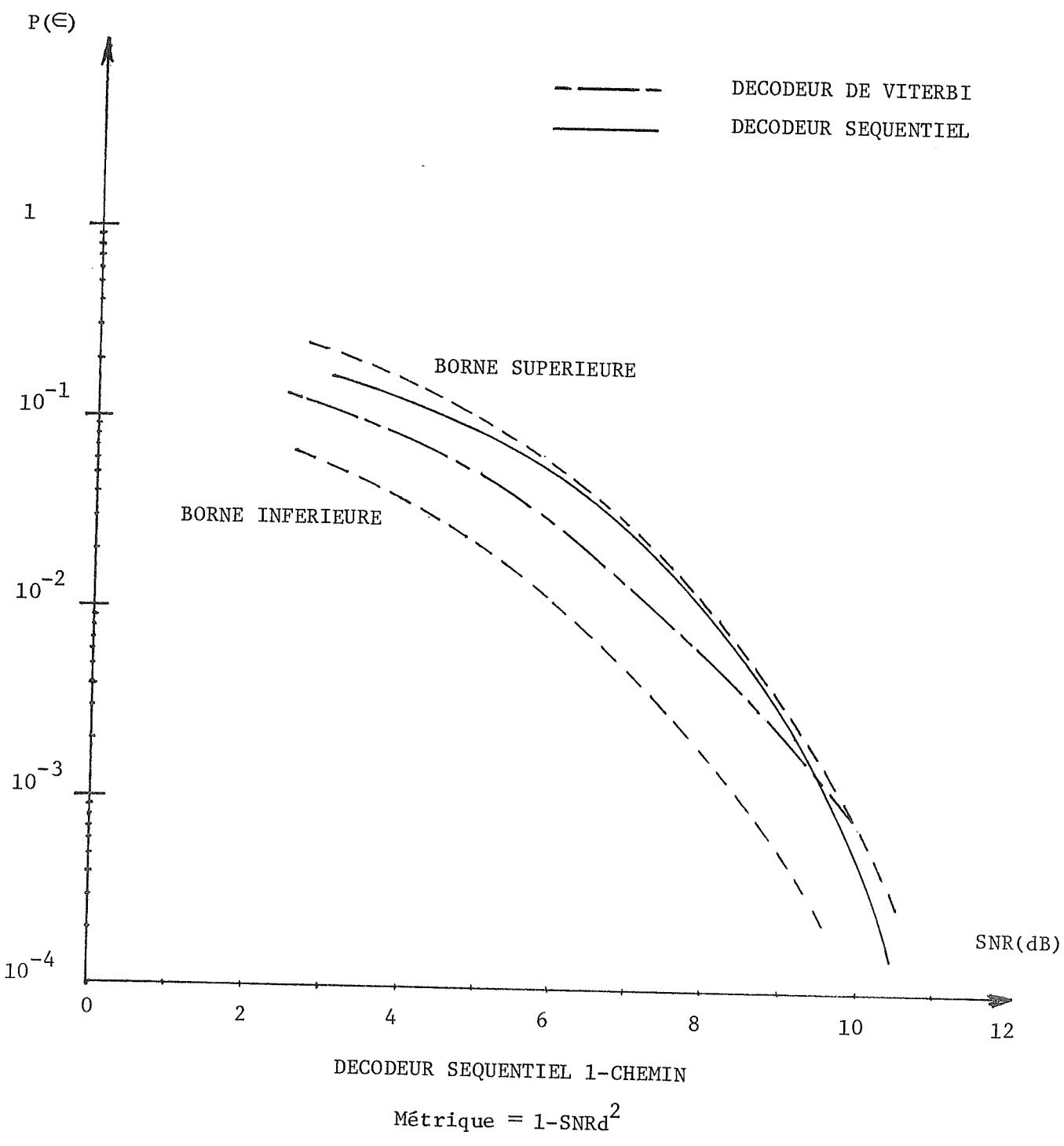


FIGURE 4.13

PROBABILITE D'ERREUR V.S. SNR(dB)

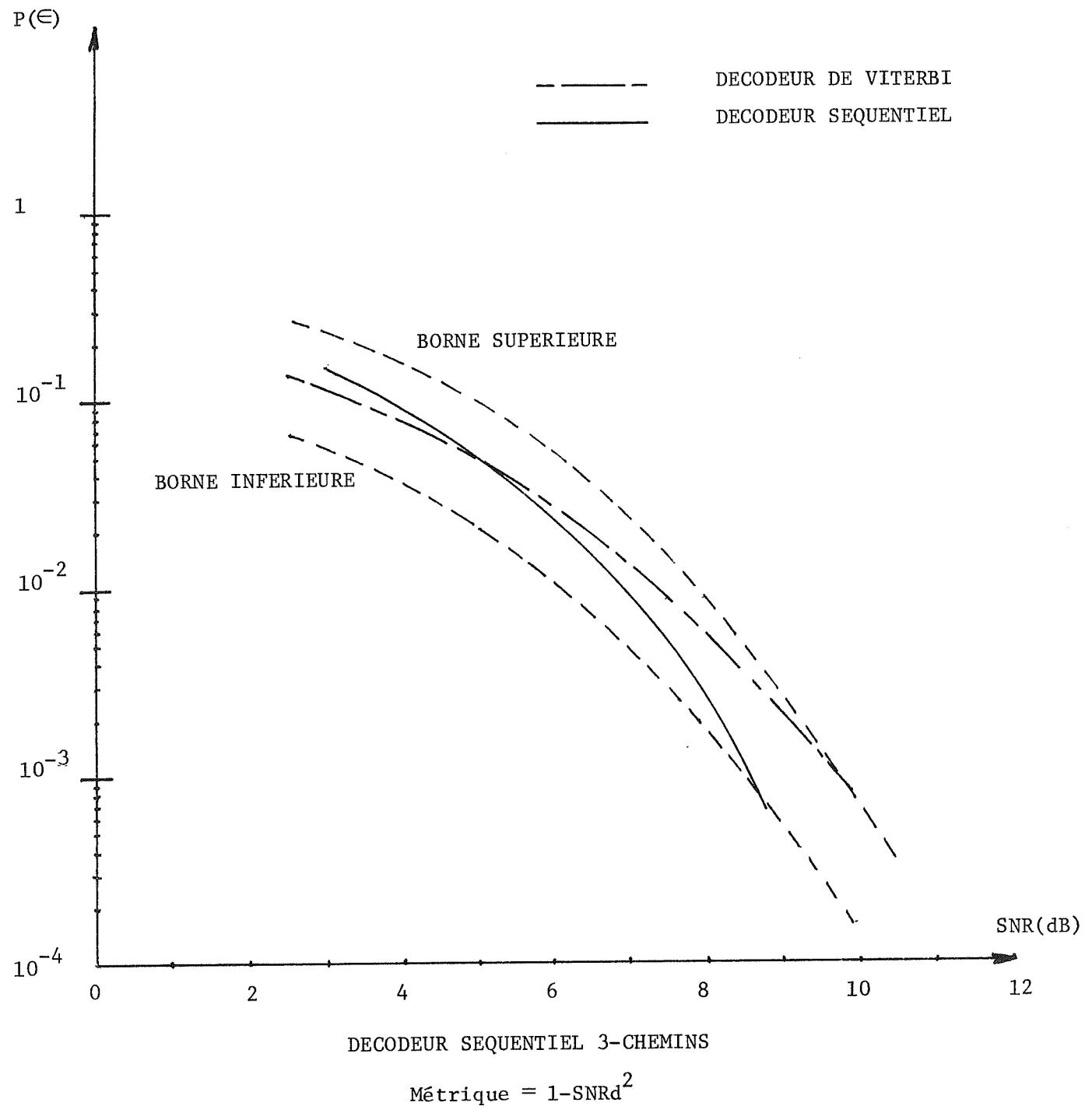


FIGURE 4.14

CANAL 0.33, 0.67, 1.0, 0.67, 0.33

- 10,000 symboles transmis
- Nombre de calculs effectués par un décodeur de Viterbi = 160,000
- $(\text{SNR})_{\text{dB}} = 3\text{dB}$

Décodeurs Séquentiels Multi-Chemins

$$\boxed{\text{Métrique entière } M = 10(1-d^2)}$$

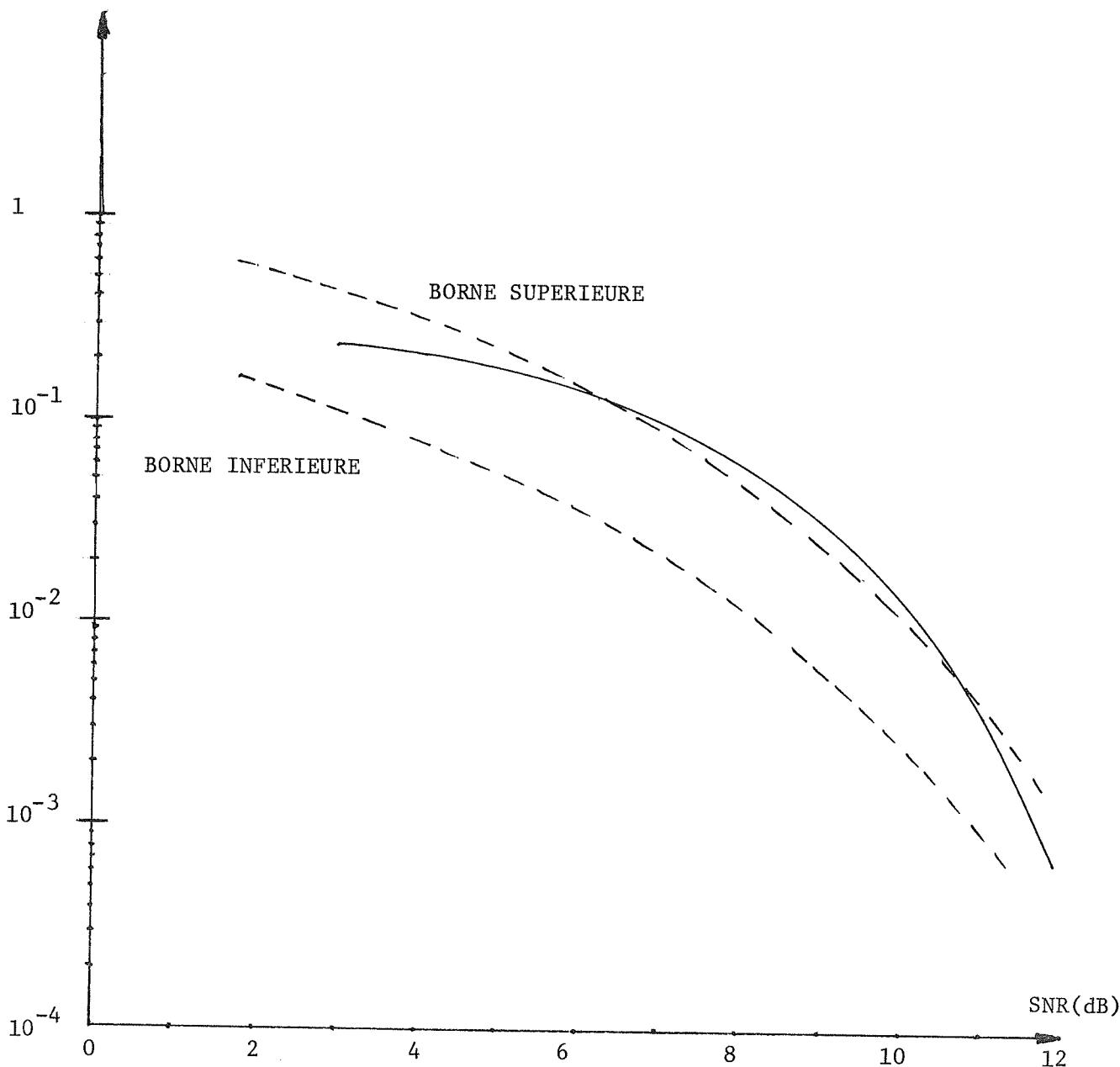
	Nombre de calculs théorique (sans retours en arrière)	Nombre de calculs réel (avec retours en arrière)	Supplément causé par les retours en arrière	Rapport du nombre de calculs réel avec celui du Viterbi
2-chemins	19,980	21,217	6.69%	13.3%
6-chemins	59,500	60,427	1.57%	37.8%
	$\boxed{\text{Métrique réelle } M_1 = 1.0 - \text{SNR}d^2}$			
1-chemin	10,000	16,713	67.1%	10.4%
3-chemins	29,900	32,797	9.69%	20.5%

TABLEAU 3

4.10 CANAL 0.38, 0.6, 0.6, 0.38

Pour ce canal seule la métrique réelle de l'équation 4.56 est utilisée. Les figures 4.15, 4.16 et 4.17 illustrent les performances de décodeurs 1-chemin, 3-chemins et 5-chemins. Avec le décodeur 1-chemin la borne supérieure est dépassée vers 7 dB de rapport signal sur bruit pour ensuite être recoupée vers 11dB. Avec les autres décodeurs la borne n'est pas dépassée. Il est intéressant de remarquer que le décodeur 3-chemins est presque aussi bon que le décodeur 5-chemins en faisant, à peu près, 40% de calculs de moins; par rapport au décodeur de Viterbi le décodeur 3-chemins ne fait que 40.8% des calculs alors que le 5-chemins en fait 65.8% (Tableau 4). Le décodeur 3-chemins peut donc être jugé le plus avantageux.

PROBABILITE D'ERREUR V.S. SNR(dB)



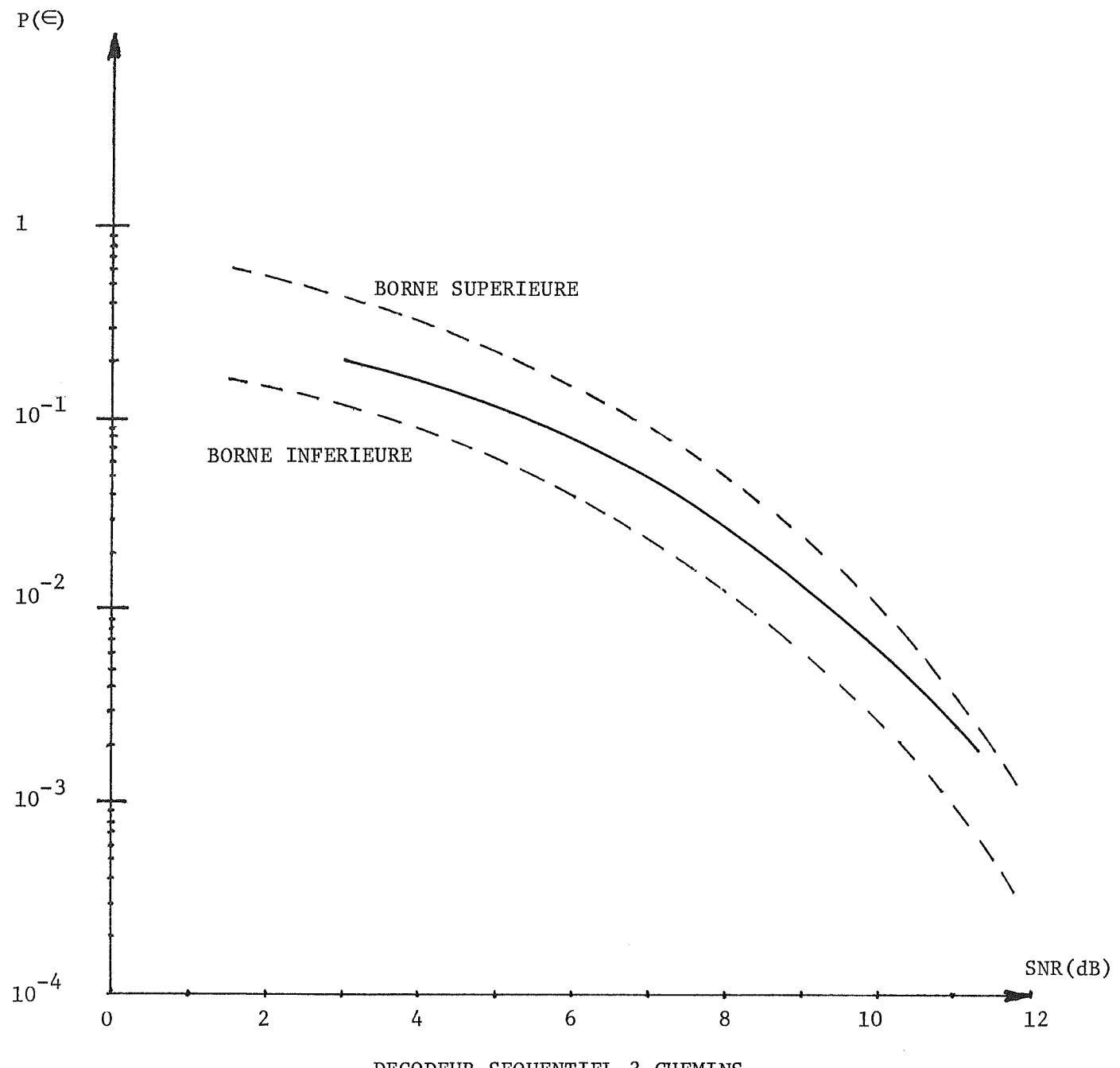
DECODEUR SEQUENTIEL 1-CHEMIN

$$\text{Métrique} = 1 - \text{SNR}_d^2$$

CANAL: $f_0 = 0.38$, $f_1 = 0.6$, $f_2 = 0.6$, $f_3 = 0.38$

FIGURE 4.15

PROBABILITE D'ERREUR V.S. SNR(dB)

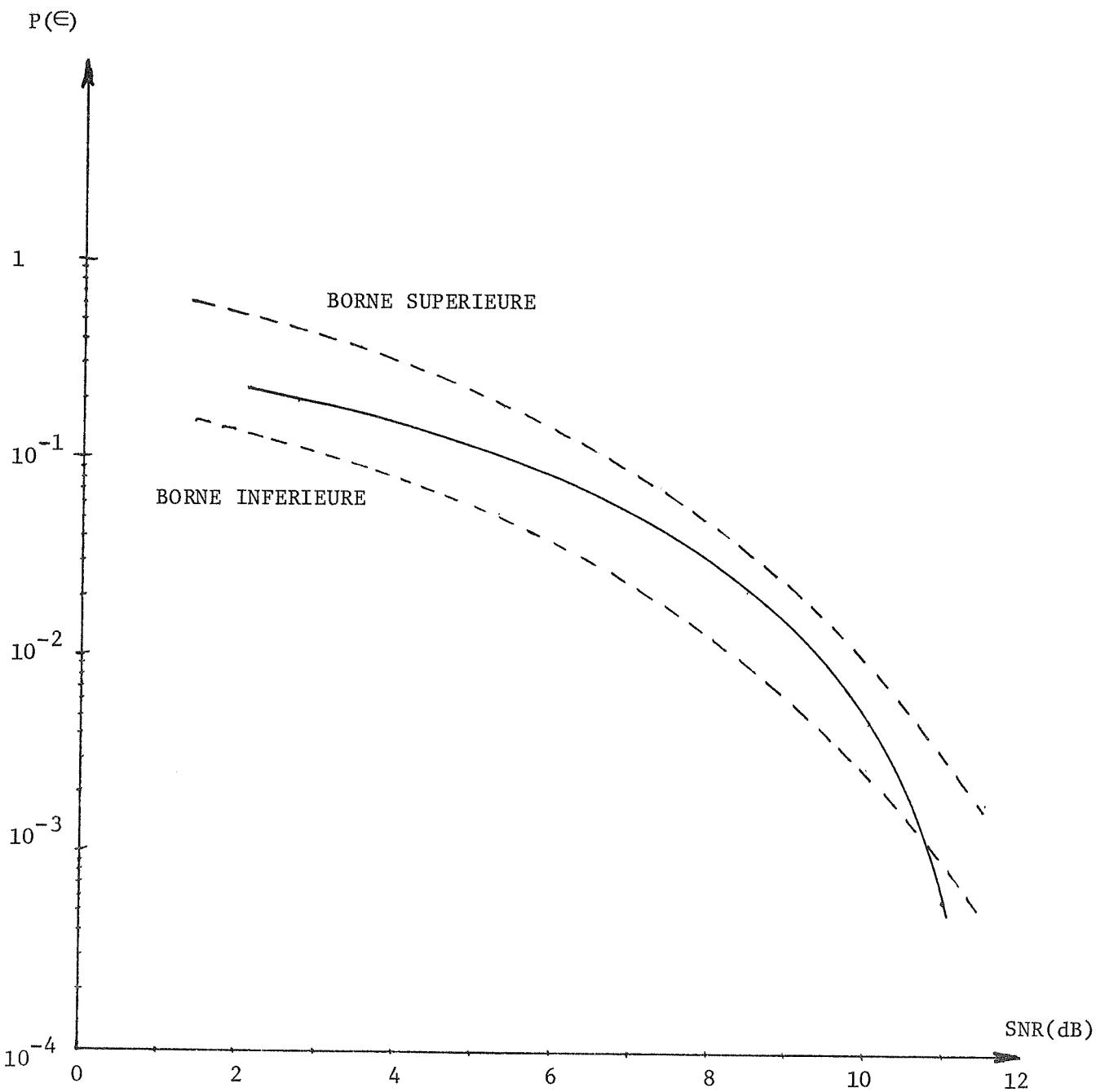


$$\text{Métrique} = 1 - \text{SNR}_d^2$$

$$\text{CANAL: } f_0 = 0.38, f_1 = 0.6, f_2 = 0.6, f_3 = 0.38$$

FIGURE 4.16

PROBABILITE D'ERREUR V.S. SNR(dB)



DECODEUR SEQUENTIEL 5-CHEMINS

$$\text{Métrique} = 1 - \text{SNR}_d^2$$

CANAL: $f_0 = 0.38, f_1 = 0.6, f_2 = 0.6, f_3 = 0.38$

FIGURE 4.17

CANAL 0.38, 0.6, 0.6, 0.38

- 10,000 symboles transmis
- Nombre de calculs effectués par un décodeur de Viterbi = 80,000
- $(\text{SNR})_{\text{dB}} = 3\text{dB}$

Décodeurs Séquentiels Multi-Chemins

$$\text{Métrique réelle } M_1 = 1.0 - \text{SNRd}^2$$

	Nombre de calculs théorique (sans retours en arrière)	Nombre de calculs réel (avec retours en arrière)	Supplément causé par les retours en arrière	Rapport du nombre de calculs réel avec celui du Viterbi
1-chemin	10,000	14,496	45%	18.1%
3-chemins	29,900	32,625	9.11%	40.8%
5-chemins	49,600	52,630	6.11%	65.8%

TABLEAU 4

4.11 CANAL 0.5, 1.0, 0.5

Pour ce canal les essais avec les décodeurs séquentiels sont de nouveau comparés avec ceux de Magee (1972) avec un Viterbi. Des essais de décodeurs avec la métrique entière de l'équation (4.58) sont également présentés. Ces derniers montrent encore une fois que cette métrique n'est guère reluisante, comme en témoignent les figures 4.18 et 4.19. En effet le décodeur à chemin unique donne une performance très médiocre comparativement au Viterbi; la borne est dépassée à partir d'un rapport signal sur bruit de 3dB et la situation ne s'améliore pas même pour des rapports élevés. Il y a une nette amélioration avec le décodeur 2-chemins comme le montre la figure 4.19. Mais c'est encore l'utilisation de la métrique réelle de l'équation (4.56) qui donne les meilleurs résultats. Les figures 4.20, 4.21, et 4.22 illustrent les performances de décodeurs 1-chemin, 2-chemins, et 3-chemins avec la métrique réelle. Une comparaison entre les figures 4.20 et 4.18 montre l'énorme avantage qu'offre une métrique réelle par rapport à une métrique entière dans le cas du décodeur à chemin unique. De plus, avec un décodeur 3-chemins à métrique réelle la borne supérieure n'est pas dépassée.

En examinant le tableau 5 il est intéressant de noter les différents pourcentages de suppléments de calculs causés par les retours en arrière, surtout lors de l'utilisation d'une métrique entière. Avec cette dernière, pour un rapport signal sur bruit de 3dB, un décodeur à chemin unique, donnant une performance médiocre, doit

effectuer un supplément de calcul de 15.8% à cause des retours en arrière; en utilisant le décodeur 2-chemins le supplément de calculs tombe à 1.16% et les performances sont bien meilleures. Ce contraste assez frappant illustre bien la flexibilité des décodeurs multi-chemins, même avec des métriques relativement inefficaces..

Comme prévu les suppléments de calculs sont plus élevés avec les décodeurs utilisant la métrique réelle qui favorise les retours en arrière.

PROBABILITE D'ERREUR V.S. SNR(dB)

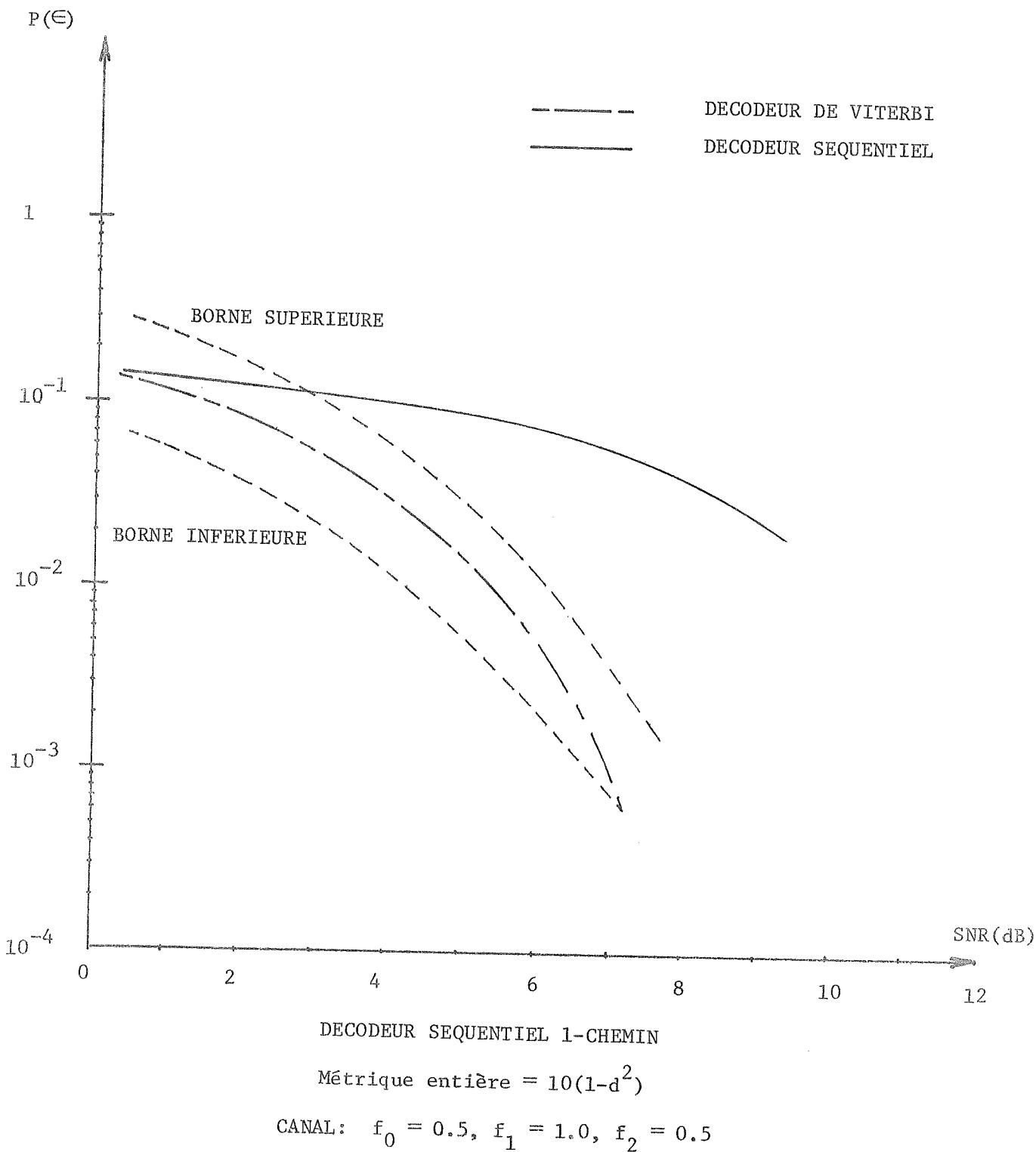


FIGURE 4.18

PROBABILITE D'ERREUR V.S. SNR(dB)

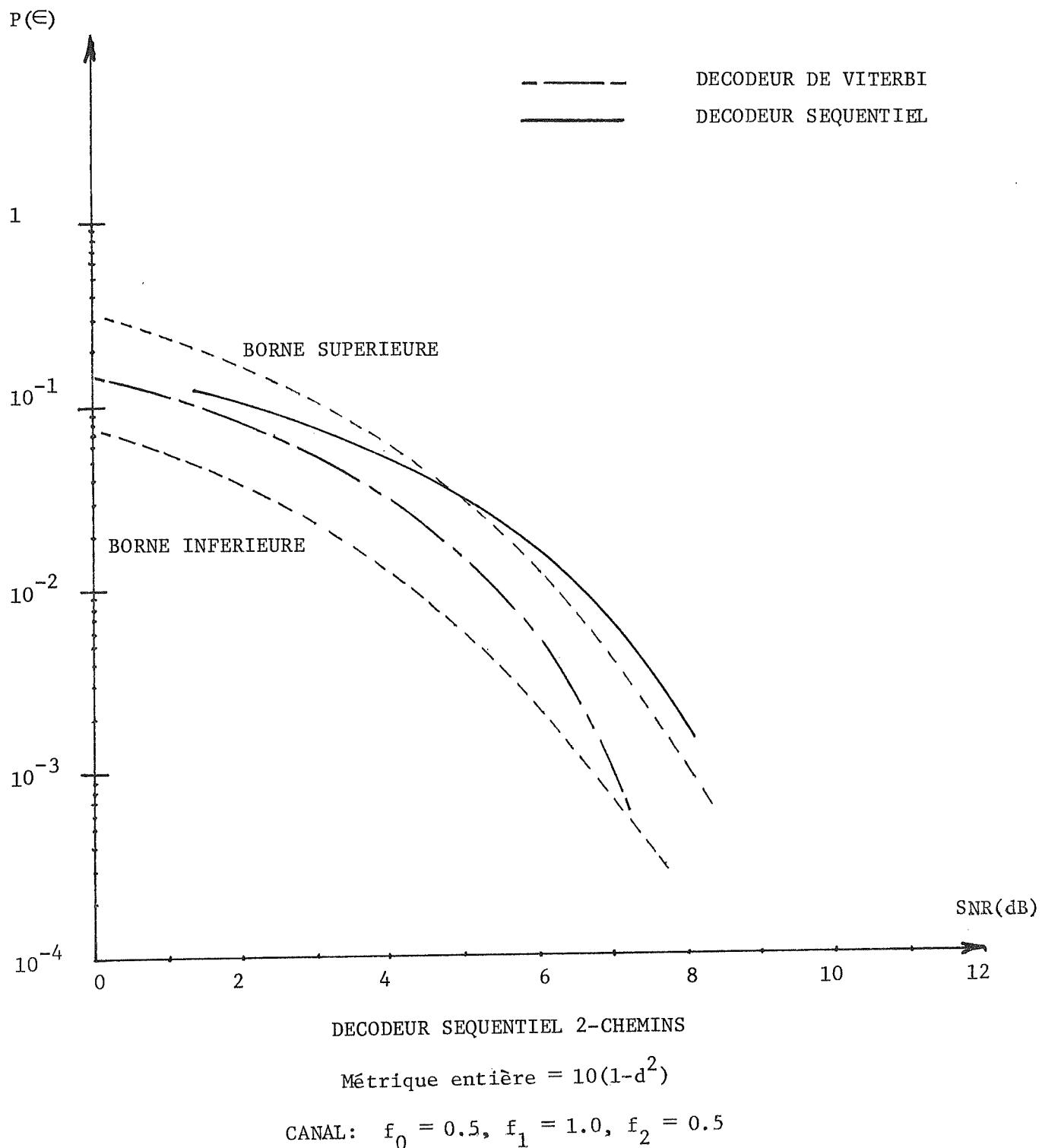


FIGURE 4.19

PROBABILITE D'ERREUR V.S. SNR(dB)

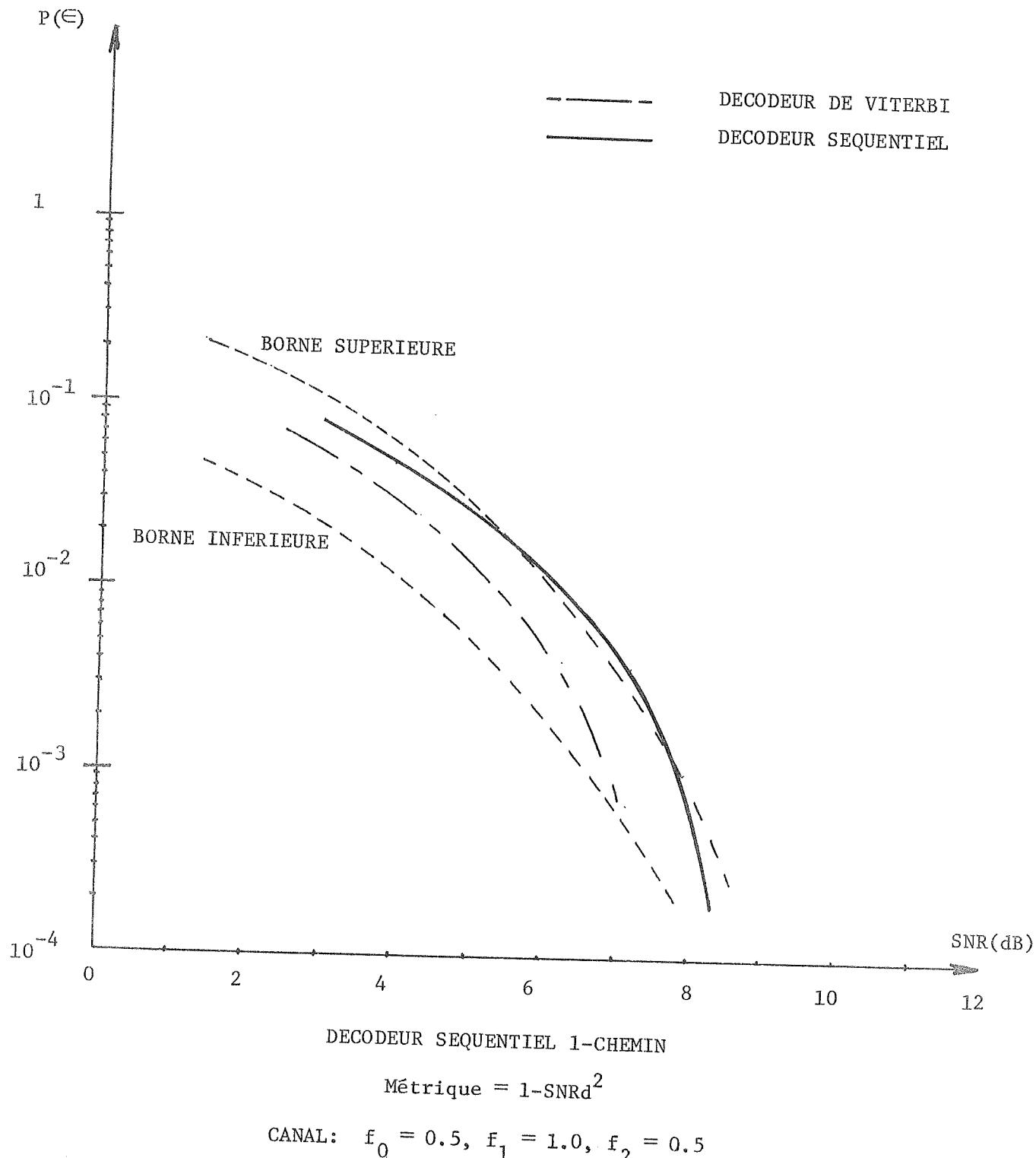


FIGURE 4.20

PROBABILITE D'ERREUR V.S. SNR(dB)

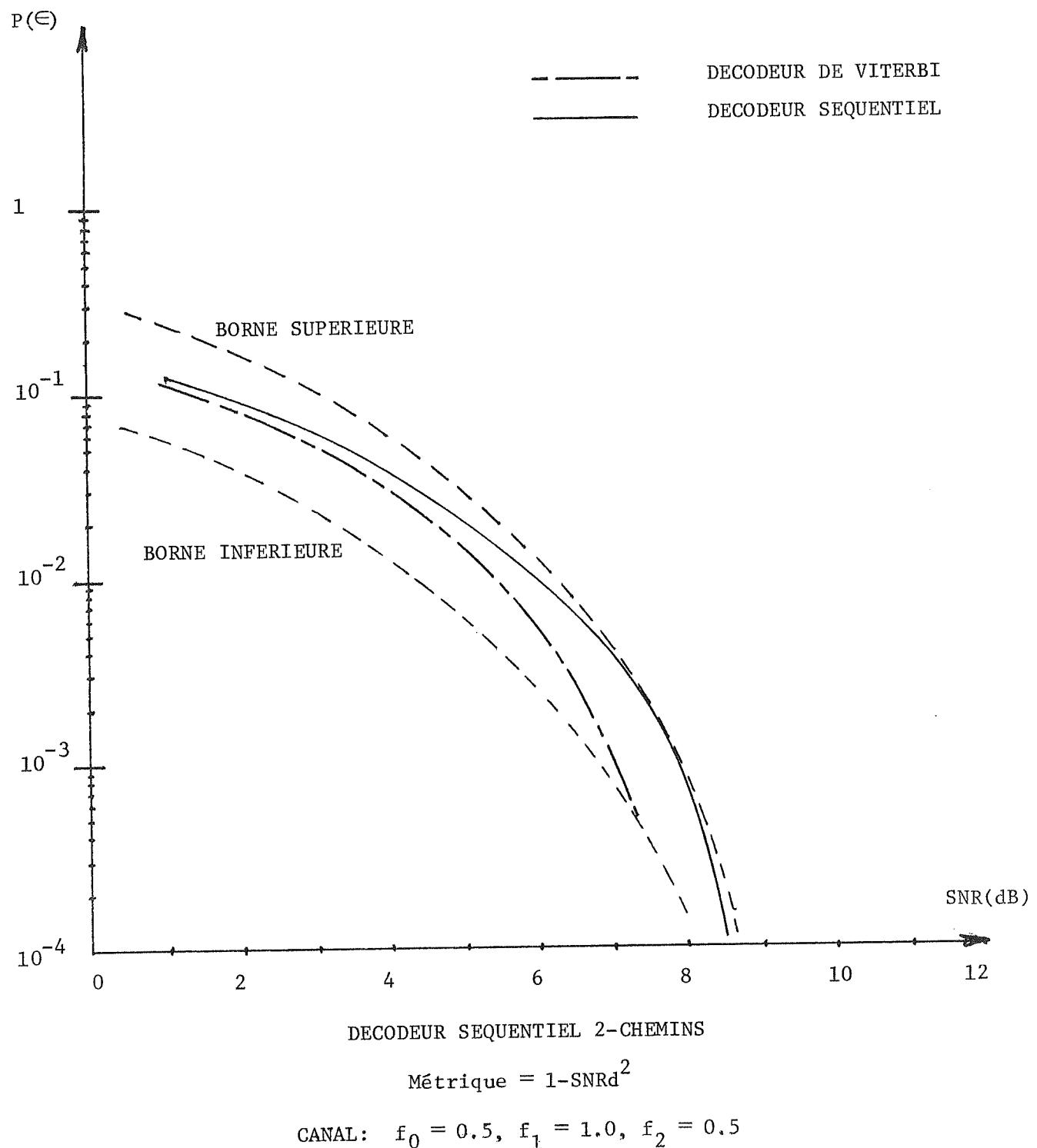


FIGURE 4.21

PROBABILITE D'ERREUR V.S. SNR(dB)

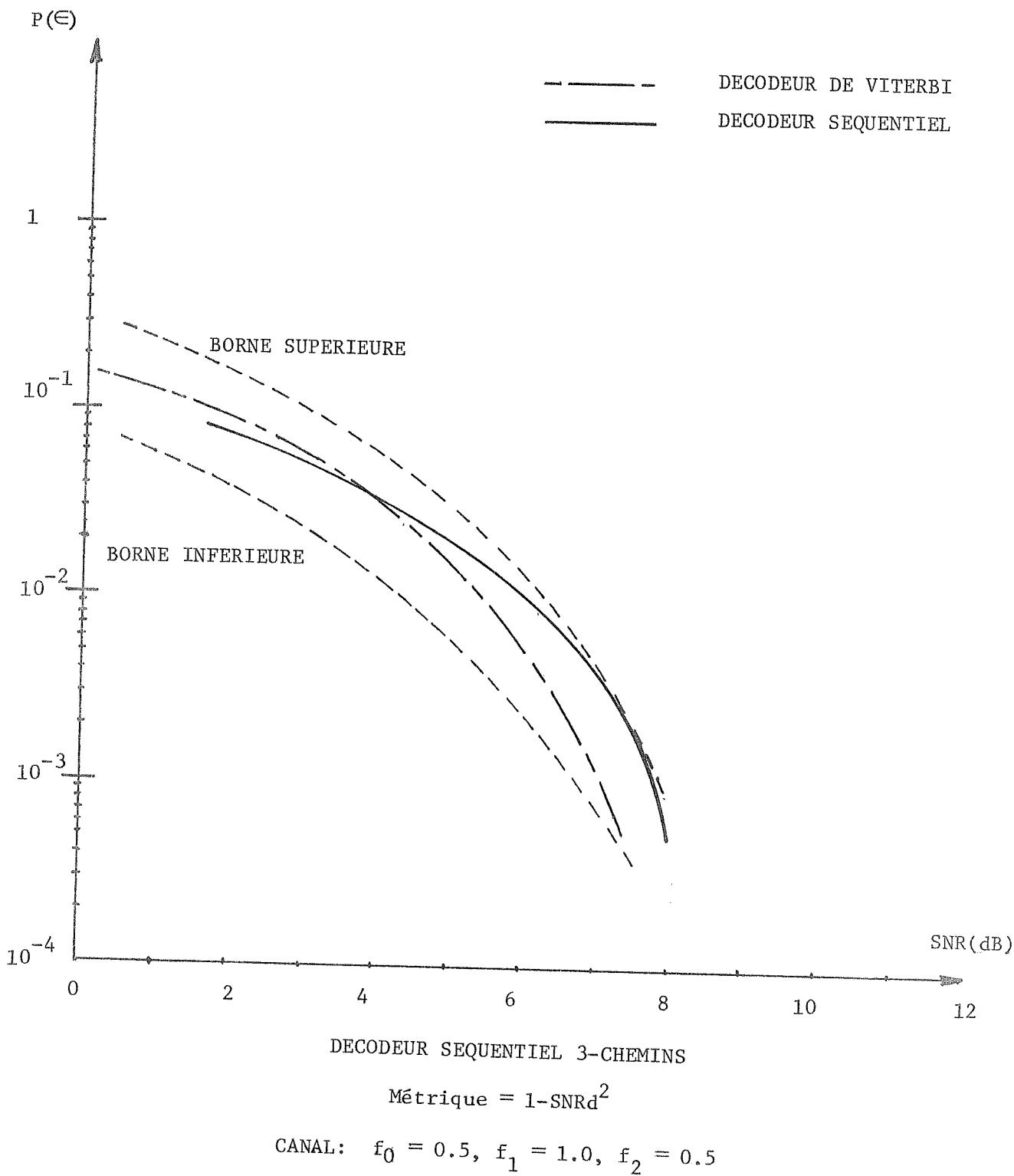


FIGURE 4.22

CANAL 0.5, 1, 0.5

- 10,000 symboles transmis
- Nombre de calculs effectués par un décodeur de Viterbi = 40,000
- $(SNR)_{dB} = 3dB$

Décodeurs Séquentiels Multi-Chemins

$$\boxed{\text{Métrique entière } M = 10(1-d^2)}$$

	Nombre de calculs théorique (sans retours en arrière)	Nombre de calculs réel (avec retours en arrière)	Supplément causé par les retours en arrière	Rapport du nombre de calculs réel avec celui du Viterbi
1-chemin	10,000	11,581	15.8%	29.0%
2-chemins	19,980	20,212	1.16%	50.5%
$\boxed{\text{Métrique réelle } = M_1 = 1.0 - SNRd^2}$				
1-chemin	10,000	12,865	28.7%	32.2%
2-chemins	19,975	21,053	5.4%	52.6%
3-chemins	29,900	30,630	2.44%	76.6%

TABLEAU 5

Pour compléter les essais une comparaison est faite avec les résultats de Crozier (et al., 1979). Le canal étudié a une réponse impulsionnelle donnée par

$$f(t) = \begin{cases} 0 & t \leq 0 \\ A(1 - e^{-2\pi F_o t}) & 0 \leq t \leq T \\ f(t) e^{-2\pi F_o (t-T)} & t \geq T \end{cases} \quad (4.61)$$

où F_o est la fréquence de coupure du canal, A est une constante, et $R = T^{-1}$ est le taux de transmission de la séquence d'entrée. Deux valeurs sont alors considérées pour le taux de transmission;

$$R_1 = 4F_o \quad (4.62)$$

$$R_2 = 6F_o \quad (4.63)$$

Si les symboles d'entrée sont d'énergie unitaire alors A est fixé à 1. Crozier suppose qu'un codeur convolutionnel avec une longueur de contrainte de 3 représente ce canal avec assez de précision. Avec $t = T$, $2T$ et $3T$ ceci donne pour les coefficients de pondération du codeur,

$$R_1 = 4F_o \rightarrow f_0 = 0.792, f_1 = 0.165, f_2 = 0.034 \quad (4.64)$$

$$R_2 = 6F_o \rightarrow f_0 = 0.649, f_1 = 0.228, f_2 = 0.080 \quad (4.65)$$

A partir de cela Crozier évalue la performance d'un décodeur de Viterbi de mémoire 2, en traçant la probabilité d'erreur de bit en fonction du rapport signal sur bruit. Le rapport signal sur bruit

qu'utilise Crozier est le suivant,

$$\text{SNR} = \frac{1}{\sigma_N^2} \quad (4.66)$$

Pour pouvoir insérer les résultats de Crozier dans un graphique ayant comme abscisse le rapport signal à bruit de l'équation (4.53) soit,

$$\text{SNR} = \frac{1}{2\sigma_N^2} \quad (4.67)$$

il a fallu les décaler de 3dB vers la gauche. Le décodeur de Viterbi utilisé par Crozier est alors comparé avec un décodeur séquentiel à chemin unique utilisant la métrique réelle de l'équation (4.56). La figure 4.23 illustre le cas $R_1 = 4F_o$, et la figure 4.24 le cas $R_2 = 6F_o$. Il apparaît que le décodeur séquentiel, dans les deux cas, a une performance comparable à celle du Viterbi. Même en utilisant une métrique favorisant les retours en arrière, l'effort de calcul du décodeur séquentiel est inférieur à 27% de celui du décodeur de Viterbi dans les deux cas. De plus, avec le décodeur séquentiel, un nombre plus grand de coefficients de pondération peut être utilisé pour augmenter la précision de la représentation du canal; ceci deviendrait vite prohibitif avec un décodeur de Viterbi qui a une croissance exponentielle de l'effort de calcul avec la longueur de contrainte.

PROBABILITE D'ERREUR V.S. SNR(dB)

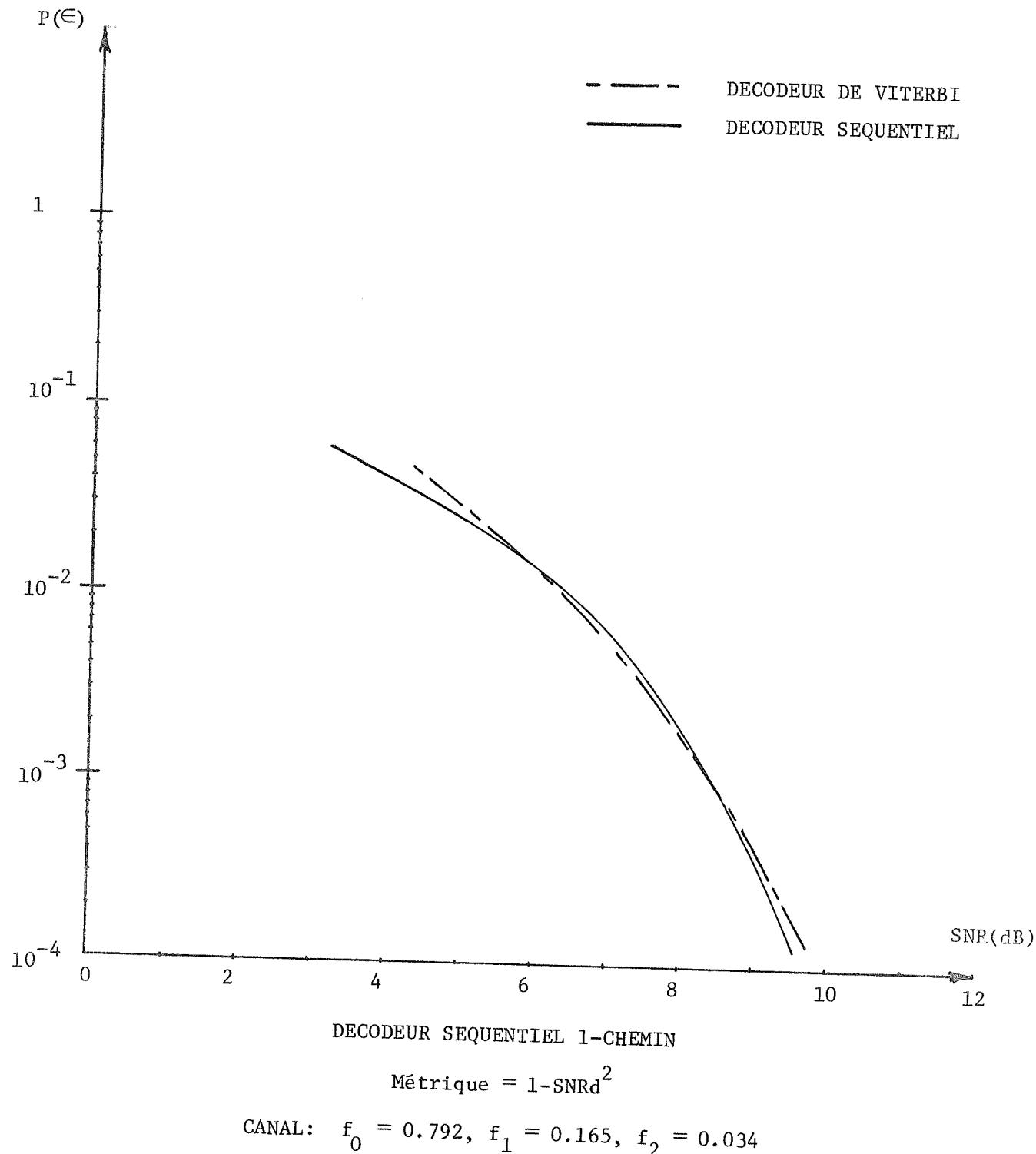


FIGURE 4.23

PROBABILITE D'ERREUR V.S. SNR(dB)

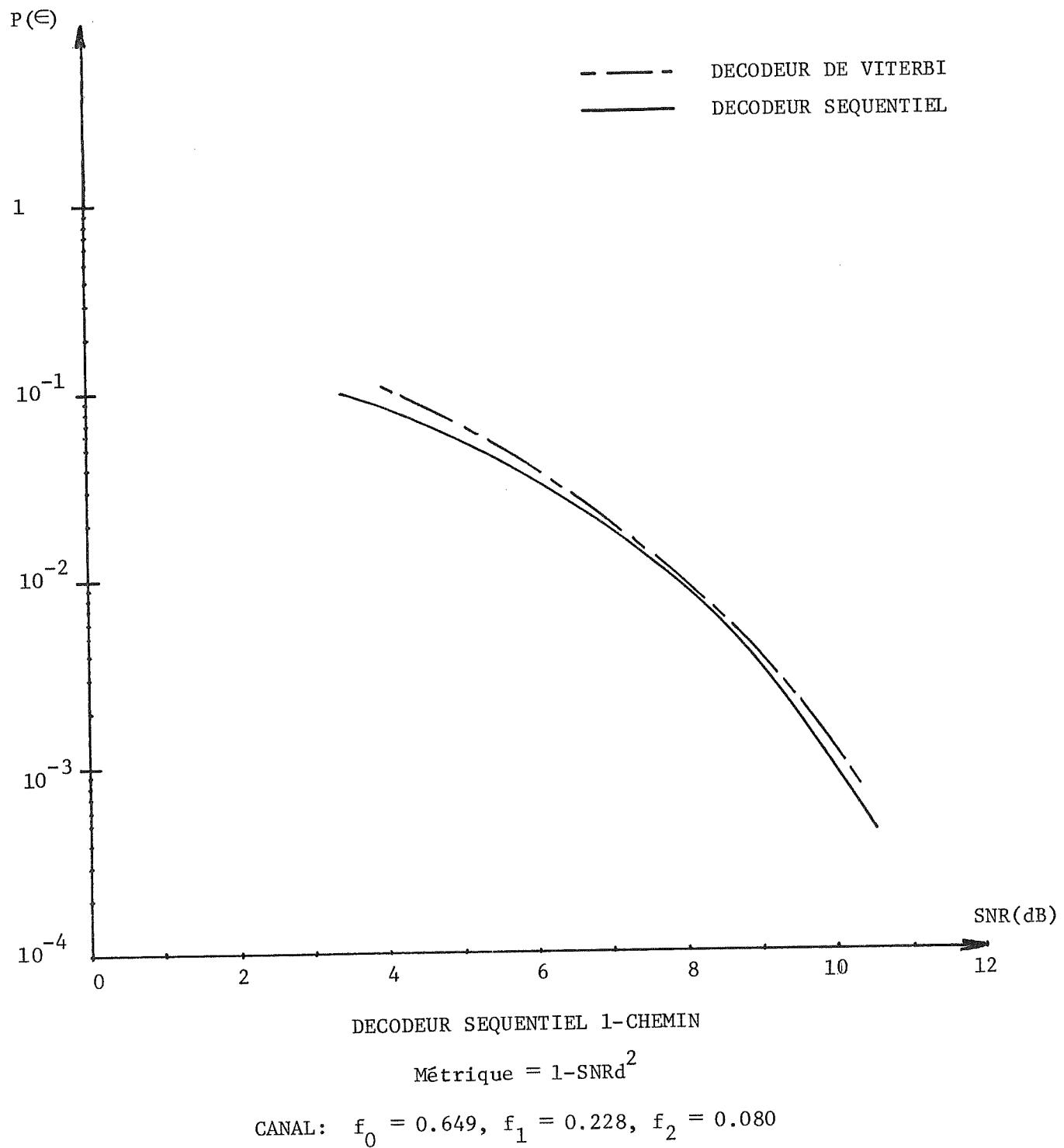


FIGURE 4.24

- 10,000 symboles transmis
- Nombre de calculs effectués par un décodeur de Viterbi = 40,000
- $(\text{SNR})_{\text{dB}} = 3\text{dB}$

Décodeurs Séquentiels Multi-Chemins

$$\text{Métrique} = 1 - \text{SNRd}^2$$

CANAL 0.792, 0.165, 0.034

	Nombre de calculs théorique (sans retours en arrière)	Nombre de calculs réel (avec retours en arrière)	Supplément causé par les retours en arrière	Rapport du nombre de calculs réel avec celui du Viterbi
1-chemin	10,000	10,405	4.05%	26.0%
	CANAL 0.649, 0.228, 0.080			
1-chemin	10,000	10,685	6.85%	26.7%

TABLEAU 6

Le tableau qui suit donne des exemples de nombres moyens de calculs obtenus avec les divers essais utilisant comme métrique $M_1 = 1 - \text{SNR}_d^2$. Les quatres premiers canaux étudiés dans ce chapitre sont considérés. Avec ce tableau il est vérifié que le nombre moyen de calculs pour un décodeur séquentiel N-chemins tend vers N pour des rapports signal sur bruit élevés.

nombre de chemins	Longueur 7			Longueur 5			Longueur 4			Longueur 3	
	1	3	7	1	3		1	3	5	1	2
SNR(dB)											
3.00	2.36	3.89	7.69	1.67	3.28		1.45	3.26	5.26	1.29	2.11
4.00	1.84	3.63	7.60	1.62	3.20		1.38	3.16	5.12	1.23	2.05
7.50	1.83	3.42	7.48	1.36	3.05		1.29	3.09	5.02	1.08	2.01
9.00	1.82	3.24	7.35	1.22	3.02		1.23	3.05	5.00	1.04	2.00
11.0	1.80	3.06	7.33	1.09	3.00		1.09	3.01	-	1.01	-
13.50	1.50	3.01	7.05	1.02	-		1.05	3.00	-	1.00	-
16.0	1.07	3.00	7.00	1.01	-		1.02	-	-	-	-
17.0	1.00	-	-	1.00	-		1.00	-	-	-	-

NOMBRES MOYENS DE CALCULS

AVEC LA METRIQUE $1 - \text{SNR}_d^2$

TABLEAU 7

CHAPITRE 5

CONCLUSION

Les essais sur ordinateur ont montré que les décodeurs séquentiels multi-chemins pouvaient efficacement corriger des cas d'interférence entre symboles même quand l'étalement d'un symbole affecte de nombreux voisins. Les performances de ces décodeurs sont compétitives avec celles du décodeur optimal de Viterbi avec l'avantage dans tous les cas d'un moins grand nombre de calculs et même en général, ce qui peut être décisif, d'un nombre considérablement moindre de calculs. Cet avantage permet aux décodeurs séquentiels de s'appliquer à des canaux très dispersifs pour lesquels l'utilisation d'un décodeur de Viterbi est prohibitive, compte tenu de la croissance exponentielle de l'effort de calcul de ce dernier en fonction de la dispersivité d'un canal. Pour des canaux représentés par des codeurs à faibles longueurs de contrainte, comme le canal (0.5, 1, 0.5) décrit à la section 4.10, l'emploi du décodeur de Viterbi peut être envisagé pour une performance optimale bien qu'il soit possible, même dans ces cas là, avec un décodeur séquentiel d'obtenir de bonnes performances avec toujours l'avantage d'un effort de calcul réduit. Par contre les essais décrits à la section 4.7 pour un canal de longueur 7 illustre un cas où l'emploi d'un décodeur de Viterbi serait fortement désavantageux quant au nombre de calculs.

Dans ce travail, des décodeurs séquentiels multi-chemins simples,

sans détection de reconvergences et munis de métriques empiriques ont révélé le potentiel d'application de ce type de décodeurs au problème de l'interférence entre symboles. Les essais présentés se limitaient à des canaux dont la dispersivité impliquait un maximum de 7 symboles. Toutefois le programme de simulation conçu conjointement avec ce travail pourrait facilement se modifier pour traiter des canaux plus dispersifs dans des travaux ultérieurs. Des décodeurs séquentiels plus sophistiqués qui prolongent un nombre variable de chemins en fonction de l'intensité du bruit aléatoire, et qui peuvent détecter les reconvergences, devraient également faire l'objet de travaux ultérieurs. De plus, des travaux pourraient porter sur l'obtention de métriques optimales pour les décodeurs séquentiels multi-chemins adaptés aux canaux dispersifs. Ceci donnerait encore plus de poids au choix de tels décodeurs à la place du décodeur de Viterbi pour la correction de l'interférence entre symboles.

BIBLIOGRAPHIE

- 1) CROZIER, S. et al., (1979), "Microprocessor Based Implementation and Testing of a Viterbi Detector", National Telecommunication Conference, Novembre 1979.
- 2) DUFOUR, M., (1980), "Décodage séquentiel: simulation et analyse de la file d'attente et de la pile", Mémoire de maîtrise en ingénierie, Ecole Polytechnique de Montréal, septembre 1978.
- 3) FANO, R.M., (1963), "A Heuristic of Probabilistic Decoding", IEEE Transactions on Information Theory, Vol. IT-9, pp. 64-73, Avril 1963.
- 4) FORNEY, G.D., (1972), "Maximum-Likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference", IEEE Transactions on Information Theory, Vol. IT-18, No. 3, pp. 363-378, Mai 1972.
- 5) FORNEY, G.D., (1972-b), "Lower Bounds on Error Probability in the Presence of Large Intersymbol Interference", IEEE Transactions on Communications, pp. 76-77, Février 1972.
- 6) FORNEY, G.D., (1973), "The Viterbi Algorithm", Proceedings of the IEEE, pp. 268-277, Mars 1973.
- 7) HACCOUN, D., (1974), "Multiple-Path Stack Algorithms for Decoding Convolutional Codes", Ph.D. dissertation, Université McGill, Octobre 1974.

- 8) HACCOUN, D. et FERGUSON, M.J., (1975), "Generalized Stack Algorithms for Decoding Convolutional Codes", IEEE Transactions on Information Theory, Vol. IT-21, No. 6, pp. 638-651, Novembre 1975.
- 9) IBM Staff, (1959), "Random Number Generation and Testing", IBM Data Processing Techniques Report, C20-8011, 1959.
- 10) JANELLE, A. et HACCOUN, D., (1978), "Décodage Adaptatif des Codes Convolutionnels", Rapport technique EP78-R-18, Ecole Polytechnique, Juin 1978.
- 11) JELINEK, F., (1969), "A fast sequential Decoding Algorithm using a stack", IBM Journal of Research and Development, Vol. 13, pp. 675-685, Novembre 1969.
- 12) KAWAS-KALEH, G., (1977), "Réception de Données par Décodage Récurrent en Présence de Bruit et d'Interférence Entre Symboles", Thèse de Doctorat, ENST-E-77001, Mars 1977.
- 13) LINDSEY, W.C. et SIMON, M.K., (1973), "Telecommunication Systems Engineering", Information and System Sciences Series, Thomas Kailath, editor, Prentice-Hall inc., New Jersey, 1973.
- 14) MAGEE, F.R. Jr., (1972), "Filtering Techniques for Time Dispersive Channels", Ph.D Thesis, Northeastern University, Boston Mass., Août 1972.
- 15) MAGEE, F.R. Jr., et PROAKIS, J.G., (1973), "Adaptive Maximum-Likelihood Sequence Estimation for Digital Signaling in the Presence of Intersymbol Interference", IEEE Transactions on Information Theory, Vol. IT-19, pp. 120-124, Janvier 1973.

- 16) MAGEE, F.R. Jr. et PROAKIS, J.G., (1973-b), "An Estimate of the Upper Bound on Error Probability for Maximum-Likelihood Estimation on Channels Having a Finite-Duration Pulse Response", IEEE Transactions on Information Theory, pp. 699-702, Septembre 1973.
- 17) MASSEY, J.L. et al, (1972), "Certain Infinite Markov Chains and Sequential Decoding", Discrete Mathematics, Vol. 3, pp. 163-175, Septembre 1972.
- 18) McLANE, P.J., (1980), "A residual Intersymbol Interference Error Bound for Truncated-State Viterbi Detectors", IEEE Transactions on Information Theory, Vol. IT-26, No. 5, pp. 548-553, Septembre 1980.
- 19) PROAKIS, J.G. et MILLER, J.H., (1969), "An Adaptive Receiver for Signaling Through Channels With Intersymbol Interference", IEEE Transactions on Information Theory, Vol. IT-15, pp. 484-497, Juillet 1969.
- 20) PROAKIS, J.G., (1978), "Advances in Equalization for Intersymbol Interference", Communication Systems and Random Process Theory, NATO Advanced Studies Institute Series, Skwirzyns J. editor, Sijthoff and Noordhoff publishers, Août 1978.
- 21) VITERBI, A.J., (1967), "Error Bounds for Convolutional Codes and Asymptotically Optimum Decoding Algorithm", IEEE Transactions on Information Theory, Vol. IT-13, pp. 260-269, Avril 1967.
- 22) WOZENCRAFT, J.M. et JACOBS, J.M., (1965), "Principles of Communications Engineering", John Wiley and Sons Inc., New Yo 1965.

- 23) WYLIE, C.R. Jr., (1966), "Advanced Engineering Mathematics",
New York: McGraw-Hill, 1966, ch. 11.
- 24) ZIGANGIROV, K., (1966), "Some Sequential Decoding Procedures",
Problemi Peredachi Informatsii, Vol. 2, No. 4, pp. 13-
15, 1966.

- 8) HACCOUN, D. et FERGUSON, M.J., (1975), "Generalized Stack Algorithms for Decoding Convolutional Codes", IEEE Transactions on Information Theory, Vol. IT-21, No. 6, pp. 638-651, Novembre 1975.
- 9) IBM Staff, (1959), "Random Number Generation and Testing", IBM Data Processing Techniques Report, C20-8011, 1959.
- 10) JANELLE, A. et HACCOUN, D., (1978), "Décodage Adaptatif des Codes Convolutionnels", Rapport technique EP78-R-18, Ecole Polytechnique, Juin 1978.
- 11) JELINEK, F., (1969), "A fast sequential Decoding Algorithm using a stack", IBM Journal of Research and Development, Vol. 13, pp. 675-685, Novembre 1969.
- 12) KAWAS-KALEH, G., (1977), "Réception de Données par Décodage Récurrent en Présence de Bruit et d'Interférence Entre Symboles", Thèse de Doctorat, ENST-E-77001, Mars 1977.
- 13) LINDSEY, W.C. et SIMON, M.K., (1973), "Telecommunication Systems Engineering", Information and System Sciences Series, Thomas Kailath, editor, Prentice-Hall inc., New Jersey, 1973.
- 14) MAGEE, F.R. Jr., (1972), "Filtering Techniques for Time Dispersive Channels", Ph.D Thesis, Northeastern University, Boston Mass., Août 1972.
- 15) MAGEE, F.R. Jr., et PROAKIS, J.G., (1973), "Adaptive Maximum-Likelihood Sequence Estimation for Digital Signaling in the Presence of Intersymbol Interference", IEEE Transactions on Information Theory, Vol. IT-19, pp. 120-124, Janvier 1973.

- 16) MAGEE, F.R. Jr. et PROAKIS, J.G., (1973-b), "An Estimate of the Upper Bound on Error Probability for Maximum-Likelihood Estimation on Channels Having a Finite-Duration Pulse Response", IEEE Transactions on Information Theory, pp. 699-702, Septembre 1973.
- 17) MASSEY, J.L. et al, (1972), "Certain Infinite Markov Chains and Sequential Decoding", Discrete Mathematics, Vol. 3, pp. 163-175, Septembre 1972.
- 18) McLANE, P.J., (1980), "A residual Intersymbol Interference Error Bound for Truncated-State Viterbi Detectors", IEEE Transactions on Information Theory, Vol. IT-26, No. 5, pp. 548-553, Septembre 1980.
- 19) PROAKIS, J.G. et MILLER, J.H., (1969), "An Adaptive Receiver for Signaling Through Channels With Intersymbol Interference", IEEE Transactions on Information Theory, Vol. IT-15, pp. 484-497, Juillet 1969.
- 20) PROAKIS, J.G., (1978), "Advances in Equalization for Intersymbol Interference", Communication Systems and Random Process Theory, NATO Advanced Studies Institute Series, Skwirzynski, J. editor, Sijthoff and Noordhoff publishers, Août 1978.
- 21) VITERBI, A.J., (1967), "Error Bounds for Convolutional Codes and Asymptotically Optimum Decoding Algorithm", IEEE Transactions on Information Theory, Vol. IT-13, pp. 260-269, Avril 1967.
- 22) WOZENCRAFT, J.M. et JACOBS, J.M., (1965), "Principles of Communications Engineering", John Wiley and Sons Inc., New York 1965.

ÉCOLE POLYTECHNIQUE DE MONTRÉAL



3 9334 00289168 5