CODES DE CORRECTION D'ERREURS POUR LA TRANSMISSION DE LA TÉLÉVISION NUMÉRIQUE† – PHASE 2

RAPPORT FINAL

par

Jean-Yves Chouinard, Daniel Cayer et Marc Trichard

Département de génie électrique

Université d'Ottawa

pour

Industrie Canada

Contrat 67qsg-5-6181

Approvisionnement et services

Gouvernement du Canada

Mars 1996

†Ce projet de recherche est financé en partie par le Programme des centres d'excellence de langue française
d'Industrie Canada.

CODES DE CORRECTION D'ERREURS POUR LA TRANSMISSION DE LA TÉLÉVISION NUMÉRIQUE† – PHASE 2

RAPPORT FINAL

par

Jean-Yves Chouinard, Daniel Cayer et Marc Trichard

Département de génie électrique

Université d'Ottawa

pour

Industrie Canada

Contrat 67qsg-5-6181

Approvisionnement et services

Gouvernement du Canada

Mars 1996

†Ce projet de recherche est financé en partie par le Programme des centres d'excellence de langue française d'Industrie Canada.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Description du projet de recherche

La télévision avancée (TVA) est sur le point de résulter en une implémentation entièrement numérique. En Amérique du Nord, l'ACATS[1] (*Advisory Committee on Advanced Television Services*) ainsi que la *Grand Alliance*[2] étudient les avantages et désavantages des différents systèmes TVA proposés.

Les critères de fiabilité de la télévision haute définition (TVHD) requièrent un taux de transmission brut de l'ordre de 1.2 Gbits/s. Cette information doit être compressée à l'aide de codeur de source vidéo. La technologie actuelle permet d'obtenir des taux de bits d'environ 15 à 30 Mbits/s par l'exploitation des redondances spatiale et temporelle du signal de source TVHD, et ce, tout en maintenant une qualité d'image suffisante pour les critères de la télévision avancée.

Toutefois, ces exigences nécessitent l'utilisation de techniques de codage de canal très efficaces, car le fait de compresser le signal source par un facteur aussi élevé que 60:1 et 100:1 conduisent à la *propagation* des erreurs de transmission sur plusieurs trames, ou images vidéo, et ce, en dépit des techniques de récupération des erreurs implémentées au codeur et au décodeur de source.

Enfin, le taux de transmission des données numériques, incluant les informations vidéo, audio et de commande, est de 18 à 20 Mbits/s [WC94]. En Amérique du Nord ce signal doit être contenu dans une largeur de bande NTSC pour la télédiffusion terrestre (en pratique,

[1] L'ACATS est constituée des principaux intervenants des secteurs publics, parapublics et privés de l'Amérique du Nord.

[2] La *Grand Alliance* est formée de compagnies privées impliquées dans le développement de la transmission de la télévision numérique, telles que: AT&T, GI, MIT, Philips, Thomson, David Sarnoff et Zenith.

cela se traduit plutôt par une largeur de bande de 5 à 5.5 MHz afin de prévenir l'interférence avec les canaux adjacents). L'efficacité spectrale d'un tel système de télécommunication doit donc être d'environ 4 bits/s/Hz, ce que l'on peut obtenir avec des techniques de modulation multi-niveaux telles que la modulation d'amplitude 16-QAM.

Si l'on tolère une erreur de transmission par minute [HLP93], alors on doit pouvoir obtenir un taux d'erreur de l'ordre de 10^{-9}. Si on ignore l'utilisation éventuelle de codes de correction d'erreur, on devrait, pour arriver à un si faible taux d'erreur, transmettre le signal TVA avec une puissance de transmission prohibitivement grande et ainsi rendre la co-existence avec les signaux de télévision analogiques NTSC existants impossible. On doit donc, en conclusion, réaliser des codes de correction d'erreurs très puissants. Pour ce faire, l'emploi de techniques de codage basées sur les codes concaténés est requis.

1.2 Objectifs et plan du rapport

Les objectifs poursuivis dans le cadre de la phase 2 de ce projet de recherche se résument à la poursuite de l'étude comparative de la performance de codes de correction d'erreurs déjà proposés par les membres de la *Grand Alliance*. On analyse plus spécifiquement la performance, en terme de probabilités d'erreurs par bit et par symbole, de la modulation codée TCM (*Trellis Coded Modulation*) et plus particulièrement les codes TCM-QAM à 32 niveaux (i.e., TCM-32QAM) et TCM-64QAM à taux de code 2/3-2/3 (TCM-64QAM 2/3-2/3), ainsi que les codes concaténés Reed-Solomon avec la modulation codée TCM-QAM. On décrit le fonctionnement d'un programme de simulation développé à l'aide des logiciels *MATLAB* et *SIMULINK*, permettant de représenter les différents signaux dans une chaîne de télécommunication ATV. Ces logiciels ont été choisis en raison de leur flexibilité, leur disponibilité et leur interface graphique. Les techniques d'estimation du canal de propagation radioélectrique appliquées aux techniques de codage de canal permettraient d'améliorer encore la fiabilité de la liaison en exploitant les statistiques du canal de télévision numérique.

Le rapport final est organisé comme suit. Au chapitre 2, on considère les codes correcteurs d'erreurs aléatoires et de groupements d'erreurs concaténés TCM-QAM et Reed-Solomon. On y rappelle les principaux résultats des simulations obtenus au cours de la phase 1 du projet et présente de nouveaux résultats relatifs à la performance de ces codes Reed-Solomon concaténés avec la modulation codée TCM-QAM. Le chapitre suivant (chapitre 3) présente un programme de simulation en bande de base basé développé avec les logiciels *MATLAB* et *SIMULINK* pour la deuxième phase du projet. Le fonctionnement des principaux modules du programme de simulation y est expliqué en détail. Le chapitre 4 conclut ce rapport en s'intéressant brièvement aux techniques d'estimation du canal pouvant s'appliquer au codage de canal: on y rapporte les travaux de recherche déjà effectués par d'autres chercheurs pour les canaux de télévision numérique.

Chapter 2

Performance des codes correcteurs d'erreurs concaténés TCM-QAM et Reed-Solomon

2.1 Introduction

On s'intéresse ici à la performance des codes correcteurs d'erreurs aléatoires et de groupements d'erreurs concaténés TCM-QAM à 32 niveaux et TCM-64QAM à taux de code 2/3-2/3 avec les codes Reed-Solomon $RS(208, 188, 10)$ et $RS(255, 239, 8)$. Les performances ont été évaluées avec des simulations Monte-Carlo en bande de base.

Afin d'obtenir une efficacité spectrale de 4 bits/s/Hz, un modulateur multi-niveaux QAM assigne à chaque sous-séquence d'information de $m = 4$ bits un point d'une constellation de $2^{m+1} = 32$ points. Le gain de codage d'un code interne TCM pour un canal gaussien est d'environ 3 dB pour un taux d'erreur par bit de 10^{-5}. Un code externe Reed-Solomon (N, K, t) doit donc permettre d'atteindre un taux d'erreur par bit de 10^{-4} ou moins. Les codes RS sont particulièrement adaptés à la présence de groupements d'erreurs que l'on retrouve typiquement à la sortie d'un décodeur de Viterbi, qui lui exploite la valeur réelle des symboles reçus du canal bruité (*soft decision decoding*). Cependant, les contraintes spectrales limitent le nombre de symboles de redondance à moins de 10%.

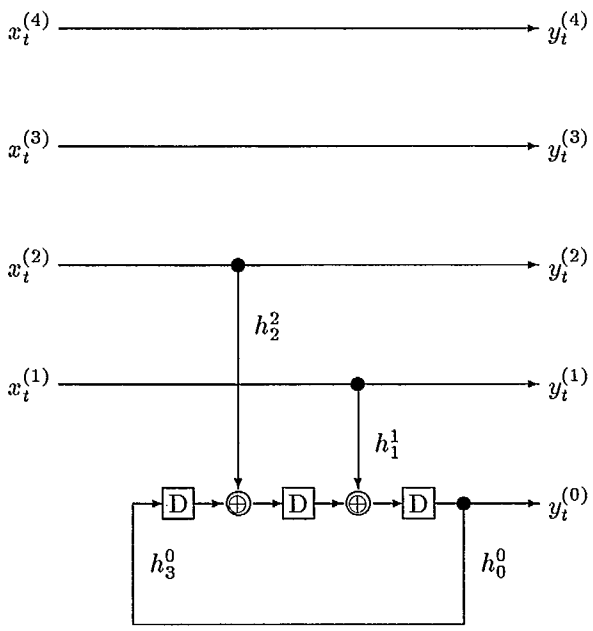

Figure 2.1: Structure du codeur en treillis TCM-32QAM (3,2,3).

2.2 Modulation codée multi-niveaux QAM

Les programmes de simulation sont conçus de manière à pouvoir choisir n'importe quel code TCM d'Ungerboeck [Ung87a, Ung87b]. Cependant, en raison du temps de calcul requis pour chacun de ces codes, nous avons sélectionnés deux codes représentatifs d'Ungerboeck: les codes TCM (3,2,3) et TCM (3,2,6). Le Tableau 2.1 donne les paramètres importants de ces deux codes TCM.

Table 2.1: Paramètres principaux des codes TCM (3,2,3) et (3,2,6).

Code TCM	$\frac{d^2_{libre}}{\Delta_0^2}$	α_{cod}	N_e	N_b	N_S	$\frac{d^2_{suivante}}{\Delta_0^2}$	β_{cod}	n_e	n_b	n_S
(3,2,3) m=4	5	$\frac{1}{2}$	13.43	24.26	40.30	6	$\frac{1}{20}$	3.25	3.98	4.88
(3,2,6) m=4	7	$\frac{7}{10}$	56 ($m\to\infty$)	-	-	-	-	-	-	-

Les paramètres du code TCM (3,2,6) sont tirés de la référence [Ung87b] alors que ceux du code TCM (3,2,3) ont été calculés et donnés dans [Tri95]. Il y a donc de légères différences car le TCM code (3,2,3) utilise les valeurs complètes et exactes des paramètres $N_e(d_{libre})$, $N_b(d_{libre})$ et $N_S(d_{libre})$, alors que pour l'autre code TCM, $N_{e\infty}(d_{libre})$ ne représente que la valeur limite de $N_e(d_{libre})$ lorsque le nombre de transitions parallèles tend vers l'infini, c'est-à-dire quand $m \to \infty$. On prévoit des différences de performances entre ces deux codes TCM, ceux-ci employant des mémoires de longueurs différentes (i.e., des longueurs de contrainte différentes). La structure des codeurs pour les codes TCM (3,2,3) et TCM (3,2,6) est illustrée aux Figures 2.1 et 2.2.

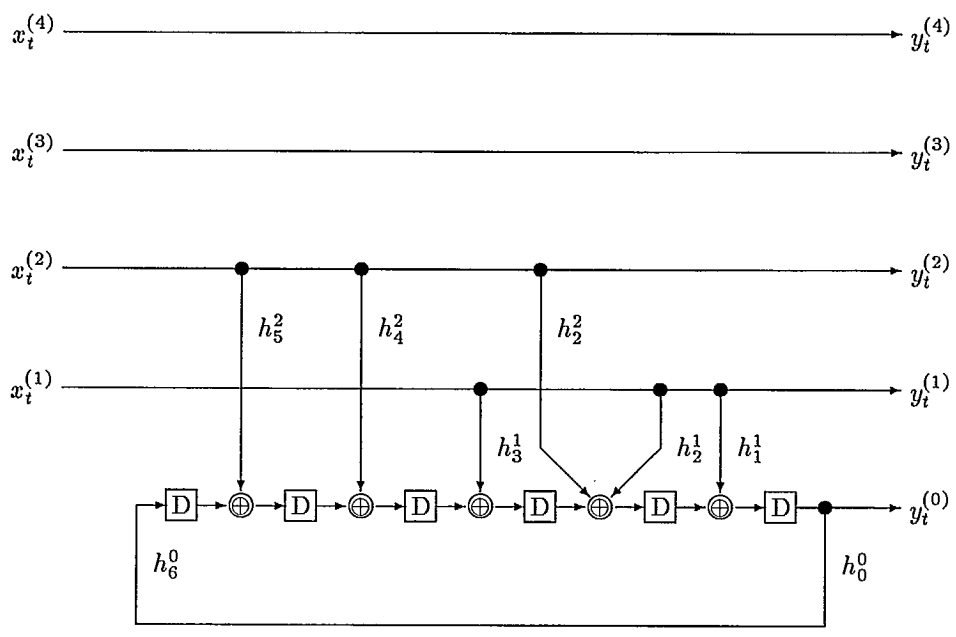

Figure 2.2: Structure du codeur en treillis TCM-32QAM (3,2,6).

La Figure 2.3 représente la constellation des signaux TCM 32-QAM. Il y a $m - k = 2$ bits non-codés ce qui résulte en 4 transitions parallèles sur le treillis et donne donc 8 sous-ensembles, ou sous-constellation de 4 points chacune. Les trois bits $y_t^{(2)}$, $y_t^{(1)}$ et $y_t^{(0)}$ spécifient l'une des huit sous-constellations alors que les deux autres, $y_t^{(4)}$ et $y_t^{(3)}$, déterminent lequel des quatre points de la sous-constellation est choisi, au temps t:

$$
\begin{array}{lcll}
y_t^{(2)}, y_t^{(1)}, y_t^{(0)} & = & 000 & \Longrightarrow \ \square \\
y_t^{(2)}, y_t^{(1)}, y_t^{(0)} & = & 001 & \Longrightarrow \ \star \\
y_t^{(2)}, y_t^{(1)}, y_t^{(0)} & = & 010 & \Longrightarrow \ \diamondsuit \\
y_t^{(2)}, y_t^{(1)}, y_t^{(0)} & = & 011 & \Longrightarrow \ \spadesuit \\
y_t^{(2)}, y_t^{(1)}, y_t^{(0)} & = & 100 & \Longrightarrow \ \heartsuit \\
y_t^{(2)}, y_t^{(1)}, y_t^{(0)} & = & 101 & \Longrightarrow \ \bullet \\
y_t^{(2)}, y_t^{(1)}, y_t^{(0)} & = & 110 & \Longrightarrow \ \bowtie \\
y_t^{(2)}, y_t^{(1)}, y_t^{(0)} & = & 111 & \Longrightarrow \ \clubsuit
\end{array}
$$

Dans la Figure 2.3, la distance la plus critique est $\Delta_3 = \sqrt{8}\Delta_0$, qui est la distance minimale entre les points d'une sous-constellation donnée. Pour les deux codes TCM (3,2,3) et TCM (3,2,6), cette distance est inférieure à la distance libre des codes: $d_{libre}(3,2,3) = \sqrt{5}\Delta_0$ et $d_{libre}(3,2,6) = \sqrt{7}\Delta_0$. La valeur de d_{libre} est obtenue pour deux *trajets* dans le treillis de Viterbi qui ne sont pas des transitions parallèles. Conséquemment, la probabilité d'un *évènement erreur* n'est pas exactement égale à la probabilité d'une erreur de symbole mais constitue plutôt une borne supérieure. Le calcul des paramètres N_S et N_b est donc particulièrement utile ici. Lorsque l'on considère la prochaine distance, on note que la performance du code TCM (3,2,3) code est affectée par la deuxième distance minimale $d_{suivante}$ sur le treillis, où $d_{suivante} = \sqrt{6}\Delta_0 < \Delta_3$. Cette distance $d_{suivante}$ est inconnue pour le code TCM (3,2,6), mais il est vraisemblable [Tri95] que $d_{suivante}(3,2,6)$ soit égale à $\sqrt{8}\Delta_0$ ou à $\sqrt{9}\Delta_0$.

Pour la modulation codée bi-unidimensionnelle TCM 64QAM à taux 2/3-2/3 il n'y a pas de transitions parallèles dans le treillis car le nombre total de bits à l'entrée du codeur TCM, c'est-à-dire $m = 2$, est le même que le nombre de bits entrant dans le codeur convolutionnel soit $k = 2$.

Les performance théoriques et obtenues par simulation en termes de taux d'erreur par symbole TCM-QAM et par bit dans un canal additif gaussien (AWGN) sont représentées aux Figures 2.4 et 2.5 respectivement pour les modulations codées TCM 32QAM et 64QAM 2/3-2/3 avec un code (3,2,3) code. On remarque une grande similarité entre les courbes théoriques et celles obtenues par simulation, et ce, pour les probabilités d'erreur par symbole $\mathcal{P}_S$ autant que pour les probabilités d'erreur par bit $\mathcal{P}_b$ pour des valeurs de rapport signal-à-bruit moyennes et grandes.

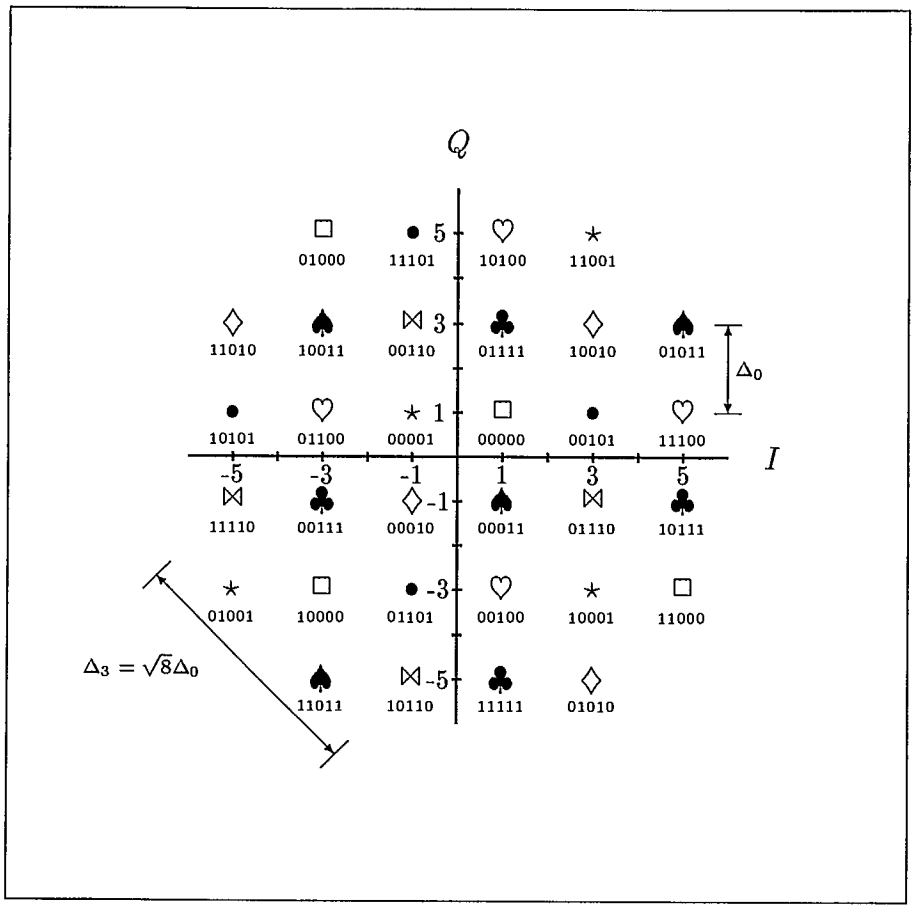

Figure 2.3: Constellation de signaux pour la modulation codée TCM 32-QAM.

La Figure 2.6 montre des courbes de probabilité d'erreur par symbole obtenues par simulation pour le TCM 32QAM et 64QAM 2/3-2/3 avec le code (3,2,6). Elle contient également une courbe asymptotique de probabilité d'erreur par *évènement* au premier ordre pour le TCM 32QAM.

Enfin, la Figure 2.7 montre les courbes simulées de probabilité d'erreur par bit pour les deux codes TCM (3,2,3) et (3,2,6), et ce pour les deux constellations TCM 32QAM et TCM 64QAM 2/3-2/3. On y note aisément le gain en performance obtenu lorsque l'on augmente la complexité des codeur et décodeur TCM de 8 états à 64 états.

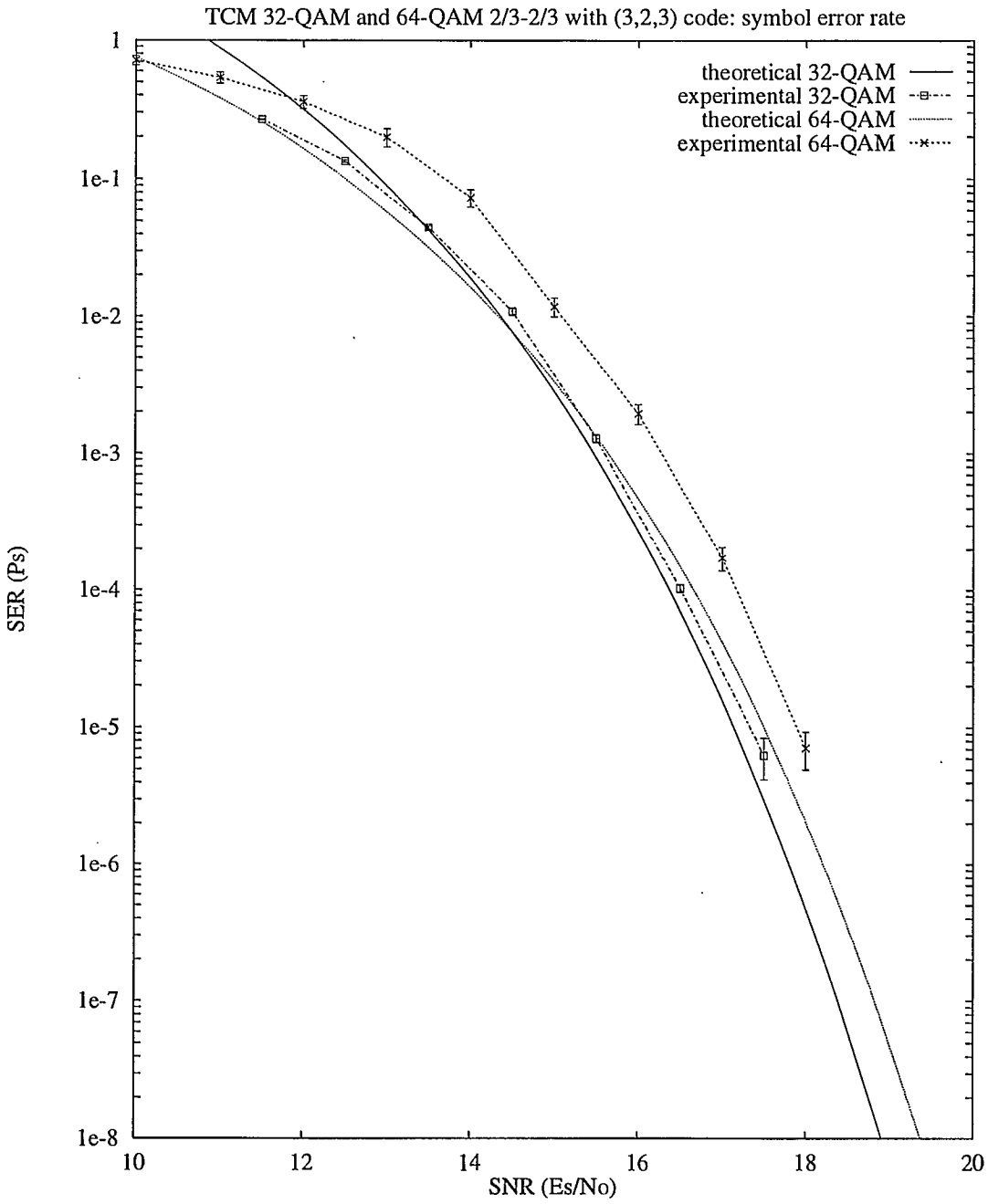

Figure 2.4: Probabilité d'erreur par symbole avec un code (3,2,3) pour la modulation codée TCM 32QAM et 64QAM 2/3-2/3.

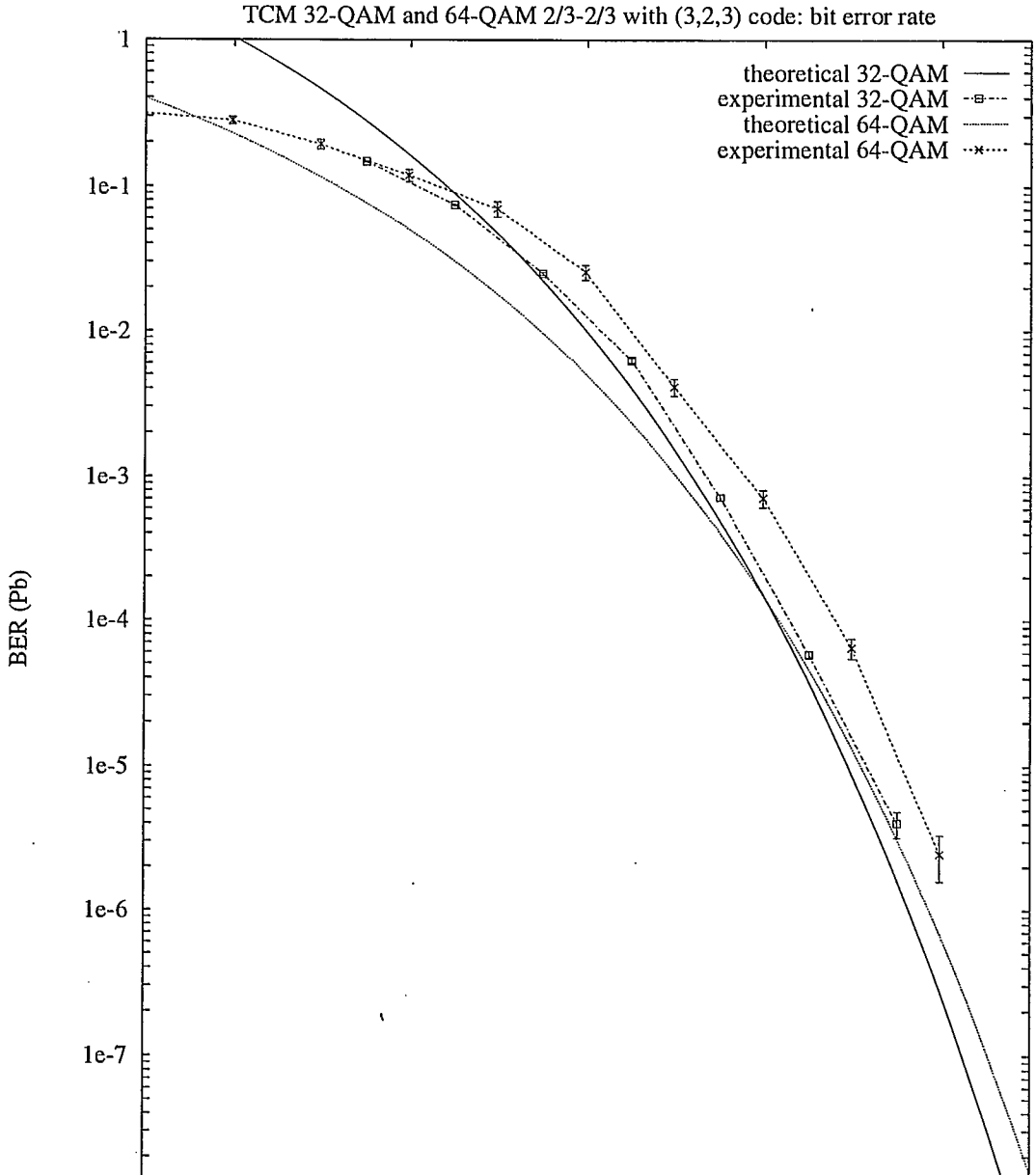

Figure 2.5: Probabilité d'erreur par bit avec un code (3,2,3) pour la modulation codée TCM 32QAM et 64QAM 2/3-2/3.

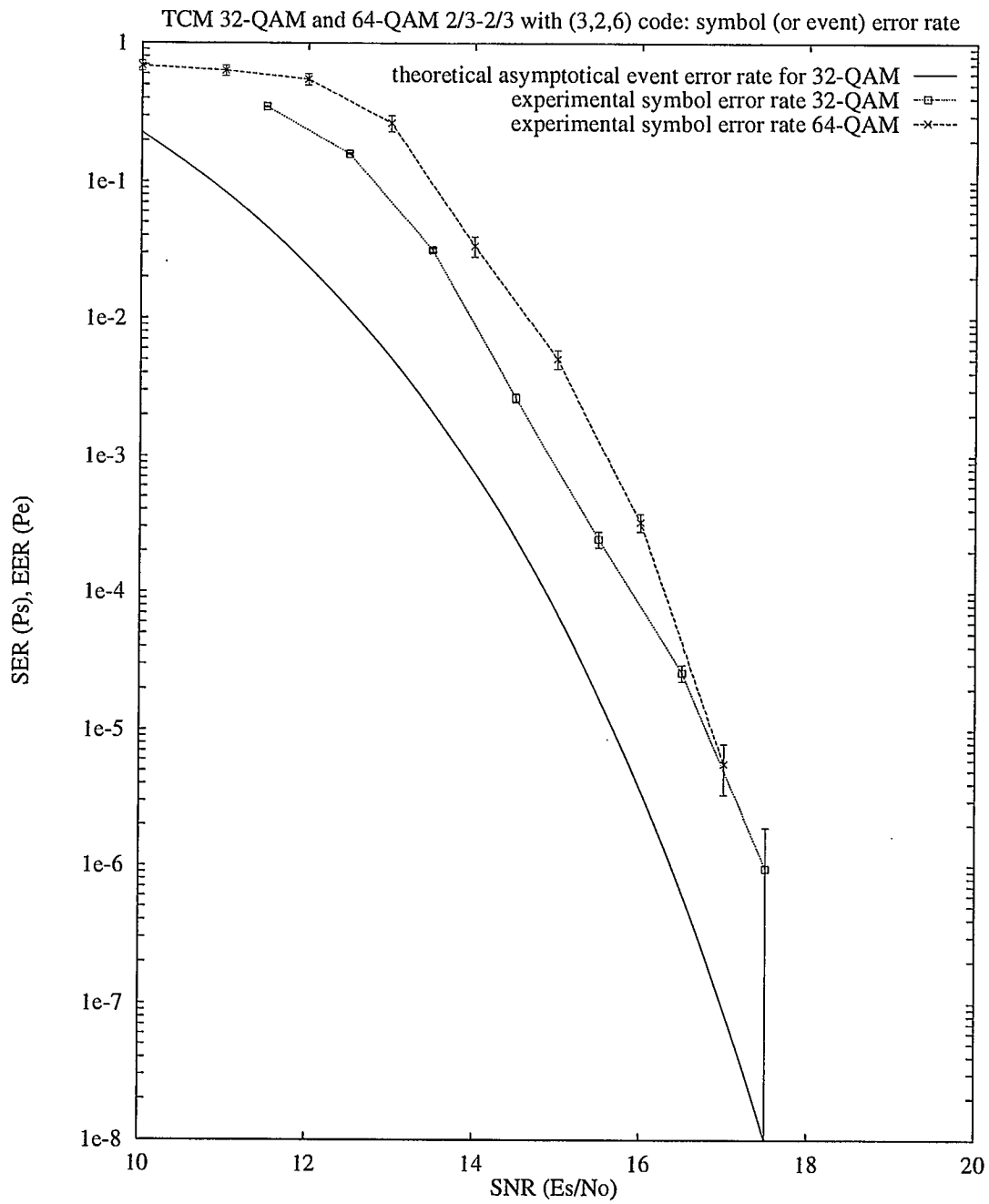

Figure 2.6: Probabilité d'erreur par symbole avec un code (3,2,6) pour la modulation codée TCM 32QAM et 64QAM 2/3-2/3 ainsi que la probabilité d'erreur par événement pour le TCM 32QAM.

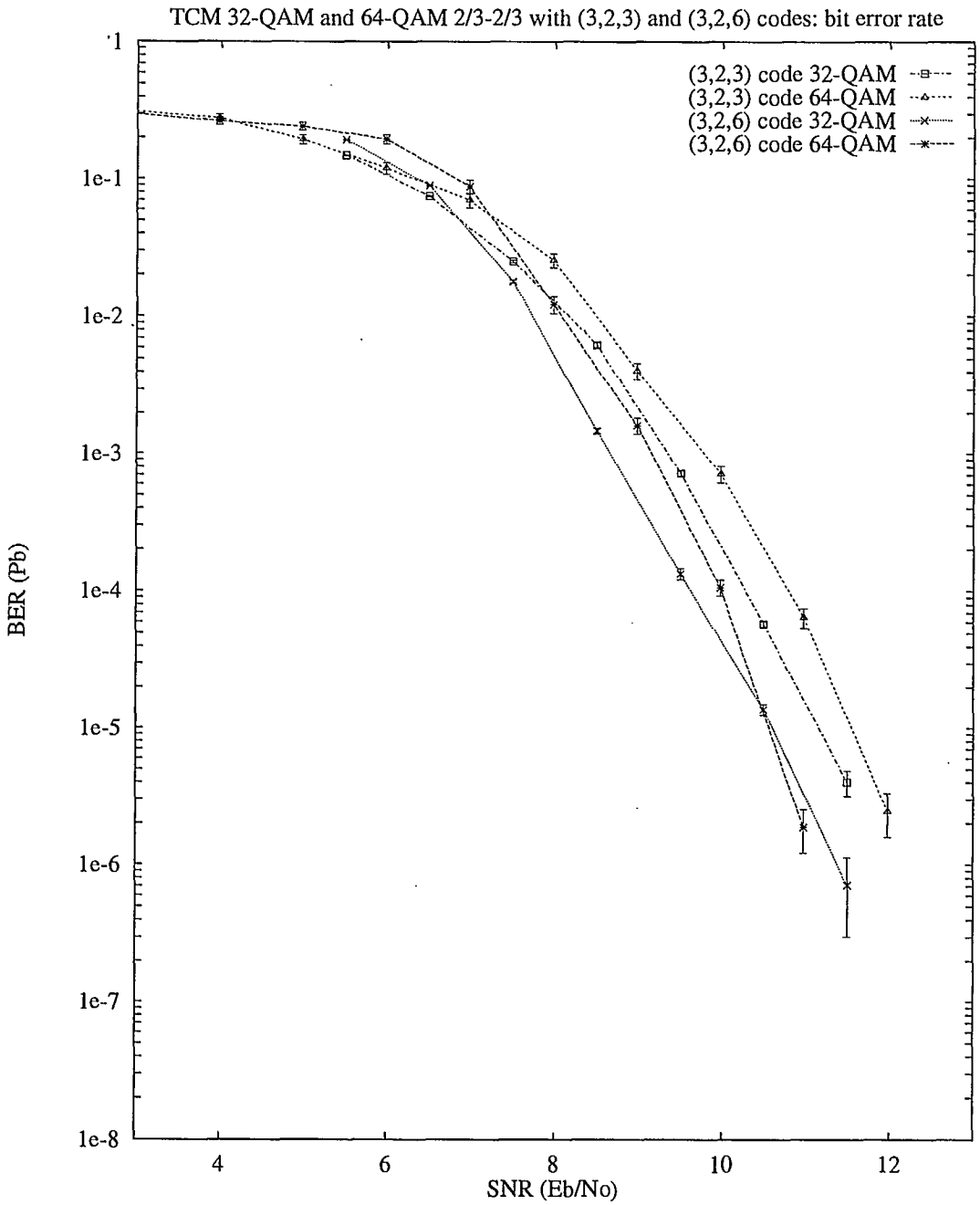

Figure 2.7: Probabilité d'erreur par bit pour les deux codes TCM (3,2,3) et (3,2,6) et les deux constellations TCM 32QAM et TCM 64QAM 2/3-2/3.

2.3 Codes correcteurs concaténés Reed-Solomon et modulation TCM-QAM

2.3.1 Introduction

À la sortie du décodeur de Viterbi, la distribution des erreurs en groupements, ou *salves*, d'erreurs, nécessite en général l'emploi de techniques de correction de groupements d'erreurs. Les codes Reed-Solomon (RS) constituent une classe importante des codes blocs en raison de leur capacité de correction de tels groupements d'erreurs.

Pour ces codes Reed-Solomon les symboles RS sont obtenus en regroupant m_{RS} bits d'information et assignés à un Corps Galois $GF[q] = GF[2^{m_{RS}}]$. Une série de k_{RS} symboles sont concaténés avec $2t$ symboles de redondance pour former un mot-code de n_{RS} symboles. Si le nombre de symboles du mot-code n_{RS} est inférieur à $N_{RS} = q - 1 = 2^{m_{RS}} - 1$, alors la séquence de symboles est complétée avec le nombre requis de zéros. Pour le système de codage concaténé qui nous intéresse ici, les mot-codes sont décomposés pour former les symboles TCM qui eux, sont transmis dans le canal bruité radioélectrique. Les symboles à la sortie du décodeur de Viterbi sont recombinés dans leur format original de mot-codes Reed-Solomon. Les groupements d'erreurs sont alors redistribués sur plusieurs symboles constituant un mot-code Reed-Solomon. Si t symboles, ou moins, sont en erreur parmi les n_{RS} symboles d'un mot-code RS, il est alors possible de les corriger et ainsi récupérer les k_{RS} symboles d'information originaux.

2.3.2 Description codes correcteurs Reed-Solomon utilisés

Pour nos simulations, nous avons essentiellement employés trois codes Reed-Solomon. Le code $RS(15, 9, 3)$ n'a été utilisé que pour tester et vérifier la construction des Corps Galois et les procédures de codage et décodage Reed-Solomon. Les Corps de Galois utilisés dans ce rapport pour déterminer les différents polynômes générateurs $g_{RS(n,k,t)}(x)$ des codes correcteurs Reed-Solomon $RS(n, k, t)$ sont décrits à l'annexe A (pages 52 et suivantes).

Code $RS(15, 9, 3)$:

- Corps de Galois $GF[16]$; $m_{RS} = 4$ (symboles hexadécimaux)
- Longueur des mots-codes $N_{RS} = 15$
- Nombre de symboles d'information $K_{RS} = 9$
- Capacité de correction $t = 3$ symboles hexadécimaux
- Longueur effective des mots: $n_{RS} = N_{RS}$ et $k_{RS} = K_{RS}$
- Taux du code $R_c = 60\%$ (redondance: $\frac{n-k}{n} = 40\%$)

Polynôme générateur $g_{RS(15,9,3)}(x)$ du code $RS(15, 9, 3)$:

$$\begin{aligned} g_{RS(15,9,3)}(x) &= (x-\alpha)(x-\alpha^2)(x-\alpha^3)(x-\alpha^4)(x-\alpha^5)(x-\alpha^6) \\ g_{RS(15,9,3)}(x) &= x^6 + \alpha^{10}\, x^5 + \alpha^{14}\, x^4 + \alpha^4\, x^3 + \alpha^6\, x^2 + \alpha^9\, x + \alpha^6 \end{aligned}$$

Les deux codes Reed-Solomon qui suivent ont été choisis pour effectuer la comparaison des performances de codes ayant des taux de code différents. Il s'agit des codes Reed-Solomon $RS(208, 188, 10)$ et $RS(255, 239, 8)$.

Code $RS(208, 188, 10)$:

- Corps de Galois $GF[q] = GF[2^{m_{RS}}] = GF[2^8]$; $m_{RS} = 8$ (octets)
- Longueur des mots-codes $N_{RS} = 255$
- Nombre de symboles d'information $K_{RS} = 235$
- Capacité de correction $t = 10$ octets
- Longueur effective des mots: $n_{RS} = 208$ et $k_{RS} = 188$
- Taux du code $R_c = 90.385\% \approx 90\%$ (redondance: $\frac{n-k}{n} = 9.615\% \approx 10\%$)

Polynôme générateur $g_{RS(208,188,10)}(x)$ du code $RS(208, 188, 10)$:

$$\begin{aligned} g_{RS(208,188,10)}(x) &= \prod_{i=1}^{2t}(x-\alpha^i) = (x-\alpha)(x-\alpha^2)(x-\alpha^3)\ldots(x-\alpha^{19})(x-\alpha^{20}) \\ g_{RS(208,188,10)}(x) &= x^{20} + \alpha^{18}\, x^{19} + \alpha^{62}\, x^{18} + \alpha^{82}\, x^{17} + \alpha^{54}\, x^{16} + \alpha^{66}\, x^{15} + \\ &\quad \alpha^{169}\, x^{14} + \alpha^{33}\, x^{13} + \alpha^{195}\, x^{12} + \alpha^{211}\, x^{11} + \alpha^{190}\, x^{10} + \alpha^{232}\, x^9 + \\ &\quad \alpha^{237}\, x^8 + \alpha^{96}\, x^7 + \alpha^{253}\, x^6 + \alpha^{171}\, x^5 + \alpha^{180}\, x^4 + \alpha^{229}\, x^3 + \\ &\quad \alpha^{230}\, x^2 + \alpha^{207}\, x + \alpha^{210} \end{aligned}$$

Code $RS(255, 239, 8)$:

- Corps de Galois $GF[q] = GF[2^{m_{RS}}] = GF[2^8]$; $m_{RS} = 8$ (octets)
- Longueur des mots-codes $N_{RS} = 255$
- Nombre de symboles d'information $K_{RS} = 239$
- Capacité de correction $t = 8$ octets
- Longueur effective des mots: $n_{RS} = 255$ and $k_{RS} = 239$
- Taux du code $R_c = 93.725\% \approx 94\%$ (redondance: $\frac{n-k}{n} = 6.275\% \approx 6\%$)

Polynôme générateur $g_{RS(255,239,8)}(x)$ du code $RS(255, 239, 8)$:

$$g_{RS(255,239,8)}(x) = \prod_{i=1}^{2t}(x - \alpha^i) = (x - \alpha)(x - \alpha^2)(x - \alpha^3)\dots(x - \alpha^{15})(x - \alpha^{16})$$

$$\begin{aligned} g_{RS(255,239,8)}(x) = {} & x^{16} + \alpha^{121}\, x^{15} + \alpha^{106}\, x^{14} + \alpha^{110}\, x^{13} + \alpha^{113}\, x^{12} + \alpha^{107}\, x^{11} + \\ & \alpha^{167}\, x^{10} + \alpha^{83}\, x^9 + \alpha^{11}\, x^8 + \alpha^{100}\, x^7 + \alpha^{201}\, x^6 + \alpha^{158}\, x^5 + \\ & \alpha^{181}\, x^4 + \alpha^{195}\, x^3 + \alpha^{208}\, x^2 + \alpha^{240}\, x + \alpha^{136} \end{aligned}$$

2.3.3 Simulations avec la modulation codée 32QAM et les codes Reed-Solomon

Les Figures 2.8 et 2.9 montrent les résultats de simulations obtenus pour le code Reed-Solomon $RS(208, 188, 10)$ avec la modulation codée 32QAM en termes de taux d'erreur par symbole Reed-Solomon et de taux d'erreur par bit respectivement. Il est à noter que les symboles Reed-Solomon ne sont en fait que des octets, c'est-à-dire $m_{RS(208,188,10)} = 8$, car ils sont des éléments du Corps Galois $GF[256]$, et correspondent donc à deux symboles TCM à l'entrée et à la sortie du codeur convolutionnel. On ne peux donc pas comparer directement le taux d'erreur par symbole Reed-Solomon et par symbole TCM.

La Figure 2.10 illustre bien le gain additionnel de codage, en taux d'erreur par bit, en comparant les courbes pour le code TCM 32QAM (3,2,3) seul, et concaténé avec le code externe $RS(208, 188, 10)$.Ainsi pour un taux d'erreur par bit de 10^{-5}, on obtient des gains de $3dB$ et de $2dB$ avec le code TCM (3,2,3) et le code TCM (3,2,6).

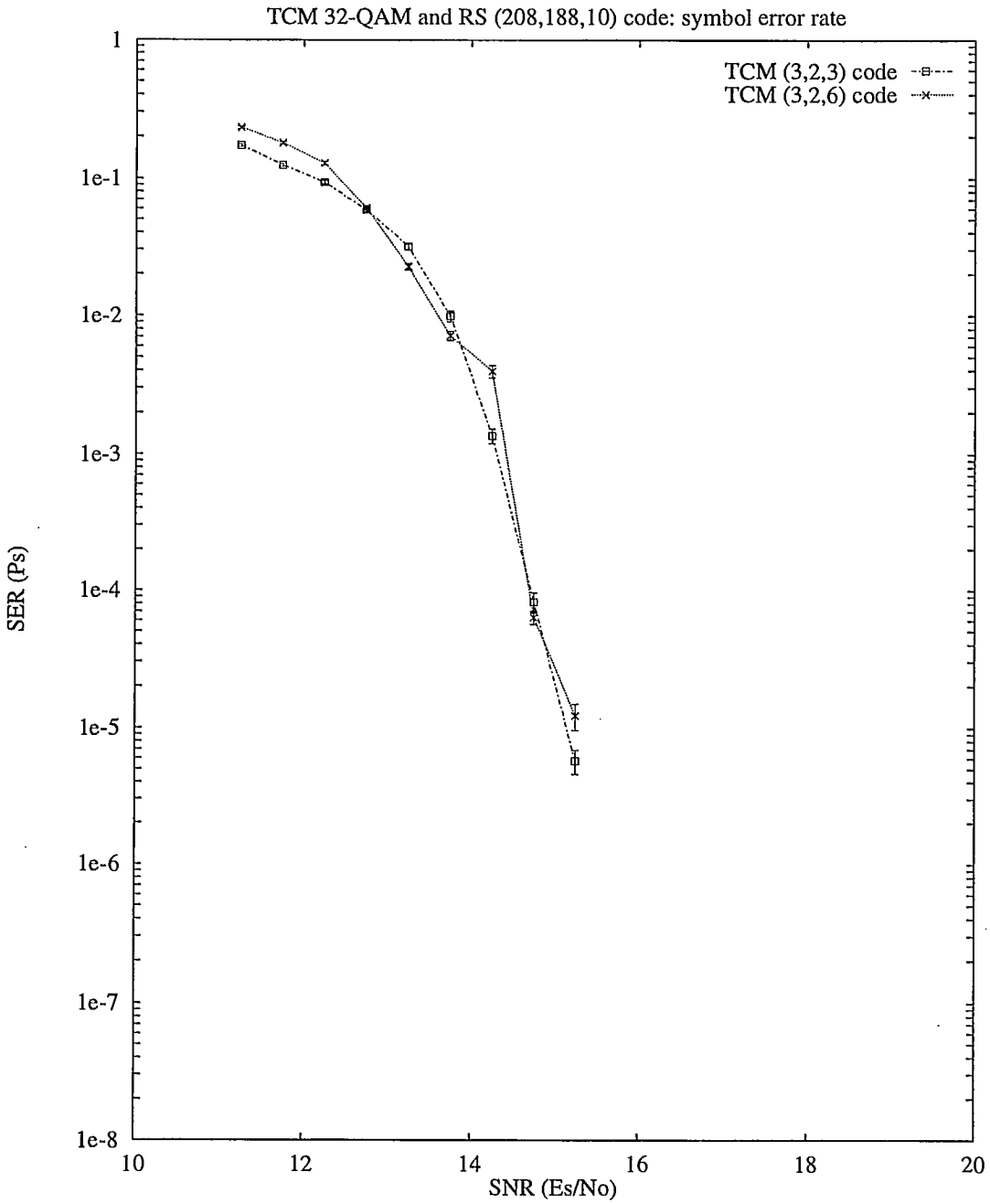

Figure 2.8: Probabilité d'erreur par symbole RS avec les codes TCM (3,2,3) et (3,2,6) utilisant la modulation 32QAM.

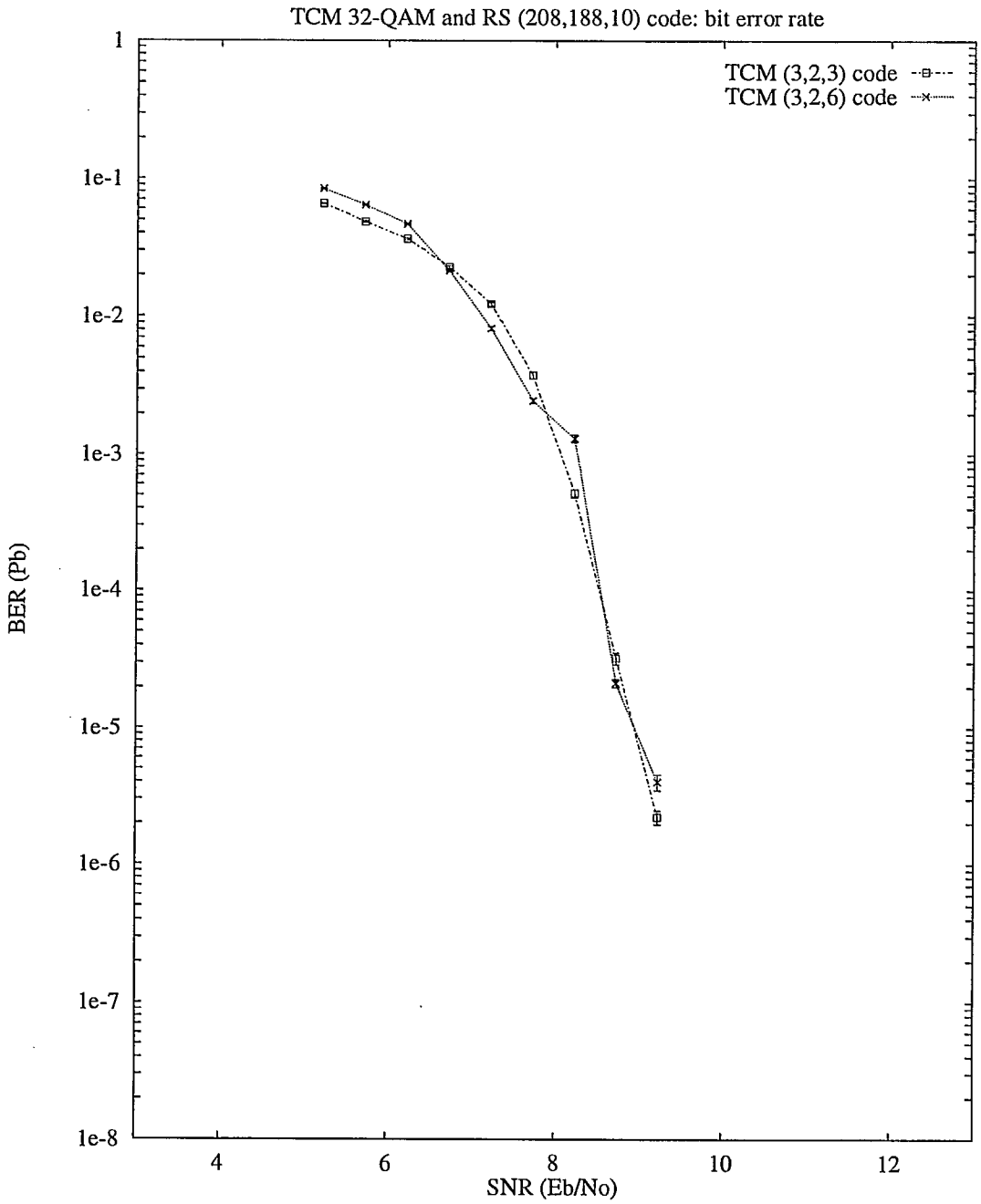

Figure 2.9: Probabilité d'erreur par bit avec les codes TCM (3,2,3) et (3,2,6) utilisant la modulation 32QAM.

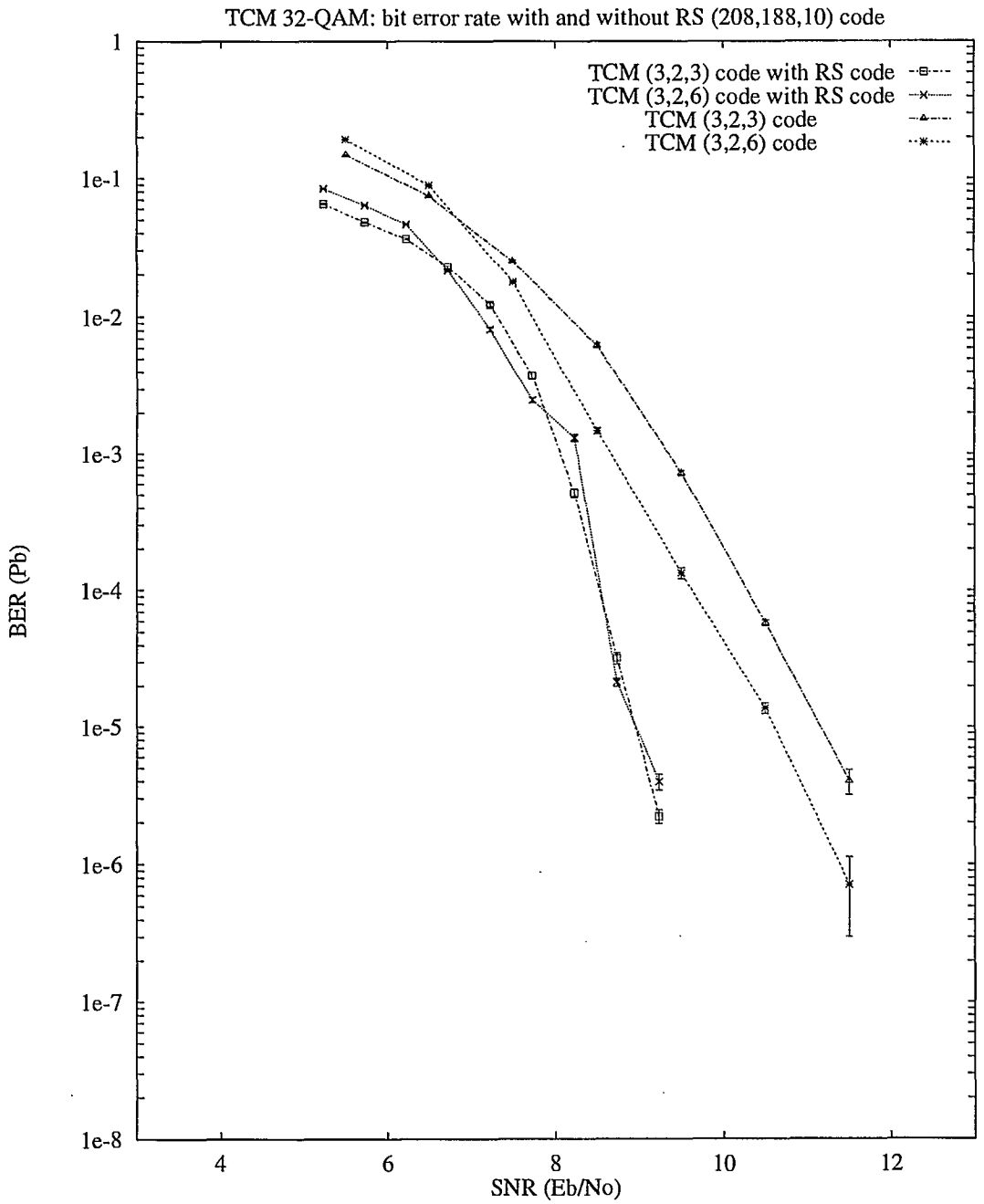

Figure 2.10: Comparaison des probabilités d'erreur par bit avec les codes TCM (3,2,3) et (3,2,6) sans et avec code externe Reed-Solomon.

2.4 Entrelacement

Considérons en premier lieu celui des deux codes Reed-Solomon ayant la plus faible capacité de correction t, i.e., le code $RS(255, 239, 8)$. La Figure 2.11 et la Figure 2.12 donnent respectivement le taux d'erreur par symbole (octet RS) et le taux d'erreur par bit, lorsque le système de télécommunications utilise la concaténation des codes TCM (3,2,3) et (3,2,6) avec une constellation classique 32-CROSS, et avec une constellation bi-unidimensionnelle 64QAM de taux 2/3-2/3, et avec le code $RS(255, 239, 8)$ ainsi que de l'entrelacement de symboles.

Les Figures 2.13 et 2.14 présentent les courbes de probabilité par symbole et par bit de ces codes concaténés avec les deux mêmes constellations 32-TCM QAM et 2/3-2/3 64-TCM QAM mais, cette fois-ci, avec le code Reed-Solomon plus robuste $RS(208, 188, 10)$. On peut noter que le choix de l'une ou l'autre des deux constellations donnent des résultats très similaires On peut observer également sur ces courbes que le code $RS(208, 188, 10)$ est de loin plus performant pour les valeurs de rapport signal-à-bruit moyennes et faibles. On peut enfin mentionner qu'un gain de codage d'environ 0.5 dB est obtenu lorsque l'on passe du code $RS(255, 239, 8)$ au code plus puissant $RS(208, 188, 10)$. Cette amélioration de performance correspond à une augmentation de la redondance du code Reed-Solomon de 4%; i.e., de 6% à 10%.

Enfin considérons l'effet de l'entrelacement sur la performance des systèmes concaténés étudiés: les Figures 2.15 et 2.16 permettent de faire cette comparaison. La profondeur d'entrelacement est fixée à 8. Avec cette profondeur d'entrelacement relativement modeste, un gain de codage additionnel 1 dB est obtenu, confirmant bien l'utilité d'entrelacer les symboles.

2.5 Conclusion

L'objectif d'un taux d'erreur par bit de l'ordre de 10^{-9} n'a pu être obtenu par les simulations de type Monte-Carlo par ordinateur. Cependant, à un taux d'erreur de l'ordre de 10^{-5}, on remarque que les courbes de probabilités d'erreur sont presque verticales. À toutes fins pratiques, il semble que l'on obtienne cet objectif pour 1 dB additionnel de rapport signal à bruit. Il est important de noter que cette performance est obtenue aussi bien avec l'une ou l'autre des deux techniques de modulation codée, à savoir le TCM 32QAM conventionnel et le TCM 64QAM de taux 2/3-2/3, lorsqu'ils sont concaténés avec des codes Reed-Solomon exploitant l'entrelacement de symboles.

Il en ressort donc qu'un système de transmission numérique utilisant le codage TCM en conjonction avec des codes Reed-Solomon et de l'entrelacement constituent une stratégie

potentielle pour la télédiffusion de la télévision avancée en Amérique du Nord.

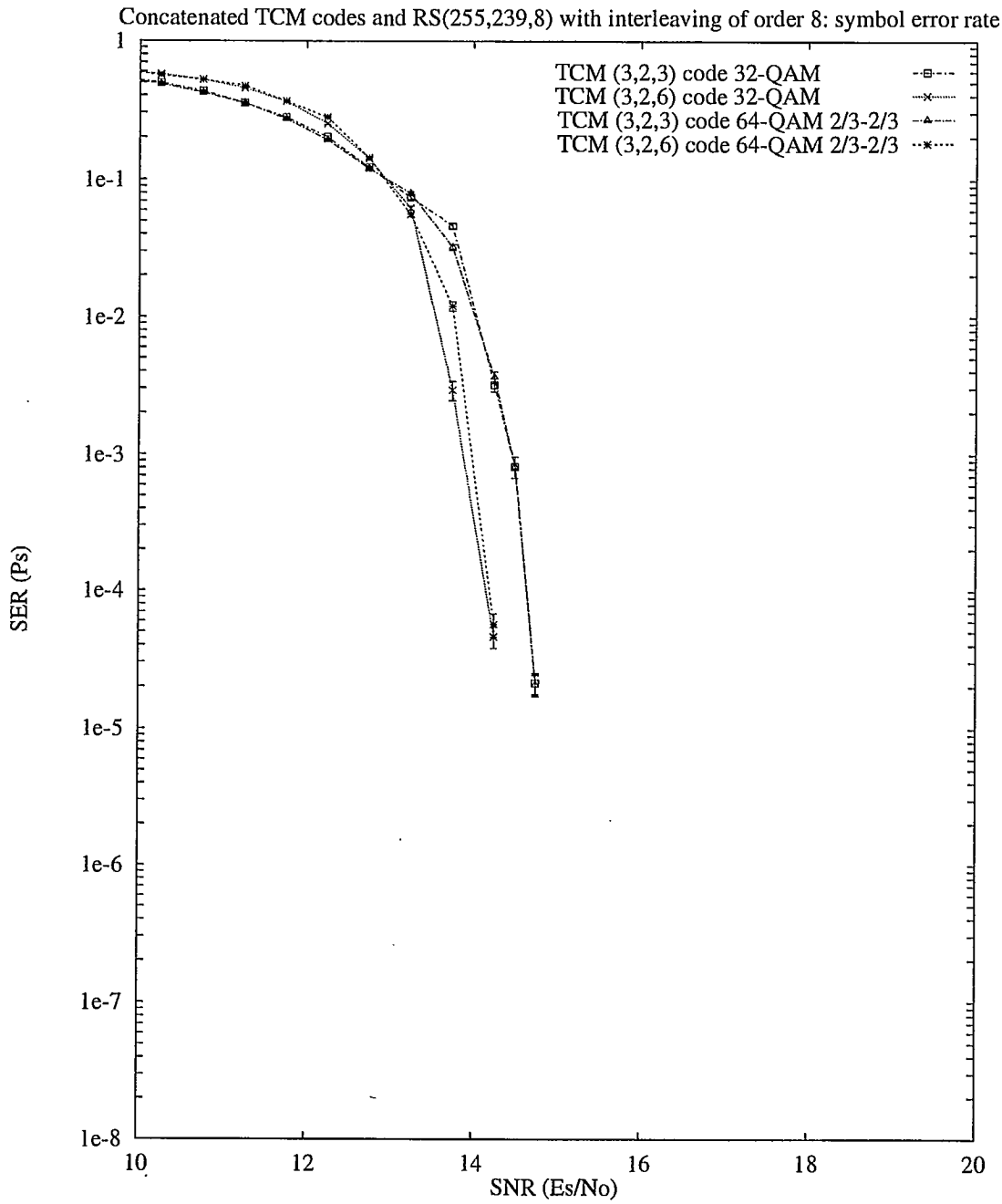

Figure 2.11: Probabilité d'erreur par symbole RS avec les codes TCM (3,2,3) et (3,2,6) avec la constellation 32QAM et un code RS(255,239,8).

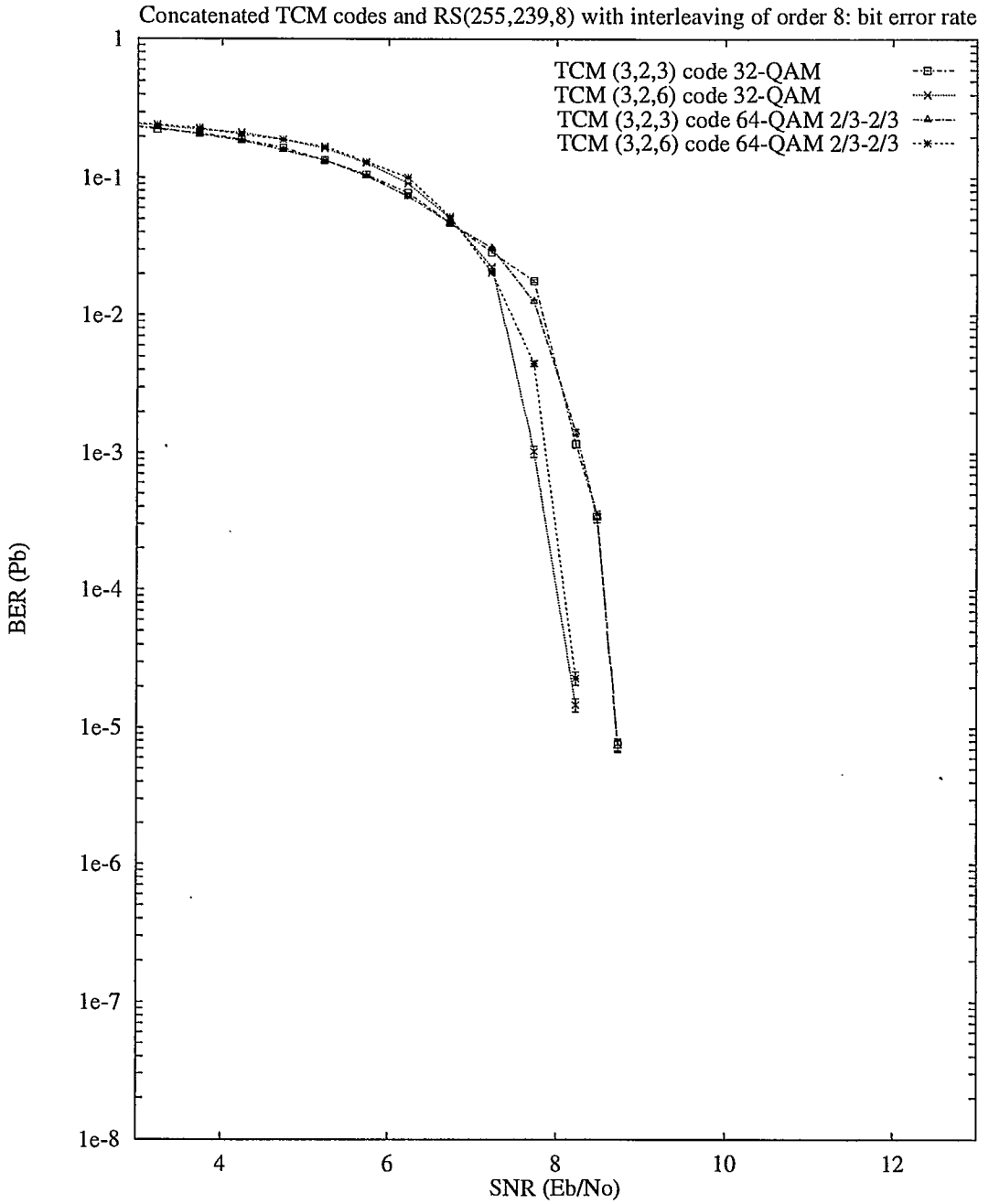

Figure 2.12: Probabilité d'erreur par bit avec les codes TCM (3,2,3) et (3,2,6) avec la constellation 32QAM et un code RS(255,239,8).

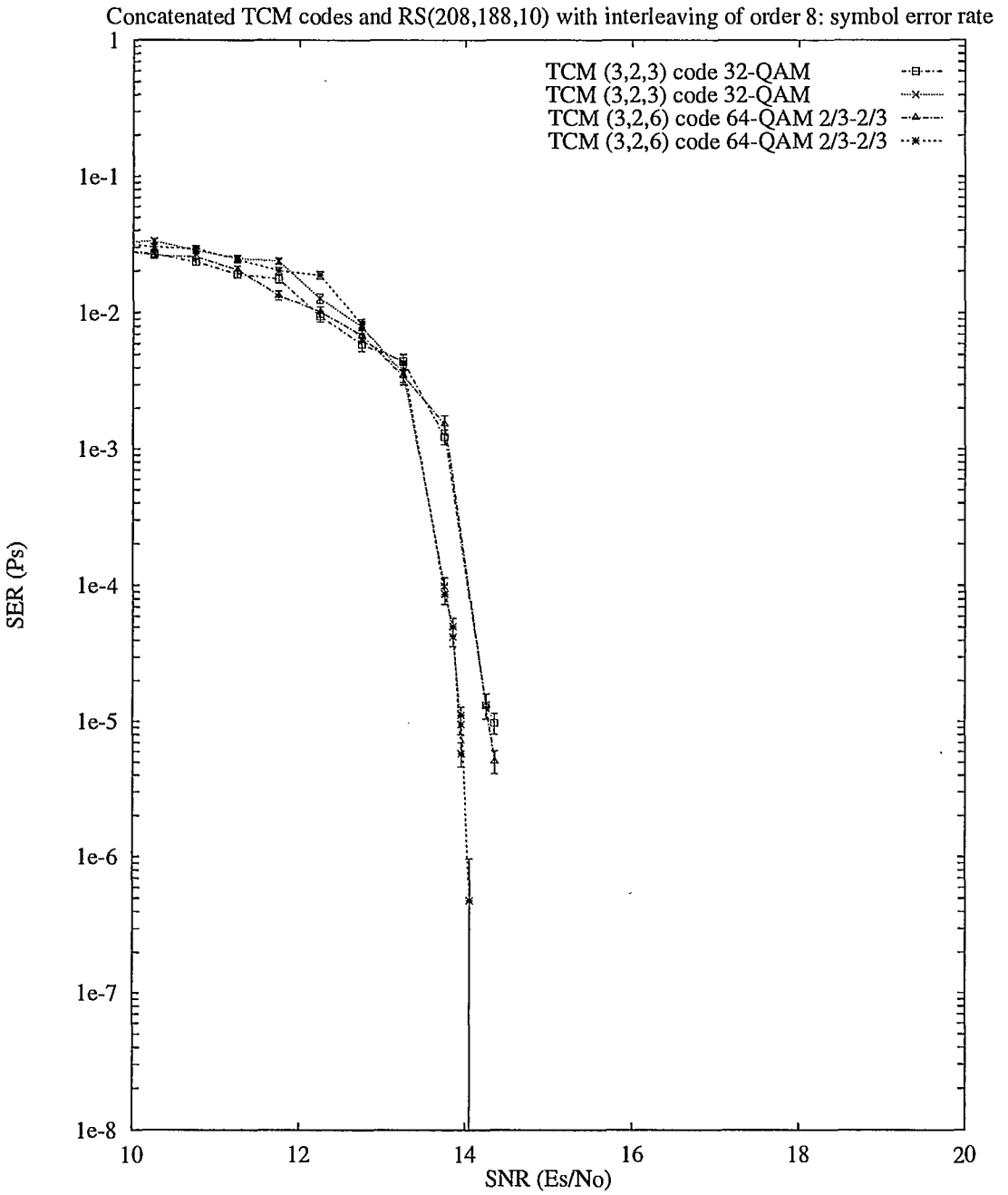

Figure 2.13: Probabilité d'erreur par symbole RS avec les codes TCM (3,2,3) et (3,2,6) avec la constellation 32QAM et un code RS(208,188,10).

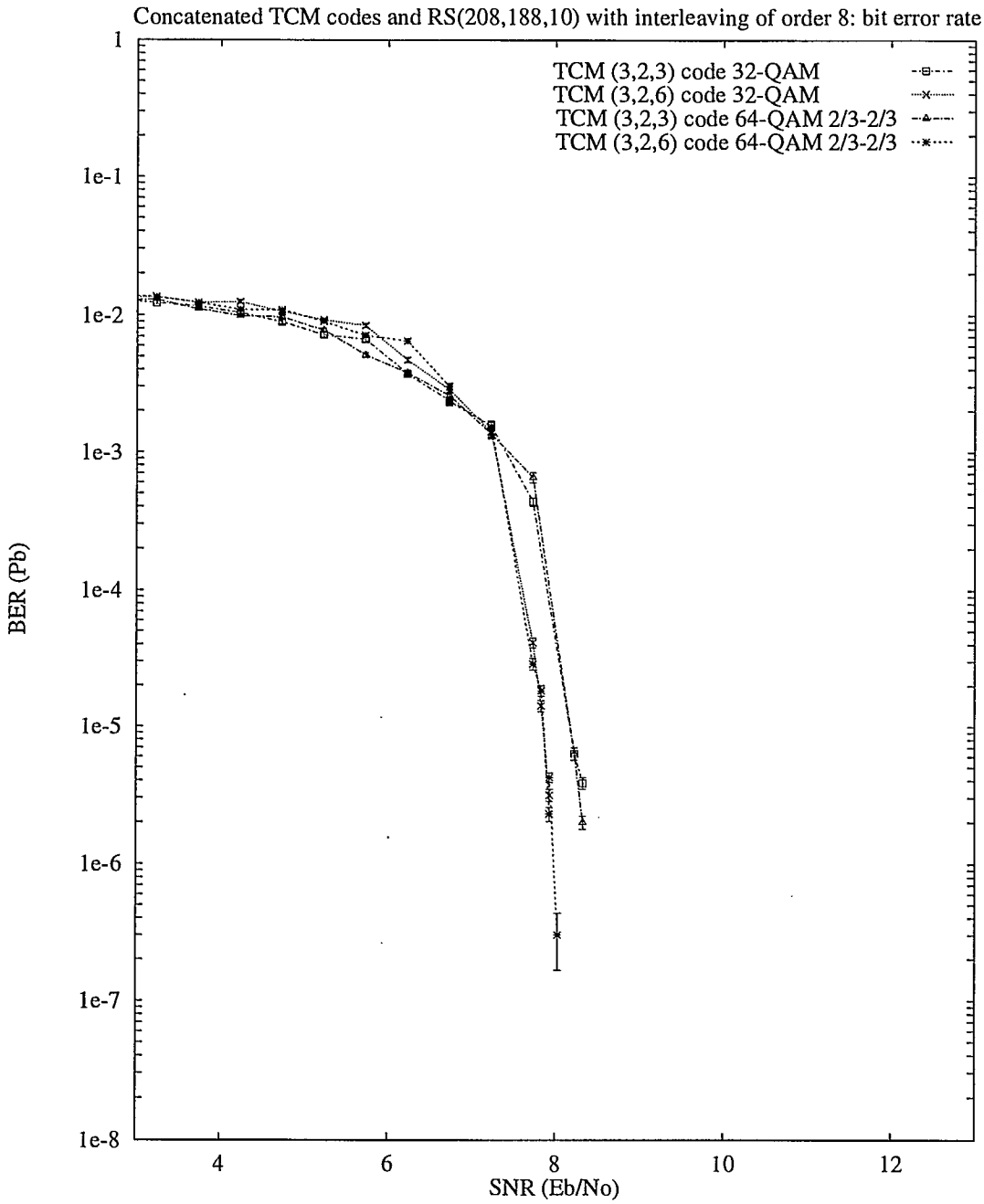

Figure 2.14: Probabilité d'erreur par bit avec les codes TCM (3,2,3) et (3,2,6) avec la constellation 32QAM et un code RS(208,188,10).

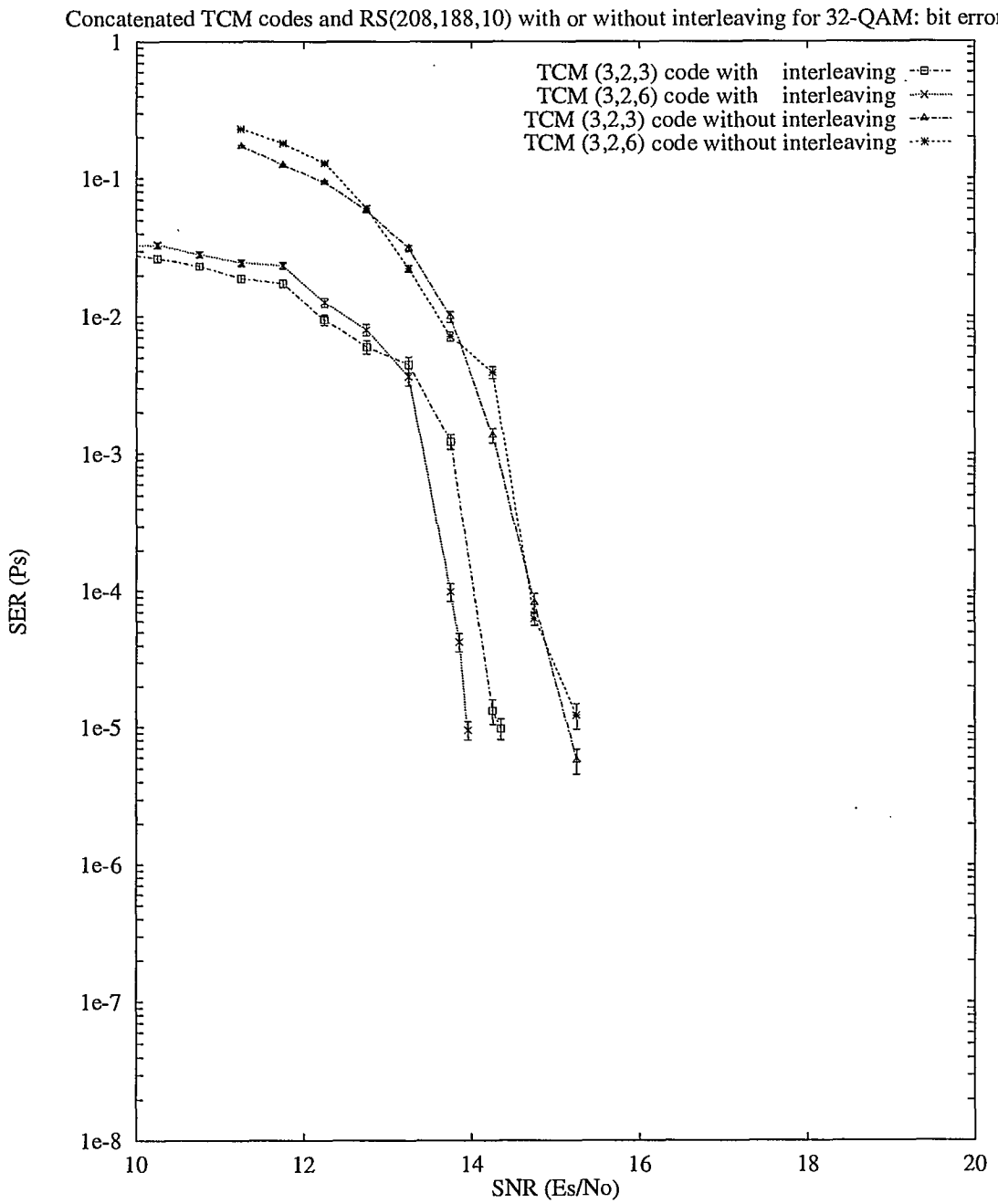

Figure 2.15: Probabilité d'erreur par symbole RS avec le code TCM (3,2,3) et constellation 32QAM, le code RS(208,188,10) et un entrelacement de symboles de profondeur 8.

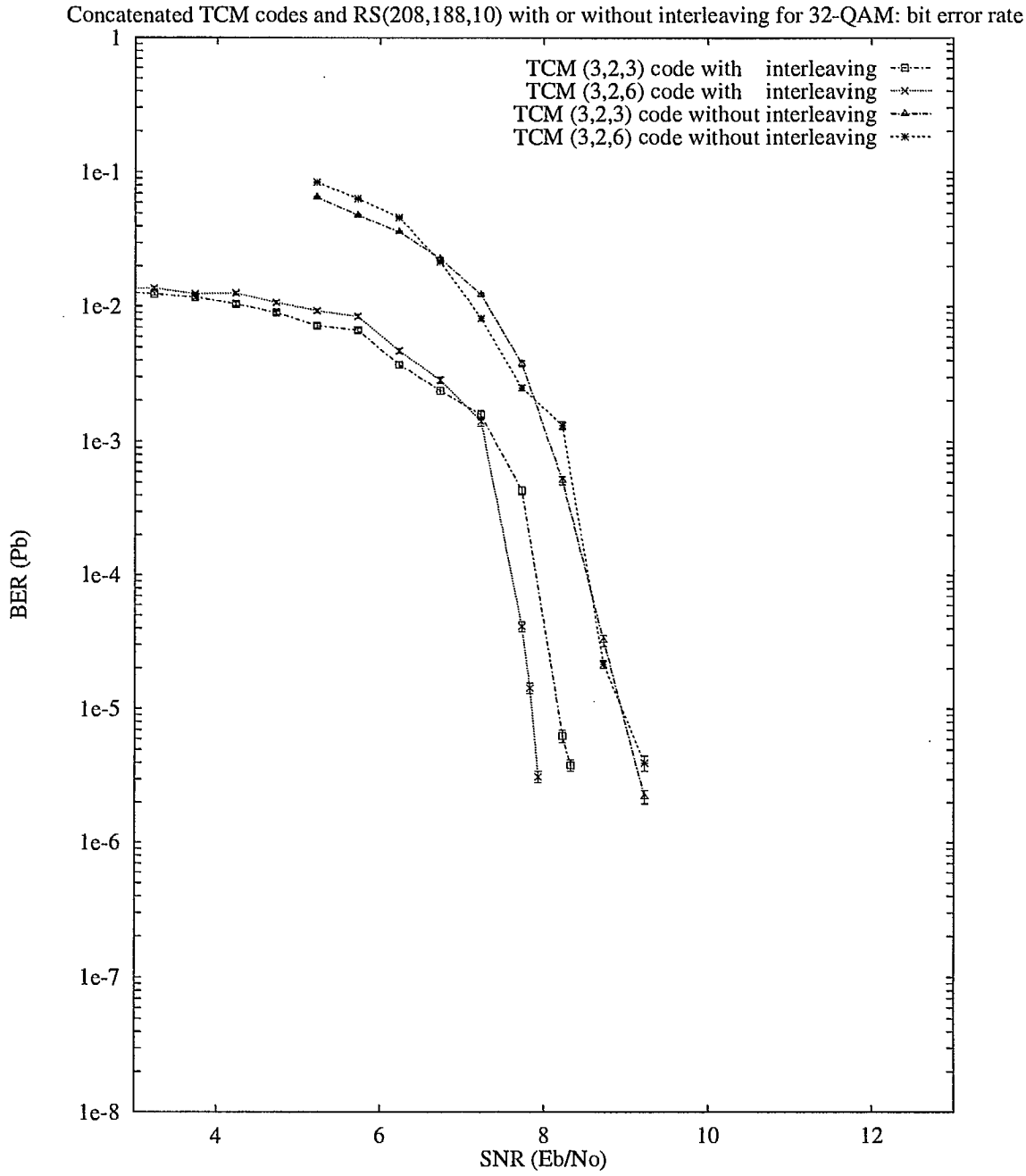

Figure 2.16: Probabilité d'erreur par bit avec le code TCM (3,2,3) et constellation 32QAM, le code RS(208,188,10) et un entrelacement de symboles de profondeur 8.

Chapter 3

Programme de simulation utilisant MATLAB et SIMULINK

3.1 Introduction

Dans ce chapitre, on décrit le fonctionnement d'un programme de simulation permettant de représenter les différents signaux dans une chaîne de télécommunication. Pour effectuer les simulations, les logiciels *MATLAB* et *SIMULINK* (de la compagnie *The MathWorks, Inc.*) ont été choisis pour leur flexibilité ainsi que pour leur disponibilité. Les principaux avantages découlant de l'utilisation de ces logiciels sont leur interface graphique et la simplicité de réalisation des modules de simulation (ou *blocs* de simulation). En contre-partie, les principaux inconvénients sont la quantité de mémoire requise ainsi que le nombre élevé de calculs que doit faire l'ordinateur. Pour simuler le système de modulation TCM 64QAM, un minimum de 8 Megaoctets est requis pour obtenir des résultats, alors que le temps nécessaire pour effectuer ces simulations est énorme (e.g. plusieurs semaines pour des taux pouvant aller à 10^{-6}) sur un ordinateur personnel employant un micro-processeur de type 486 fonctionnant à 33 MHz. Les remarques suivantes s'appliquent aux simulations proprement dites:

- Les simulations sont exécutées en bande de base.
- La méthode *Gear*[1] doit être utilisée lorsque le système est non linéaire afin d'éviter les erreurs d'échantillonnage [Mat92].
- Les simulations requièrent la collection de macros *DSP Blockset* pour le logiciel *SIMULINK* ainsi que l'ensemble de macros *Signal Processing Toolbox* pour *MATLAB*.

[1] Il s'agit de la méthode de prédiction-correction de Gear pour les systèmes avec contrainte.

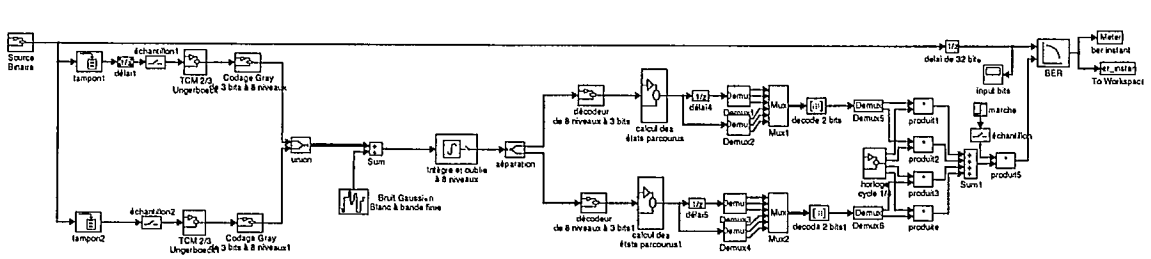

Figure 3.1: Schéma bloc du système de télécommunication simulé.

- Les variables *taux* et N_0 doivent être initialisées dans la "fenêtre de commandes" *MATLAB* avant de débuter les simulations.
- Les lignes grasses sur les diagrammes indiquent la transmission de plusieurs bits, ou symboles, en parallèle.
- On peut voir le schéma bloc de certains modules en faisant un double "click" sur l'icône les représentant.

3.2 Système de télécommunication TCM 64QAM simulé

L'architecture du système de télécommunication TCM 64QAM choisi pour les simulations est représenté à la Figure 3.1.

Les paragraphes suivants décrivent le fonctionnement et l'utilité des modules du programme de simulation qui apparaissent sur le schéma bloc du système de télécommunication de la Figure 3.1. Il est à noter qu'un X à la fin du nom d'un module indique que plus d'un module possède le même nom de base. Par exemple, la notation générique *tamponX* identifie les modules spécifiques *tampon1* et *tampon2*. Une suite consécutive de modules avec le même nom de base est indiquée par le premier et le dernier nombre séparé par un trait "-" (ex: les modules *tampon1*, *tampon2* et *tampon3* sont indiqués par les modules *tampon1-3*). Les modules développés sont décrits dans les pages qui suivent.

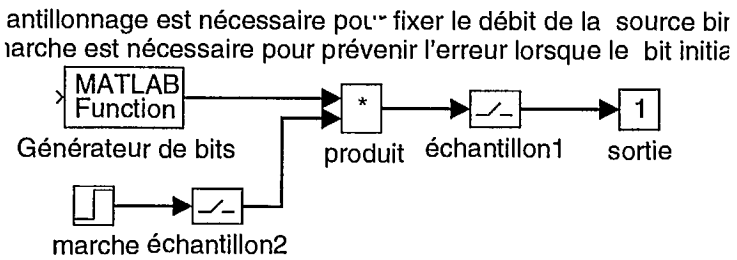

Figure 3.2: Module *source binaire.*

Figure 3.3: Module *tamponX.*

3.3 Transmetteur

Source d'information binaire

Le module *source binaire* utilisé pour les simulations a été construit à partir des fonctions *round* et *rand* de *MATLAB.* Le module *Générateur de bits* fait appel à la commande *round(rand(1,1))* qui génère des nombres aléatoires avec distribution uniforme entre "0" et "1", puis les arrondit à l'entier le plus près ("0" ou "1"). Le module *échantillon1* fixe le débit de la source à *taux* bits par seconde. La variable *taux* doit être définie dans la "fenêtre de commandes *MATLAB*" avant de commencer la simulation. Les deux modules *marche* et *produit* assurent que le premier bit produit est un "0" (lorsque le premier bit est un "1", le système produit, pour une raison inconnue, des erreurs d'échantillonnage qui faussent les résultats). Le module *échantillon2* indique à *MATLAB* que le pas de simulation doit être fixé au débit *taux*, ce qui minimise le temps requis pour le calcul de chaque bit.

Conversion série-parallèle

Les modules *tamponX* font la conversion de l'information provenant de la source binaire du mode série au mode parallèle. Ces modules sont disponibles dans la collection de macros *DSP Blockset.* Leur entrée provenant de la source binaire a un taux de *taux* bits par seconde alors que leur sortie a un taux égal à $\frac{taux}{2}$ bits par seconde, deux bits étant transmis en

Figure 3.4: Module *délaiX*.

Figure 3.5: Module *échantillonX*.

parallèle.

Fonction de retard

Les modules *délaiX* sont disponibles avec *SIMULINK*. Ils ont pour fonction d'échantillonner et de retarder leurs entrées. Le retard est déterminé par le paramètre *Sample time*, qui doit correspondre au taux de données sur la ligne. Le taux d'échantillonnage du module *délai1* est donc fixé à *taux* bits par seconde.

Échantillonnage

Les modules *échantillonX* échantillonnent leurs entrées à un taux prédéfini. Ils sont disponibles avec *SIMULINK*, mais le diagramme de leur module a été modifié afin d'être plus "symbolique". Ainsi modifés, ils assurent que seules les valeurs valides sont transmises au module suivant: à savoir le module *TCM 2/3 Ungerboeck*. Ces modules servent à changer le taux d'échantillonnage d'une ligne. Ici, les modules *échantillon1-2* ont à leur entrée un taux égal à $\frac{taux}{2}$ et à leur sortie un taux égal à $\frac{taux}{4}$.

Codage convolutionnel

Comme leur nom l'indique, les modules *TCM 2/3 UngerboeckX* ont été conçus pour produire le codage d'Ungerboeck (codage convolutionnel au taux $\frac{2}{3}$). Les trois bits en parallèle à la sortie dépendent des deux bits présents à l'entrée ainsi que des trois bits précédents.

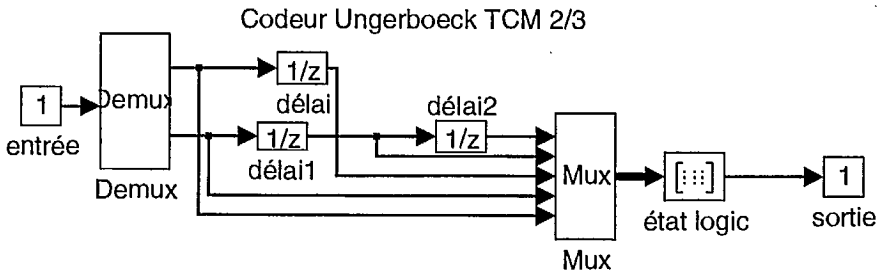

Figure 3.6: Module *TCM 2/3 UngerboeckX*.

Les modules "délaisX" servent ici de mémoires pour le codeur convolutionnel. Le module *"état logic"* fait le *mapping* requis entre les bits d'entrée et le contenu des mémoires avec la sortie et ce, à l'aide d'un tableau. Pour une séquence quelconque de cinq bits (i.e. bits d'entrée et contenu des mémoires) produite par la source binaire, soit: x_1, x_2, x_3, x_4, x_5, il y a trente-deux combinaisons différentes possibles pour générer les trois bits à la sortie (y_1, y_2 et y_3). Chaque colonne représente une combinaison des cinq bits avec le résultat correspondant:

x_1	00000000000000001111111111111111
x_2	00000000111111110000000011111111
x_3	00001111000011110000111100001111
x_4	00110011001100110011001100110011
x_5	01010101010101010101010101010101
y_1	00111100001111000011110000111100
y_2	01010101010101011010101010101010
y_3	00000000111111110000000011111111

C'est ce tableau de valeurs qu'utilise le module *état logic* pour déterminer les trois bits (y_1, y_2 et y_3) à sa sortie. Les opérations de ces modules se font toutes au même taux de $\frac{taux}{4}$.

Code de Gray de 3 bits et modulation ASK à 8 niveaux

Le module *Codage Gray de 3 bits à 8 niveauxX* code effectue le codage de Gray sur trois bits et module en ASK huit niveaux (modulation par déplacement d'amplitude à huit niveaux) les bits présents à son entrée. Les huit niveaux ASK sont $[\pm 7, \pm 5, \pm 3, \pm 1]$. Cette opération est effectuée à l'aide du module *état logic* qui utilise le tableau suivant (les colonnes représentent les combinaisons possibles de trois bits à l'entrée x_1, x_2 et x_3 avec les sorties correspondantes y):

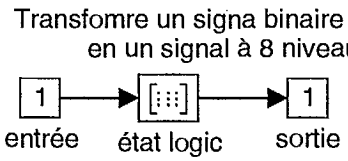

Figure 3.7: Codage de Gray sur 3 bits et modulation ASK à 8 niveaux.

Figure 3.8: Combinaison des signaux ASK en quadrature pour former le signal 64QAM.

x_1	0	0	0	0	1	1	1	1
x_2	0	0	1	1	0	0	1	1
x_3	0	1	0	1	0	1	0	1
y	-7	-5	-1	-3	7	5	1	3

Ici, le taux de symboles à la sortie du module est le même que celui à son entrée (i.e. $\frac{taux}{4}$).

Combinaison des signaux ASK en quadrature

Le module *union* combine en quadrature ses deux entrées (signaux 8ASK réels) pour former un symbole complexe 64QAM. Le premier signal ASK constitue la partie réelle du symbole complexe 64QAM alors que le deuxième signal ASK en constitue la partie imaginaire. Dans le cas qui nous intéresse, la sortie de ce module produit donc l'équivalent en bande de base d'un signal 64QAM. Ce module est disponible dans la collection de macros *DSP Blockset* de *SIMULINK*.

3.4 Canal de transmission

Bruit additif gaussien blanc

Du bruit additif gaussien blanc à largeur de bande finie est alors additionné au signal

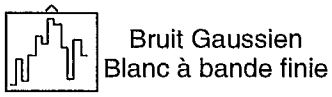

Figure 3.9: Bruit additif gaussien blanc à largeur de bande finie.

Figure 3.10: Module de sommation vectorielle.

64QAM à l'aide du module *Sum* et du module *Bruit Gaussien Blanc à Bande finie*. La sortie de ce dernier module est échantillonné au taux $\frac{taux}{0.4}$ tel que défini par sa liste de paramètres afin de générer dix échantillons de bruit pour chaque échantillon du signal 64QAM (les échantillons 64QAM arrivant à un taux de $\frac{taux}{4}$). La densité spectrale de puissance du bruit, N_0, indiquée dans la liste de paramètres doit, quant à elle, être définie dans la fenêtre de commandes *MATLAB*. Ce module est aussi disponible avec le logiciel *SIMULINK*.

Sommation vectorielle

Le module *Sum* de *SIMULINK* effectue la somme vectorielle de ses entrées (la somme ce fait symbole complexe par symbole complexe).

3.5 Récepteur

Filtre adapté

Le module *Intègre et oublie à 8 niveaux* a été conçu pour faire l'estimation des symboles à la réception corrompus par le canal gaussien à l'aide d'un filtre adapté. Le signal reçu à l'entrée de ce module est intégré sur une période de quatre bits (la période d'un symbole 64QAM) à l'aide du module *Intégrateur*. À la fin de cette période de quatre bits, le module *échantillon* échantillonne la sortie de l'intégrateur et la garde pour la prochaine période de quatre bits. Les modules *délai* et *Sum* permettent de déterminer la valeur de l'intégration du dernier symbole seul. Ces derniers sont nécessaires puisque le module *Intégrateur* intègre son entrée continuellement pendant toute la durée de la simulation. Pour obtenir la valeur désirée, il faut donc soustraire de la sortie de l'intégrateur sa valeur cumulative jusqu'à

l'instant du symbole précédent. C'est pour cette raison que l'on doit utiliser le module *délai* avec un retard de $\frac{4}{taux}$.

Le module *Gain* amplifie la valeur à son entrée par la valeur indiquée qui est de $\frac{taux}{4}$. Ainsi, sa sortie pourra être comparée avec les niveaux de seuils choisis $[\pm 6, \pm 4, \pm 2, 0]$, pour déterminer lequel des symboles $[\pm 7, \pm 5, \pm 3, \pm 1]$ doit être estimé. Par exemple, si l'intégration donne une valeur plus grande ou égale à 6, alors le symbole est estimé à +7; si le résultat de l'intégration donne une valeur plus grande ou égale à 4 et plus petite que 6, alors le symbole est estimé comme étant +5, et ainsi de suite pour les autres symboles. Ces opérations de comparaison sont effectuées à l'aide des modules *constanteX*, *comparaisonX*, *logicX*, *GainX* et *Sum1*. Les modules *GainX* sont nécessaires puisque la sortie des modules *comparaison* (qui est "0" ou "1") doit être transformée avec la valeur appropriée $[\pm 7, \pm 5, \pm 3, \pm 1]$. À chaque instant, une seule des entrées du module *Sum1* est non nulle, ce qui permet de transmettre la valeur adéquate vers la sortie. Les modules *marche*, *échantillon1* et *produit* sont nécessaires pour maintenir la sortie à "zéro" jusqu'à ce qu'une valeur soit disponible, c'est-à-dire après $\frac{8}{taux}$ secondes, ou une période de deux symboles 64QAM. L'effet de ce module est donc celui d'un filtre adapté pour chaque niveau de symbole ASK: $[\pm 7, \pm 5, \pm 3, \pm 1]$, avec des niveaux de seuils: $[\pm 6, \pm 4, \pm 2, 0]$, respectivement. Le taux d'échantillonnage à l'entrée de ce module est le même que le taux d'échantillonnage du bruit ($\frac{taux}{0.4}$), alors que le taux d'échantillonnage à la sortie est le quart de celui de la source ($\frac{taux}{4}$). Les deux composantes ASK réelle et imaginaire de chaque symbole 64QAM sont traitées en même temps et individuellement par ce module, réduisant ainsi la complexité du décodeur.

Séparation des composantes en phase et en quadrature

Le module *séparation* du *DSP Blockset* sépare son entrée complexe (c'est-à-dire le symbole 64QAM) en deux composantes: celle du haut représentant le signal ASK en phase (partie réelle) alors que la branche du bas représente le signal ASK en quadrature (partie imaginaire).

Décodeur de Gray à 3 bits

Le module *décodeur de 8 niveaux à 3 bits* a été réalisé pour décoder le symbole ASK à huit niveaux présenté à son entrée avec 3 bits. Il s'agit tout simplement de la fonction inverse de celle effectuée par le module *Codage Gray de 3 bits à 8 niveaux* au transmetteur. L'entrée est en même temps comparée avec les huit états possibles $[\pm 7, \pm 5, \pm 3, \pm 1]$ à l'aide des modules *constante0-7* et *comparaisonX*. La branche pour laquelle la comparaison est vraie aura alors une valeur de "1" alors que les sept autres branches donneront une valeur nulle. Chacune des branches est ensuite multipliée par les trois bits correspondants à l'aide

des modules *constante8-15* et *produitX*. Une seule des entrées du module *Sum* sera ainsi choisie (ou multipliée par "1") et transmise à la sortie puisque les sept autres termes seront tous nuls (i.e. "[0 0 0]" ou multipliés par "0").

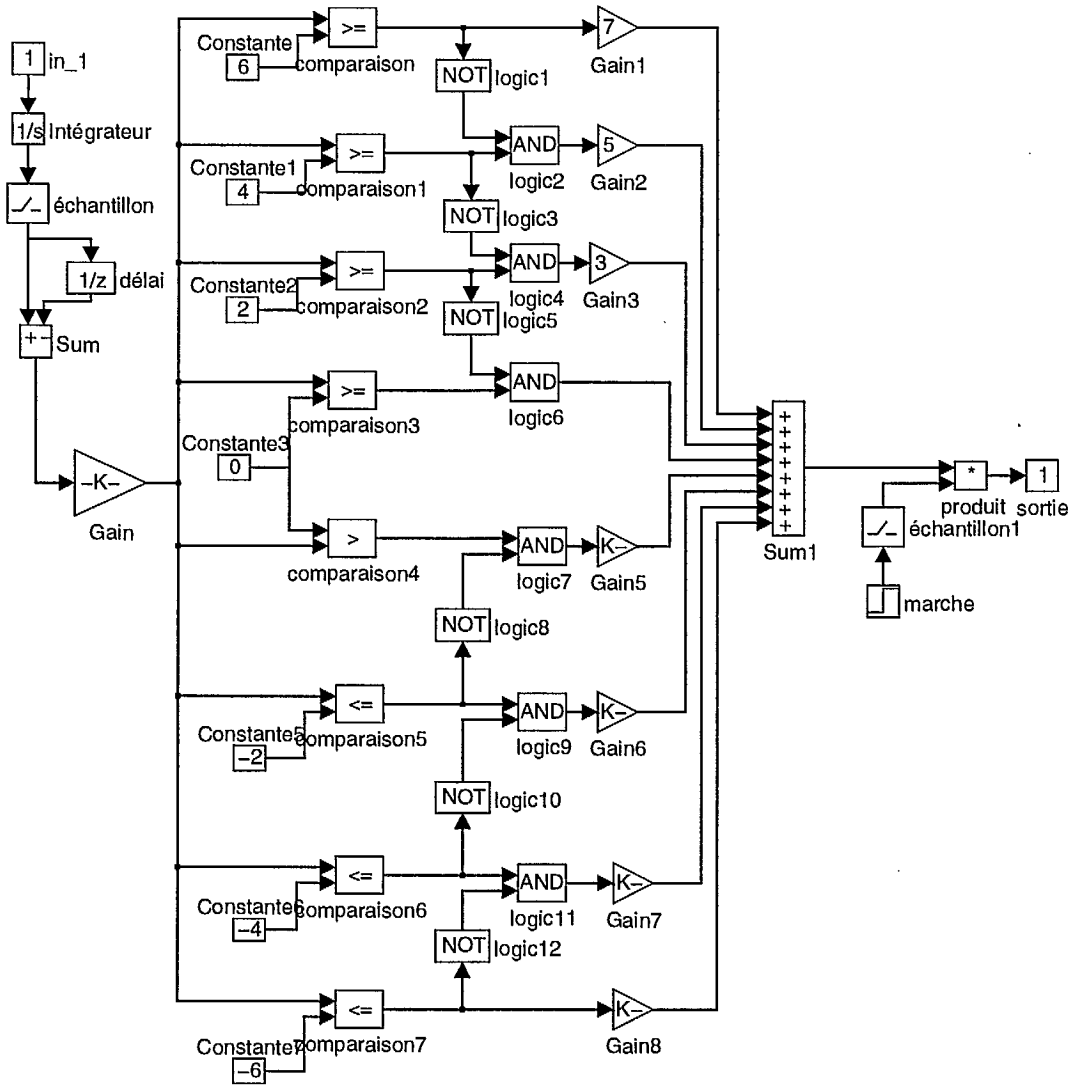

Figure 3.11: Filtre adapté.

Figure 3.12: Séparation des composantes ASK en phase et en quadrature du signal 64QAM.

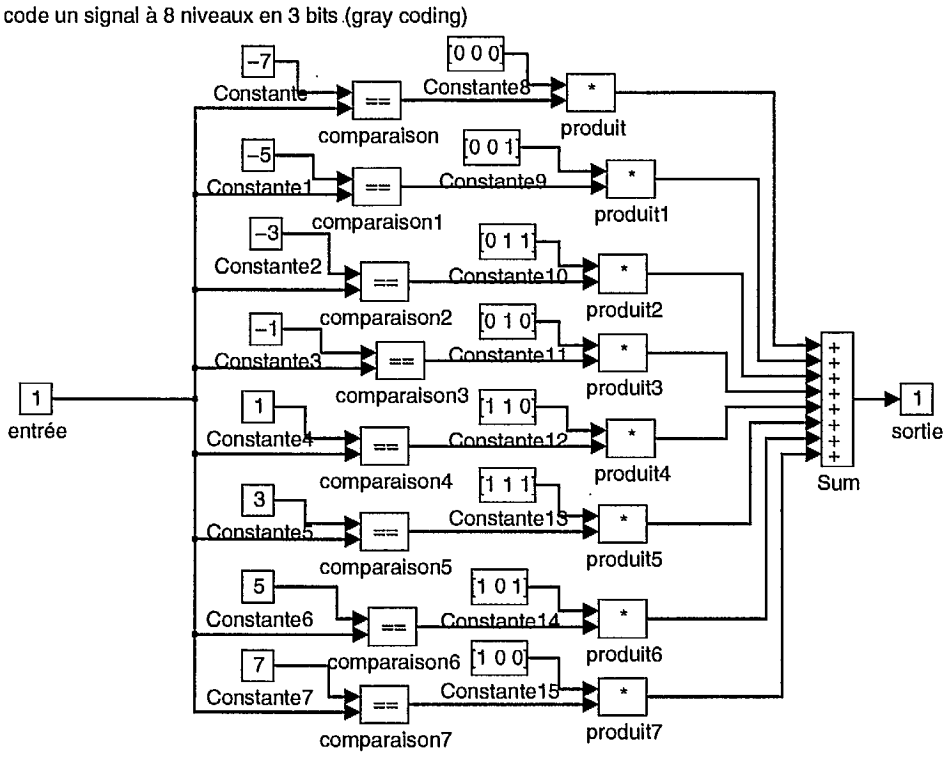

Figure 3.13: Décodeur de Gray à 3 bits (i.e. huit niveaux).

Algorithme de Viterbi

Le module *calcul des états parcourus*, montré à la Figure 3.14 a été conçu pour implémenter une partie de l'algorithme de décodage de codes convolutifs de Viterbi. Ce module détermine, à partir de la somme cumulative des métriques de branche, la séquence d'états la plus probable produite par le codeur d'Ungerboeck au transmetteur. La sortie de ce module consiste donc en trois bits représentant les états du treillis qui ont le plus probablement étés parcourus. Ici, les huit opérations sont effectuées parallèlement, permettant de calculer cette séquence plus rapidement. D'abord, les modules *état précédant et métriques pour XXX* déterminent l'état précédent le plus probable menant à chacun des états ainsi que les métriques de branche précédentes. Ensuite, les modules *calcul de la branche XXX* calculent les nouvelles métriques de branche arrivant à chaque état. Ces modules calculent également pour chaque état l'état parcouru cinq périodes plus tôt. L'état qui possède la métrique cumulative la plus faible a l'étiquette de son état parcouru cinq périodes plus tôt multiplié par le facteur "1", alors que les sept autres sont multipliés par "0" (grâce aux modules *logic1-8*). La somme de ceux-ci (le module *Sum*) conduit donc à l'état le plus probable parcouru cinq périodes plus tôt. Le module *plus petit métrique* produit le facteur "1" (ou "0") requis pour la multiplication selon que l'état correspondant possède la plus faible métrique ou non.

Les modules *état précédent et métriques pour XXX* (un pour chacun des huit états) servent à déterminer les états précédents les plus probables menant à chaque état ainsi que les métriques cumulatives correspondantes. Les calculs effectués dépendent du mot de code reçu ainsi que des métriques cumulatives des quatre états précédents possibles pour chaque état.

En observant le treillis du code convolutionnel, on remarque que les états [000, 010, 100 et 110] ne peuvent avoir comme états précédents que les états [000, 100, 001, 101] alors que les états [001, 011, 101, 111] ne peuvent être précédés que par les états [010, 110, 011, 111]. Les métriques cumulatives de chaque état pour la période précédente sont accessibles grâce aux modules *1* à *8* qui agissent ici de mémoire. Pour déterminer l'état précédent le plus probable pour un état quelconque, le module *état précédant et métriques pour XXX* correspondant compare les trois bits du mot de code reçu avec quatre différentes sous-séquences de trois bits représentant les mots de code valides pour les quatre états précédents possibles. Les modules *logic1-4* (il s'agit en fait de porte logiques XOR, i.e. *ou-exclusif*) sont utilisés pour que la comparaison se fasse au niveau binaire. Les trois bits résultant de la comparaison sont additionnés par les modules *Sum*. Le module *Sum5* fait alors la somme de ces quatre résultats pour déterminer si l'état en question est accessible à cet instant. En effet, si la somme des distances de Hamming des quatre mots de code possibles est plus grande que "4", on sait d'après les connexions dans le treillis, que l'état ne peut pas être atteint à cet instant donné. Si c'est le cas, les modules *Sum5*, *Constante5*, *compa*, *1*, *2*, *3*, *12* et *produit16* changent la nouvelle valeur de la métrique pour l'état en question (la deuxième sortie) à une valeur de "999", indiquant que l'état ne peut pas être atteint à cet instant. Par contre,

si l'état est valide, un seul des modules *Sum1* à *Sum4* a une sortie de valeur "0", les trois autres ayant une valeur de "1" ou "2" à leur sortie. Les modules *Sum6* à *Sum9* ajoutent à ces nombres les valeurs précédentes des métriques cumulatives pour les états correspondants. Les modules *compa1-6*, *logic5-18*, *Constante6-9*, *produit1-4* et *Sum10-11* déterminent alors laquelle et emmagazine l'état précédent survivant (i.e. les deux sorties de chaque module). Les modules *4-14* servent à vérifier si l'état précédent choisi est valide. Si l'état précédent ou la métrique calculée ne sont pas valides, alors les trois bits indiquant l'état précédent (la première sortie) sont alors fixés à la valeur "9 9 9".

Les modules *calcul de la branche XXX* ont été réalisés afin de calculer les nouvelles branches "survivantes" menant à chaque état. Ces modules calculent également pour chaque état, l'état parcouru cinq périodes plus tôt. Les trois bits à l'entrée représentent un des quatre états précédents valides. Cette entrée est comparée à l'aide des modules *compa1-4* à chacun de ces quatre états (les modules *Constante1-4*). L'état précédent choisi a sa branche multipliée par "1" alors que les trois autres sont multipliées par "0" à l'aide des modules *1-4*, *produit1-4* et *brancheXXX*. Le module *Sum* indique alors la branche choisie au module *Demux*. Les modules *Mux* et *Demux1* recalculent la nouvelle branche en décalant les états d'une position et en insérant l'état précédent indiqué à l'entrée en première position. Le dernier état est transmis à la sortie du module (la deuxième sortie). Les modules *5-16* servent à déterminer si la branche est valide. Si elle ne l'est pas, une branche illégale est transmise à la première sortie (la nouvelle branche) et un état illégal ([9 9 9]) est aussi transmis à la deuxième sortie (l'état estimé).

Le module *plus petit métrique* a comme entrée les métriques de chaque état. Sa fonction est de déterminer lequel des états a la plus petite métrique et de l'indiquer en fixant la sortie correspondante à "1" et les autres sept sorties à "0". Cette opération s'effectue à l'aide des modules de comparaison numérique ($<$, $=$ et $>$) et logiques (porte logique AND).

Les modules *délai4-5*, *Demux1-4* et *Mux1-2* réarrangent les bits représentant les états parcourus pour les deux branches (en phase et en quadrature) de façon à ce que les modules *décode 2 bits* puissent déterminer les quatre bits d'information codés. Le décodage se fait à l'aide d'un tableau dans les modules *décode 2bits* pour lesquels les entrées représentent une suite d'états (six bits). Le tableau a été construit à l'aide du treillis du code: pour toute suite d'états parcouru valide, on peut déterminer les deux bits codés. Les trois premiers bits représentent le premier état et les trois derniers bits représentent l'état suivant. On se retrouve donc avec les soixante-quatre combinaisons possibles de ces six bits. Cependant, certaines de ces combinaisons sont illégales puisque le codeur d'Ungerboeck restreint la séquence d'états selon le treillis du code.

x_1	00000000000000000000000000000000111111111111111111111111111111111
1er état: x_2	00000000000000001111111111111111000000000000000011111111111111111
x_3	00000000111111110000000011111111000000001111111100000000111111111
x_4	00001111000011110000111100001111000011110000111100001111000011111
2ième état: x_5	00110011001100110011001100110011001100110011001100110011001100110011
x_6	01010101010101010101010101010101010101010101010101010101010101010101
sortie: y_1	$0x0x1x1x0x0x1x1xx0x0x1x1x0x0x1x10x0x1x1x0x0x1x1xx0x0x1x1x0x0x1x1$
y_2	$0x1x0x1x0x1x0x1xx0x1x0x1x0x1x0x10x1x0x1x0x1x0x1xx0x1x0x1x0x1x0x1$

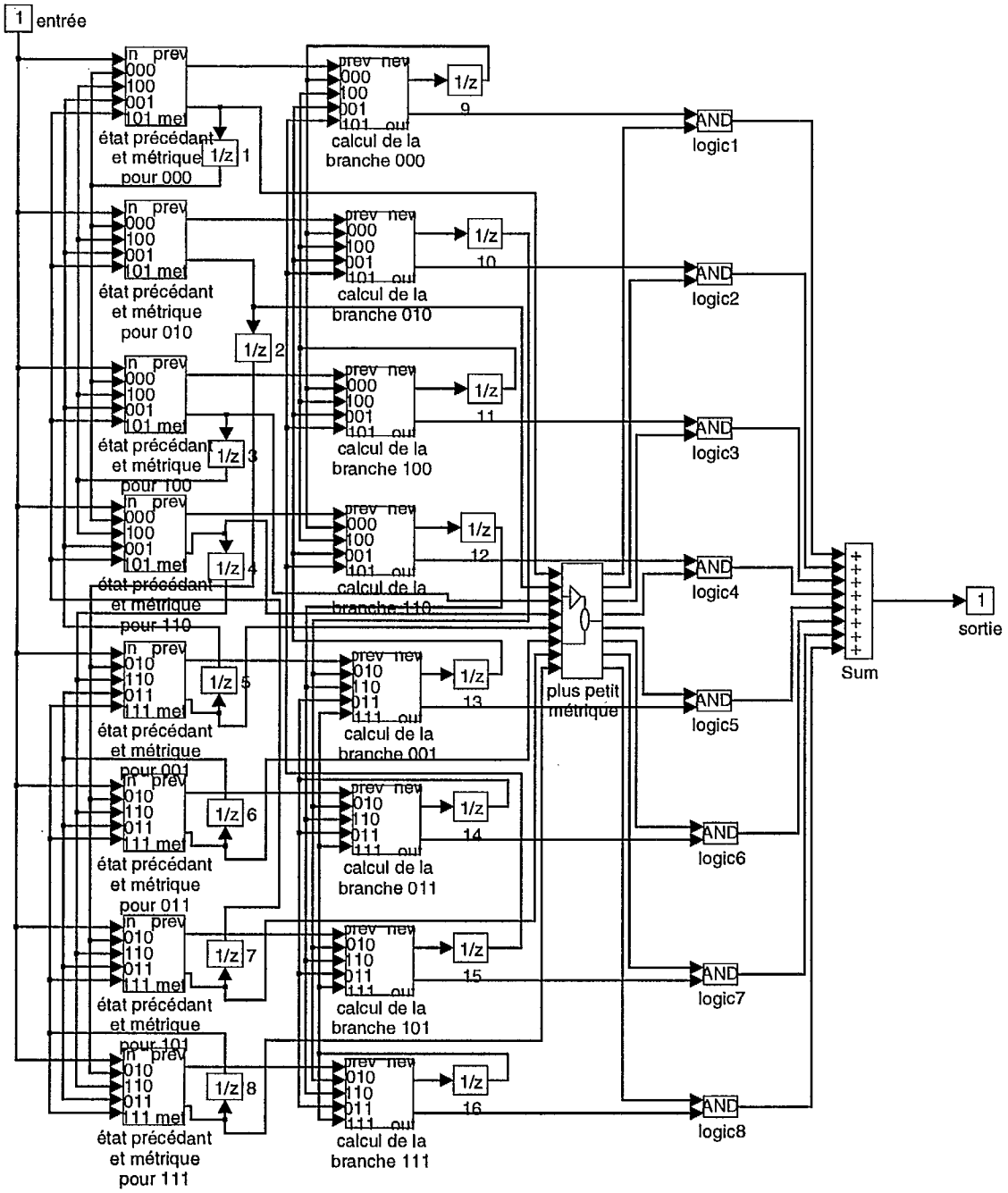

Figure 3.14: Décodeur de Viterbi.

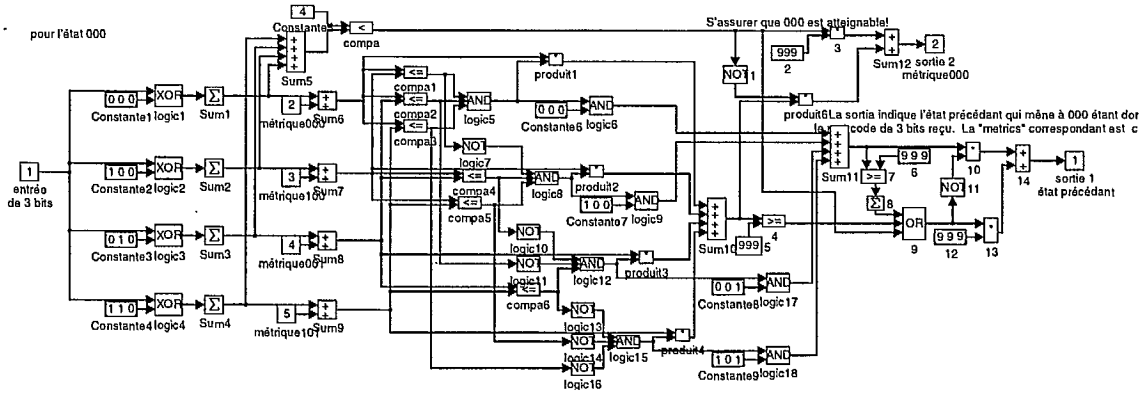

Figure 3.15: Calcul de la métrique cumulative.

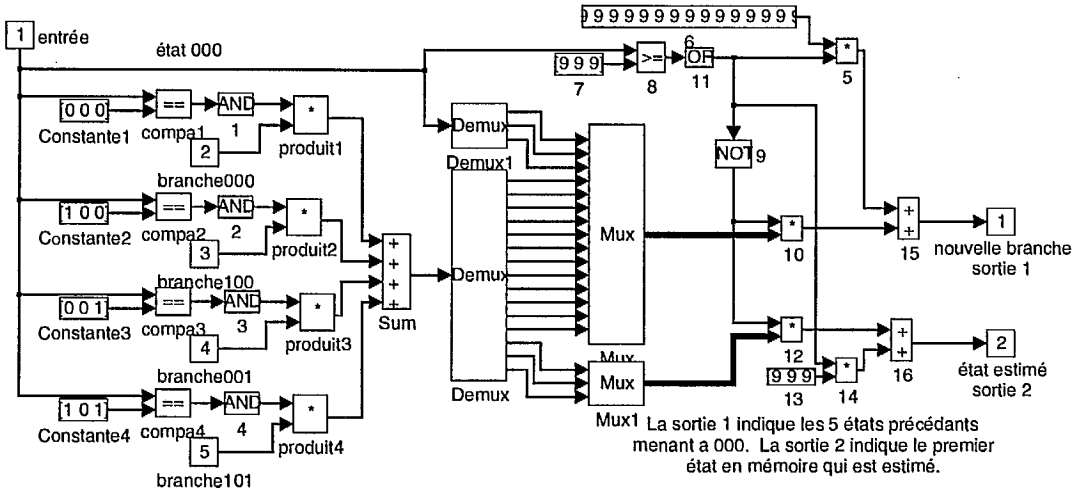

Figure 3.16: Détermination de chemin survivant.

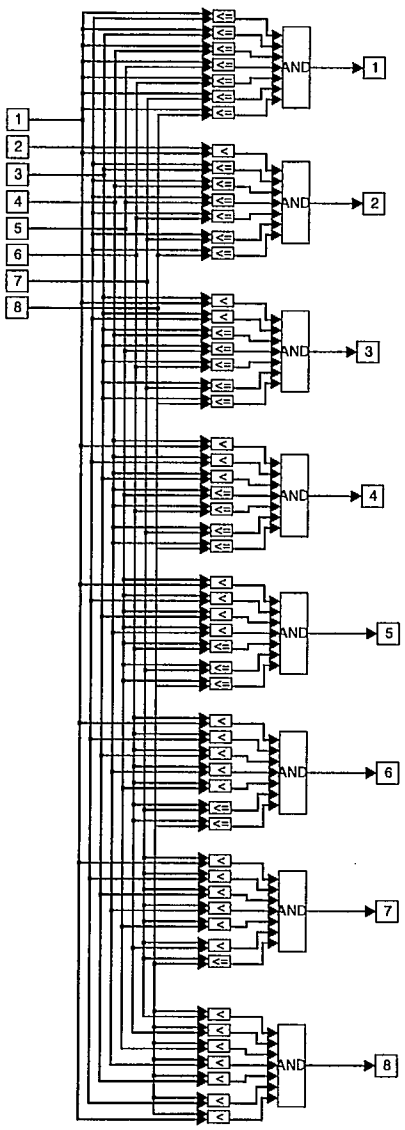

Figure 3.17: Détermination de la plus petite métrique de branche.

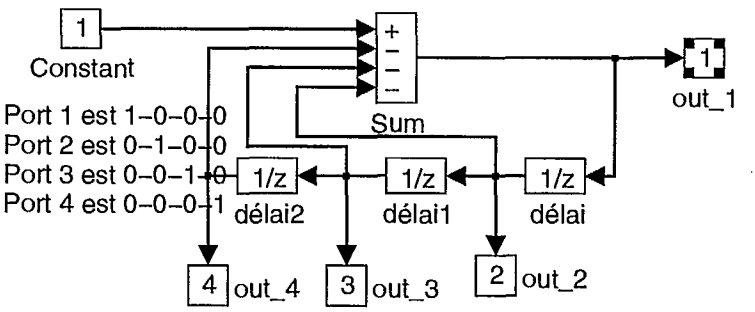

Figure 3.18: Horloge.

Démultiplexeur

Les quatre bits ainsi décodés, à savoir deux par décodeur de Viterbi, sont alors démultiplexés et replacés dans leur ordre original à l'aide des modules *Demux1-2*, *horloge cycle 1/4*, *produit1-4* et *Sum1* (les modules *marche*, *échantillon* et *produit5* sont nécessaires afin de s'assurer qu'aucune sortie ne sera acceptée jusqu'à ce que le premier bit provenant de la source d'information arrive).

Horloge

Le module *horloge cycle 1/4* permet de générer quatre signaux qui servent à échantillonner les sorties des modules *décode 2 bits* de façon à réorganiser l'information binaire. À chaque instant, un seul de ces quatre signaux a une valeur de "1" (les trois autres ont chacun une valeur de "0"). Les modules *délaiX* agissent comme mémoire.

3.6 Analyse des données

Taux d'erreur

Le module de calcul de taux d'erreur *BER* a deux entrées: la séquence de bits originaux provenant directement du transmetteur (avec le retard approprié) et la séquence de bits décodés à la sortie du récepteur par le décodeur de Viterbi. Ce module évalue le taux

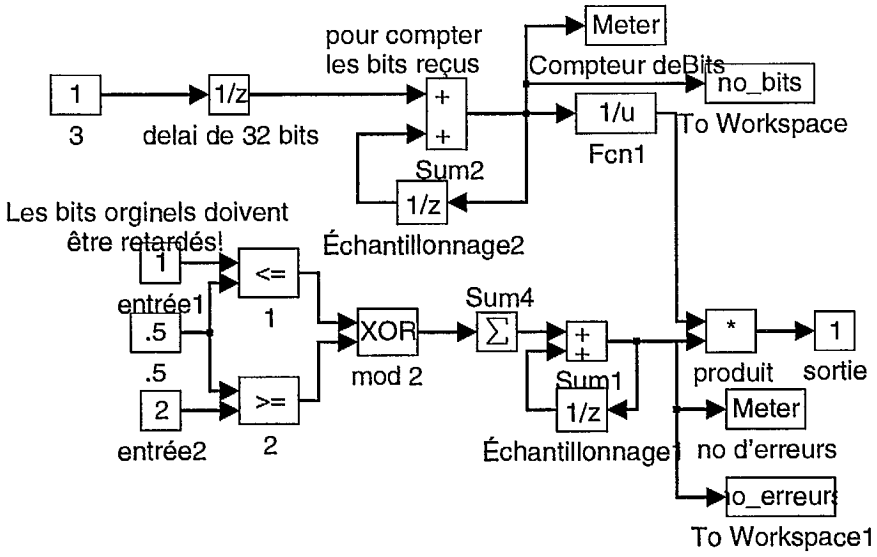

Figure 3.19: Module permettant de calculer le taux d'erreur.

d'erreur instantané du système de télécommunication. Le résultat peut être observé dans la fenêtre appelée "BER instantané". Les bits sont comparés un à un, mais un niveau de seuil de 0.5 a été ajouté pour corriger des erreurs de niveaux introduit par le logiciel *SIMULINK* (au lieu de retrouver 0 on obtient 1^{-16}). Les modules *mod 2* et *Sum4* servent seulement si la comparaison se fait à plus d'un bit à la fois. Les modules *Sum1* et *Échantillonnage1* comptent le nombre total des bits reçus erronés. Pour leur part, les modules *3*, *délai de 32 bits*, *Sum2* et *Échantillonnage2* comptent le nombre total des bits reçus. Le module de délai est nécessaire afin que le compte ne débute que lorsque le premier bit est disponible à la sortie du récepteur. Le module *Fcn* inverse son entrée, et avec le module *produit*, le rapport du nombre de bits erronés sur le nombre total de bits reçus est calculé. Les modules *Meter* montrent la valeur numérique de leur entrée à l'écran de l'ordinateur durant la simulation alors que les modules *To Workspace* enregistrent la valeur de leur entrée dans la fenêtre de commandes *MATLAB*.

Les figures 3.20 et 3.21 montrent, à titre illustratif, la constellation des signaux pour le système de télécommunication TCM 64QAM avant et après le module générant le bruit additif gaussien blanc obtenu par simulation (fichier de simulation *MATLAB* "test25.m").

3.7 Conclusion

Dans ce chapitre, on a présenté le programme de simulation développé pour la phase 2 du projet de recherche. Les modules de programmation permettant d'effectuer la production

de données, la modulation codée TCM 64 QAM, la génération de bruit additif gaussien, le décodage convolutif à l'aide de l'algorithme de Viterbi ainsi que le calcul du taux d'erreur et la représentation dans le plan complexe des divers signaux sont complétés. Les modules de programmation à réaliser pour compléter le programme sont énumérés ci-après.

Le système de télécommunication simulé (e.g. fichier de simulation "test25.m") devra être validé en comparant des courbes de probabilité d'erreurs résultantes (obtenues de simulations suffisamment longues) avec les courbes de probabilités théoriques (i.e. calcul de l'expression mathématique théorique de la probabilité d'erreur d'un système TCM 64QAM). Un ordinateur (e.g. station de travail "SUN") avec un minimum de 16 Megaoctets de mémoire vive devrait suffire.

Le multiplexage par division en fréquences orthogonales (OFDM) devra être implémenté à l'aide de la transformée de Fourier. La longueur de la trame de la transformée est de 1024 bits. Les modules *Complex Buffer*, *Complex Unbuffer*, *FFT* $C \rightarrow C$ et *IFFT* $C \rightarrow C$ du *DSP Blockset* pourront être utilisées sur les symboles 64QAM (avant et après l'addition du bruit) pour obtenir cette modulation. Les résultats de simulation devront être validés.

Le canal de transmission devra être modifié pour représenter un canal de propagation multitrajet sélectif en fréquence (en considérant la largeur de bande de 6 MHz d'un signal de télédiffusion NTSC).

Les codes correcteurs d'erreur Reed-Solomon (e.g. $RS(255, 239, 8)$ devront être ajoutés au système pour réduire davantage la probabilité d'erreur en fonction du rapport signal à bruit.

Un estimateur du canal devra aussi être construit et sa performance en fonction de la rapidité de changement du canal de Rayleigh devra également être déterminée.

Remarque: Les modules du *DSP Blockset* qui traitent les signaux complexes produisent des erreurs qui doivent être corrigées avant de pouvoir obtenir des résultats valides. Les erreurs apparaissent lorsque l'entrée d'un module complexe est un vecteur de nombres complexes. Le problème vient du fait que les modules *MUX* et *DEMUX* de *SIMULINK* ne vérifient pas les dimensions de leurs entrées et leur sorties respectivement (le nombre de bits multiplexés sur chaque ligne). Ceux-ci supposent plutôt que la dimension est toujours unitaire. Il faut donc "manuellement" indiquer leur dimension pour chaque module en choisissant un vecteur dont les éléments (un pour chaque ligne) représentent le nombre de bits sur ces lignes.

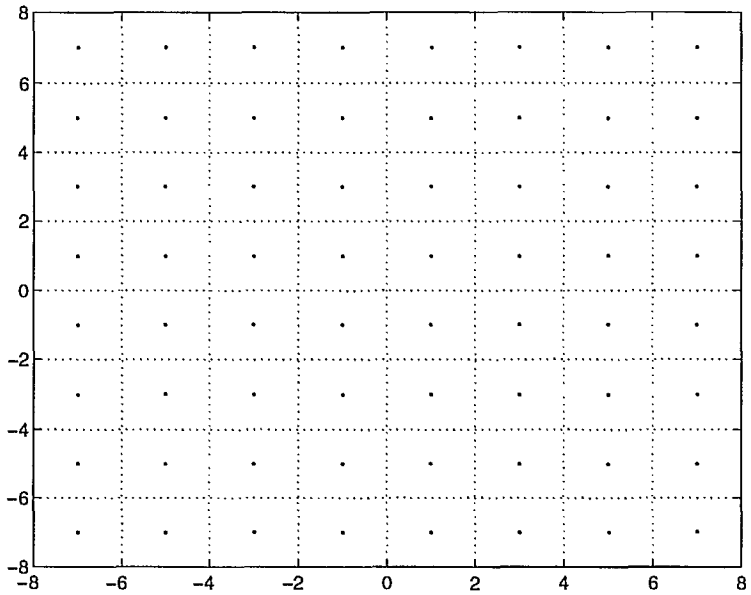

Figure 3.20: Constellation des signaux TCM 64QAM à l'entrée du module de bruit additif gaussien blanc (fichier de simulation "test25.m").

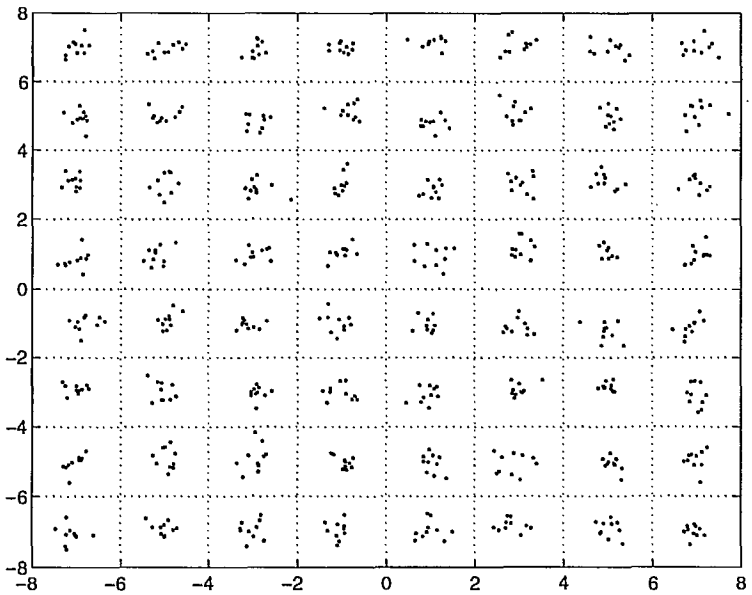

Figure 3.21: Constellation des signaux TCM 64QAM obtenu à la sortie du module de bruit additif gaussien blanc (fichier "test25.m").

Chapter 4

Égalisation

4.1 Introduction

Dans ce chapitre, on présente une brève description de techniques d'estimation du canal de propagation radioélectrique en vue d'appliquer les techniques de codage de canal [CT95] en exploitant cette connaissance des caractéristiques statistiques du canal. Dans la section suivante, section 4.2, on rapporte le travail de recherche déjà effectués par d'autres chercheurs [CETM95, MMV95] pour les canaux de télévision numérique.

4.2 Estimation du canal appliquée au codage

Dans le rapport interne du CRC [CETM95], on étudie l'application d'un estimateur de canal avec rétroaçtion filtrée des décisions pour la transmission TVHD codée à porteuses multiples MCM (*"multi-carrier modulation"*). Le système de codage MCM est brièvement décrit, suivi de la description de l'estimateur du canal et finalement sont présentés des résultats de simulations. Pour le système MCM en question, l'information binaire est d'abord transformée l'aide d'une modulation d'amplitude modulation QAM. Les symboles complexes sont ensuite groupés en trames $\mathbf{I} = \{I_n\}$ de longueur N. Une transformée inverse discrète de Fourier de N points est alors appliquée sur ces trames $\mathbf{I}$ produisant ainsi la fonction d'un synthétiseur de fréquence. À chaque nouvelle trame ainsi produite est ajoutée une séquence-test contenant les $L-1$ derniers coefficients ($N \geq L$) de la trame précédente. Ceci rend le système MCM *insensible* aux distorsions créées par la propagation multivoies et ce pour des retards allant jusqu'à $(L-1)T$. Le rôle de cette séquence-test consiste non seulement à éviter l'interférence entre des trames consécutives, mais aussi à convertir la convolution linéaire de la séquence de données $\mathbf{i} = \{i_m\}$ par la réponse impulsionnelle du canal discret $\mathbf{h} = \{h_m\}$ en une convolution

circulaire:

$$\mathbf{r} = \mathbf{i} \otimes \mathbf{h} + \mathbf{z}$$

où $\mathbf{z} = \{z_m\}$ est un vecteur de bruit et $\mathbf{r} = \{r_m\}$ est le vecteur du signal reçu. En faisant la transformée de Fourier discrète du signal reçu on obtient le vecteur $\{R_n\} = H_n I_n + Z_n$, avec $n = 0, 1, ..., N-1$, où les $\{Z_n\}$ sont des variables de bruit aléatoires indépendantes avec distribution gaussienne complexe de moyenne nulle (représentant le canal avec bruit additif gaussien (AWGN)). Ici les coefficients $\{H_n\}$ représentent la fonction de transfert du canal à voies multiples:

$$H_n = \sum_{m=0}^{L-1} h_m e^{-j2\pi \frac{nm}{N}} \qquad \text{pour } n = 0, 1, \ldots, N-1$$

L'estimation du canal se fait à partir du vecteur du signal reçu $\mathbf{R} = \{R_n\}$ ainsi que sur le vecteur de données estimées $\mathbf{I} = \{I_n\}$. Un diviseur à l'entrée de l'estimateur divise chaque symbole R_n par le symbole correspondant I_n afin de produire une estimation $\tilde{\mathbf{H}}$ du canal:

$$\tilde{H}_n = \frac{R_n}{H_n} \qquad \text{pour } n = 0, \ldots, N-1$$

Des simulations sur ordinateur ont été effectuées par les auteurs afin de déterminer le taux de bits erronés d'un système 64QAM-MCM sans codage de canal avec l'algorithme d'estimation du canal. La longueur de la trame DFT $N = 1024$ et la longueur de la séquence de test $L - 1 = 127$. Avec des symboles de période $T = 183$ ns (pour 10% de *"roll-off"*), une durée de trajets multivoies allant jusqu'à 127×183 ns $= 23.2$ μs est tolérée. Les trames sont de même longueurs que celles de la transformée de Fourier discrète du système MCM, contenant des symboles 64QAM. Les symboles complexes 64QAM sont réarrangés dans leurs trames respectives, à l'aide d'un entrelaceur. Afin de permettre le décodage d'une trame indépendamment des autres, un codage en treillis est effectué sur chacun d'eux. Les trames décodées $\hat{\mathbf{I}}$ sont utilisées pour estimer le vecteur du canal $\mathbf{H}$ (estimateur de canal avec rétroaction filtrée des décisions).

Deux différents modèles de canal ont été simulés. Le premier modèle consiste en un canal de Rayleigh dispersif avec un profil d'intensité multivoie exponentiel. Le deuxième modèle consiste en un canal multivoies à valeur absolue constante, avec phases aléatoires indépendantes pour chacune des voies. Ce modèle inclut 5 échos avec des amplitudes respectives de -20 dB, -20 dB, -10 dB, -14 dB et -18 dB relativement au trajet en ligne directe. La première estimation du canal est obtenue à l'aide de la séquence de test. Cette trame peut être répétée à chaque nouvelle image (c'est-à-dire à chaque 33 ms). Par exemple, si le filtre du transmetteur a un facteur d'adoucissement (ou *"roll off"*) de 10%, des symboles de

période $T = \frac{1.1}{6}$ MHz = 183 ns, des trames DFT de 1024 symboles et un préfixe circulaire de 127 symboles, alors la durée résultante T_f d'une trame sera égale à 183 ns $\times$ (1024 + 127), c'est-à-dire $T_f = 211$ ms. Cette trame retransmise à chaque $\frac{33 \text{ ms}}{211 \ \mu\text{s}} = 156$ trames DFT, se traduit par une redondance, ou excédent, (i.e. *"overhead"*) ne dépassant pas 0.7%. Les résultats obtenus supposent que la sortie du décodeur a un taux de bit erroné allant jusqu'à 10^{-4}. Il reste donc à démontrer qu'un tel taux est réalisable avec un code de correction d'erreur.

L'article *"CD3-OFDM: A New Channel Estimation Method to Improve the Spectrum Efficiency in Digital Terrestrial Television Systems"* [MMV95] de V. Mignone, A. Morello, M. Visintin, du RAI Research Centre en Italie, décrit un système d'estimation de canal très similaire à celui développé pour le CRC [CETM95] que nous venons de décrire. Les différences principales sont qu'une largeur de bande de fréquence de 7.5 MHz a été utilisée (pour l'Europe, alors qu'en Amérique, le standard est 6 MHz). Aussi, deux canaux de transmission différents ont été étudiés et simulés. Le premier canal consistait en un canal "fixe" représentant une réception "fixe" avec une antenne directive et un seul écho de moins de -10 dB alors que le deuxième modèle consistait en un canal "portatif", représentant une réception avec antenne omnidirectionelle avec un seul écho de moins de -4 dB. En plus d'une modulation QAM, une modulation QPSK a également été simulée avec le même succès, appuyant l'efficacité de ce système d'estimation du canal.

Chapter 5

Travaux futurs

Dans ce rapport final on a présenté les schéma-blocs des sous-systèmes développés avec les programmes *Matlab* et *Simulink* développés pour la phase 2 de ce projet. Dans les paragraphes qui suivent, on indique les travaux futurs de recherche envisagés dans le cadre de ce projet. Afin de simuler la performance du système de télévision avancé (ATV) employant les techniques d'estimation et de codage de canal, les logiciels *Matlab* et *Simulink* ont été choisis comme outils de programmation. Les facteurs principaux qui ont mené à ce choix sont leur disponibilité et flexibilité. Différents codes correcteurs d'erreurs externes et internes [CT95] (e.g., codes Reed-Solomon, modulation codée, algorithmes de décodage de Gorenstein pour les codes blocs et de Viterbi pour le TCM, entrelacement de bits et de symboles complexes, etc.) pourront être construits et utilisés avec le système d'estimation de canal décrit dans les références [MMV95, CETM95] afin de déterminer si le taux de bit erroné convient à celui requis pour le système de télévision avancé.

Les étapes à suivre pour parvenir à cet objectif sont envisagées comme suit. Un premier modèle représentant un canal de transmission ATV avec bruit blanc additif a déjà été réalisé. Par la suite, d'autres modèles de canal pourront être construits afin de simuler des canaux à propagation multivoie et à atténuation Rayleigh afin de mieux refléter les conditions prévalant dans un véritable canal de radiocommunication. On compte également inclure dans nos simulations un égalisateur adaptatif permettant d'exploiter la réponse impulsionnelle du canal de transmission à large bande.

Un modulateur-démodulateur d'amplitude à bande résiduelle à huit niveaux VSB-8 (*"Vestigial Sideband Modulation"*), tel que proposé par les intervenants Américains, pourra être conçu afin d'évaluer la technique de modulation qui sera vraisemblablement adoptée au Canada pour la diffusion ATV. On simulera par la suite le code interne TCM avec entrelacement, en y incluant un système de détection du canal semblable à ceux décrits plus haut dans ce chapitre. Un modulateur-démodulateur COFDM (*"multi-carrier"*) pourra être développé

pouvant ainsi être utilisé en lieu et place du modem VSB-8 (*"single carrier"*) puisque c'est cette dernière méthode de modulation que semblent vouloir adopter les administrations européennes.

L'étude analytique de la performance théorique entreprise dans la phase 1 du projet est également prometteuse et devrait conduire à de nouveaux résultats sur la fiabilité des canaux de télévision haute définition. Les techniques de codage de canal conduisant à une *dégradation progressive* de la qualité de liaison constituent un domaine de recherche particulièrement intéressant et pertinent.

Appendix A

Description des Corps de Galois

Le Corps de Galois $GF[q] = GF[16]$ (voir annexe A.1 à la page 53) est en réalité une extension $GF[2^m] = GF[2^4]$ du Corps de Galois binaire $GF[2]$. De la même manière, le Corps de Galois $GF[q] = GF[256]$, donné à l'annexe A.2 (page 54) est une extension $GF[2^m]$ (où $m = 8$) du Corps de Galois binaire $GF[2]$.

A.1 Corps de Galois $GF[q] = GF[16]$

Table A.1: Corps de Galois $GF[q] = GF[16]$

élément de $GF[16]$	polynôme	forme binaire	forme décimale
α^0	1	0 0 0 1	1
α^1	α	0 0 1 0	2
α^2	α^2	0 1 0 0	4
α^3	α^3	1 0 0 0	8
α^4	$\alpha + 1$	0 0 1 1	3
α^5	$\alpha^2 + \alpha$	0 1 1 0	6
α^6	$\alpha^3 + \alpha^2$	1 1 0 0	12
α^7	$\alpha^3 + \alpha + 1$	1 0 1 1	11
α^8	$\alpha^2 + 1$	0 1 0 1	5
α^9	$\alpha^3 + \alpha$	1 0 1 0	10
α^{10}	$\alpha^2 + \alpha + 1$	0 1 1 1	7
α^{11}	$\alpha^3 + \alpha^2 + \alpha$	1 1 1 0	14
α^{12}	$\alpha^3 + \alpha^2 + \alpha + 1$	1 1 1 1	15
α^{13}	$\alpha^3 + \alpha^2 + 1$	1 1 0 1	13
α^{14}	$\alpha^3 + 1$	1 0 0 1	9

A.2 Corps de Galois $GF[q] = GF[256]$

Table A.2: Corps de Galois $GF[q] = GF[256]$ (partie 1)

élément de $GF[256]$	polynôme	forme binaire	forme décimale
α^0	1	0 0 0 0 0 0 0 1	1
α^1	α	0 0 0 0 0 0 1 0	2
α^2	α^2	0 0 0 0 0 1 0 0	4
α^3	α^3	0 0 0 0 1 0 0 0	8
α^4	α^4	0 0 0 1 0 0 0 0	16
α^5	α^5	0 0 1 0 0 0 0 0	32
α^6	α^6	0 1 0 0 0 0 0 0	64
α^7	α^7	1 0 0 0 0 0 0 0	128
α^8	$\alpha^4 + \alpha^3 + \alpha^2 + 1$	0 0 0 1 1 1 0 1	29
α^9	$\alpha^5 + \alpha^4 + \alpha^3 + \alpha$	0 0 1 1 1 0 1 0	58
α^{10}	$\alpha^6 + \alpha^5 + \alpha^4 + \alpha^2$	0 1 1 1 0 1 0 0	116
α^{11}	$\alpha^7 + \alpha^6 + \alpha^5 + \alpha^3$	1 1 1 0 1 0 0 0	232
α^{12}	$\alpha^7 + \alpha^6 + \alpha^3 + \alpha^2 + 1$	1 1 0 0 1 1 0 1	205
α^{13}	$\alpha^7 + \alpha^2 + \alpha + 1$	1 0 0 0 0 1 1 1	135
α^{14}	$\alpha^4 + \alpha + 1$	0 0 0 1 0 0 1 1	19
α^{15}	$\alpha^5 + \alpha^2 + \alpha$	0 0 1 0 0 1 1 0	38
α^{16}	$\alpha^6 + \alpha^3 + \alpha^2$	0 1 0 0 1 1 0 0	76
α^{17}	$\alpha^7 + \alpha^4 + \alpha^3$	1 0 0 1 1 0 0 0	152
α^{18}	$\alpha^5 + \alpha^3 + \alpha^2 + 1$	0 0 1 0 1 1 0 1	45
α^{19}	$\alpha^6 + \alpha^4 + \alpha^3 + \alpha$	0 1 0 1 1 0 1 0	90
α^{20}	$\alpha^7 + \alpha^5 + \alpha^4 + \alpha^2$	1 0 1 1 0 1 0 0	180
α^{21}	$\alpha^6 + \alpha^5 + \alpha^4 + \alpha^2 + 1$	0 1 1 1 0 1 0 1	117
α^{22}	$\alpha^7 + \alpha^6 + \alpha^5 + \alpha^3 + \alpha$	1 1 1 0 1 0 1 0	234
α^{23}	$\alpha^7 + \alpha^6 + \alpha^3 + 1$	1 1 0 0 1 0 0 1	201
α^{24}	$\alpha^7 + \alpha^3 + \alpha^2 + \alpha + 1$	1 0 0 0 1 1 1 1	143
α^{25}	$\alpha + 1$	0 0 0 0 0 0 1 1	3
α^{26}	$\alpha^2 + \alpha$	0 0 0 0 0 1 1 0	6
α^{27}	$\alpha^3 + \alpha^2$	0 0 0 0 1 1 0 0	12
α^{28}	$\alpha^4 + \alpha^3$	0 0 0 1 1 0 0 0	24
α^{29}	$\alpha^5 + \alpha^4$	0 0 1 1 0 0 0 0	48
α^{30}	$\alpha^6 + \alpha^5$	0 1 1 0 0 0 0 0	96
α^{31}	$\alpha^7 + \alpha^6$	1 1 0 0 0 0 0 0	192
α^{32}	$\alpha^7 + \alpha^4 + \alpha^3 + \alpha^2 + 1$	1 0 0 1 1 1 0 1	157
α^{33}	$\alpha^5 + \alpha^2 + \alpha + 1$	0 0 1 0 0 1 1 1	39
α^{34}	$\alpha^6 + \alpha^3 + \alpha^2 + \alpha$	0 1 0 0 1 1 1 0	78
α^{35}	$\alpha^7 + \alpha^4 + \alpha^3 + \alpha^2$	1 0 0 1 1 1 0 0	156
α^{36}	$\alpha^5 + \alpha^2 + 1$	0 0 1 0 0 1 0 1	37
α^{37}	$\alpha^6 + \alpha^3 + \alpha$	0 1 0 0 1 0 1 0	74
α^{38}	$\alpha^7 + \alpha^4 + \alpha^2$	1 0 0 1 0 1 0 0	148
α^{39}	$\alpha^5 + \alpha^4 + \alpha^2 + 1$	0 0 1 1 0 1 0 1	53
α^{40}	$\alpha^6 + \alpha^5 + \alpha^3 + \alpha$	0 1 1 0 1 0 1 0	106
α^{41}	$\alpha^7 + \alpha^6 + \alpha^4 + \alpha^2$	1 1 0 1 0 1 0 0	212
α^{42}	$\alpha^7 + \alpha^5 + \alpha^4 + \alpha^2 + 1$	1 0 1 1 0 1 0 1	181
α^{43}	$\alpha^6 + \alpha^5 + \alpha^4 + \alpha^2 + \alpha + 1$	0 1 1 1 0 1 1 1	119
α^{44}	$\alpha^7 + \alpha^6 + \alpha^5 + \alpha^3 + \alpha^2 + \alpha$	1 1 1 0 1 1 1 0	238

Tableau A.2: Corps de Galois $GF[q] = GF[256]$ (partie 2)

élément de $GF[256]$	polynôme								forme binaire								forme décimale
α^{45}	α^7	$+\alpha^6$						$+1$	1	1	0	0	0	0	0	1	193
α^{46}	α^7			$+\alpha^4$	$+\alpha^3$	$+\alpha^2$	$+\alpha$	$+1$	1	0	0	1	1	1	1	1	159
α^{47}			α^5				$+\alpha$	$+1$	0	0	1	0	0	0	1	1	35
α^{48}		α^6				$+\alpha^2$	$+\alpha$		0	1	0	0	0	1	1	0	70
α^{49}	α^7				$+\alpha^3$	$+\alpha^2$			1	0	0	0	1	1	0	0	140
α^{50}						α^2		$+1$	0	0	0	0	0	1	0	1	5
α^{51}					α^3		$+\alpha$		0	0	0	0	1	0	1	0	10
α^{52}				α^4		$+\alpha^2$			0	0	0	1	0	1	0	0	20
α^{53}			α^5		$+\alpha^3$				0	0	1	0	1	0	0	0	40
α^{54}		α^6		$+\alpha^4$					0	1	0	1	0	0	0	0	80
α^{55}	α^7		$+\alpha^5$						1	0	1	0	0	0	0	0	160
α^{56}		α^6		$+\alpha^4$	$+\alpha^3$	$+\alpha^2$		$+1$	0	1	0	1	1	1	0	1	93
α^{57}	α^7		$+\alpha^5$	$+\alpha^4$	$+\alpha^3$		$+\alpha$		1	0	1	1	1	0	1	0	186
α^{58}		α^6	$+\alpha^5$		$+\alpha^3$			$+1$	0	1	1	0	1	0	0	1	105
α^{59}	α^7	$+\alpha^6$		$+\alpha^4$			$+\alpha$		1	1	0	1	0	0	1	0	210
α^{60}	α^7		$+\alpha^5$	$+\alpha^4$	$+\alpha^3$			$+1$	1	0	1	1	1	0	0	1	185
α^{61}		α^6	$+\alpha^5$		$+\alpha^3$	$+\alpha^2$	$+\alpha$	$+1$	0	1	1	0	1	1	1	1	111
α^{62}	α^7	$+\alpha^6$		$+\alpha^4$	$+\alpha^3$	$+\alpha^2$	$+\alpha$		1	1	0	1	1	1	1	0	222
α^{63}	α^7		$+\alpha^5$					$+1$	1	0	1	0	0	0	0	1	161
α^{64}		α^6		$+\alpha^4$	$+\alpha^3$	$+\alpha^2$	$+\alpha$	$+1$	0	1	0	1	1	1	1	1	95
α^{65}	α^7		$+\alpha^5$	$+\alpha^4$	$+\alpha^3$	$+\alpha^2$	$+\alpha$		1	0	1	1	1	1	1	0	190
α^{66}		α^6	$+\alpha^5$					$+1$	0	1	1	0	0	0	0	1	97
α^{67}	α^7	$+\alpha^6$					$+\alpha$		1	1	0	0	0	0	1	0	194
α^{68}	α^7			$+\alpha^4$	$+\alpha^3$			$+1$	1	0	0	1	1	0	0	1	153
α^{69}			α^5		$+\alpha^3$	$+\alpha^2$	$+\alpha$	$+1$	0	0	1	0	1	1	1	1	47
α^{70}		α^6		$+\alpha^4$	$+\alpha^3$	$+\alpha^2$	$+\alpha$		0	1	0	1	1	1	1	0	94
α^{71}	α^7		$+\alpha^5$	$+\alpha^4$	$+\alpha^3$	$+\alpha^2$			1	0	1	1	1	1	0	0	188
α^{72}		α^6	$+\alpha^5$			$+\alpha^2$		$+1$	0	1	1	0	0	1	0	1	101
α^{73}	α^7	$+\alpha^6$			$+\alpha^3$		$+\alpha$		1	1	0	0	1	0	1	0	202
α^{74}	α^7				$+\alpha^3$			$+1$	1	0	0	0	1	0	0	1	137
α^{75}					α^3	$+\alpha^2$	$+\alpha$	$+1$	0	0	0	0	1	1	1	1	15
α^{76}				α^4	$+\alpha^3$	$+\alpha^2$	$+\alpha$		0	0	0	1	1	1	1	0	30
α^{77}			α^5	$+\alpha^4$	$+\alpha^3$	$+\alpha^2$			0	0	1	1	1	1	0	0	60
α^{78}		α^6	$+\alpha^5$	$+\alpha^4$	$+\alpha^3$				0	1	1	1	1	0	0	0	120
α^{79}	α^7	$+\alpha^6$	$+\alpha^5$	$+\alpha^4$					1	1	1	1	0	0	0	0	240
α^{80}	α^7	$+\alpha^6$	$+\alpha^5$	$+\alpha^4$	$+\alpha^3$	$+\alpha^2$		$+1$	1	1	1	1	1	1	0	1	253
α^{81}	α^7	$+\alpha^6$	$+\alpha^5$			$+\alpha^2$	$+\alpha$	$+1$	1	1	1	0	0	1	1	1	231
α^{82}	α^7	$+\alpha^6$		$+\alpha^4$			$+\alpha$	$+1$	1	1	0	1	0	0	1	1	211
α^{83}	α^7		$+\alpha^5$	$+\alpha^4$	$+\alpha^3$		$+\alpha$	$+1$	1	0	1	1	1	0	1	1	187
α^{84}		α^6	$+\alpha^5$		$+\alpha^3$		$+\alpha$	$+1$	0	1	1	0	1	0	1	1	107
α^{85}	α^7	$+\alpha^6$		$+\alpha^4$		$+\alpha^2$	$+\alpha$		1	1	0	1	0	1	1	0	214
α^{86}	α^7		$+\alpha^5$	$+\alpha^4$				$+1$	1	0	1	1	0	0	0	1	177
α^{87}		α^6	$+\alpha^5$	$+\alpha^4$	$+\alpha^3$	$+\alpha^2$	$+\alpha$	$+1$	0	1	1	1	1	1	1	1	127
α^{88}	α^7	$+\alpha^6$	$+\alpha^5$	$+\alpha^4$	$+\alpha^3$	$+\alpha^2$	$+\alpha$		1	1	1	1	1	1	1	0	254
α^{89}	α^7	$+\alpha^6$	$+\alpha^5$					$+1$	1	1	1	0	0	0	0	1	225
α^{90}	α^7	$+\alpha^6$		$+\alpha^4$	$+\alpha^3$	$+\alpha^2$	$+\alpha$	$+1$	1	1	0	1	1	1	1	1	223
α^{91}	α^7		$+\alpha^5$				$+\alpha$	$+1$	1	0	1	0	0	0	1	1	163
α^{92}		α^6		$+\alpha^4$	$+\alpha^3$		$+\alpha$	$+1$	0	1	0	1	1	0	1	1	91
α^{93}	α^7		$+\alpha^5$	$+\alpha^4$		$+\alpha^2$	$+\alpha$		1	0	1	1	0	1	1	0	182
α^{94}		α^6	$+\alpha^5$	$+\alpha^4$				$+1$	0	1	1	1	0	0	0	1	113

Tableau A.2: Corps de Galois $GF[q] = GF[256]$ (partie 3)

élément de $GF[256]$	polynôme	forme binaire	forme décimale
α^{95}	$\alpha^7 + \alpha^6 + \alpha^5 + \alpha$	1 1 1 0 0 0 1 0	226
α^{96}	$\alpha^7 + \alpha^6 + \alpha^4 + \alpha^3 + 1$	1 1 0 1 1 0 0 1	217
α^{97}	$\alpha^7 + \alpha^5 + \alpha^3 + \alpha^2 + \alpha + 1$	1 0 1 0 1 1 1 1	175
α^{98}	$\alpha^6 + \alpha + 1$	0 1 0 0 0 0 1 1	67
α^{99}	$\alpha^7 + \alpha^2 + \alpha$	1 0 0 0 0 1 1 0	134
α^{100}	$\alpha^4 + 1$	0 0 0 1 0 0 0 1	17
α^{101}	$\alpha^5 + \alpha$	0 0 1 0 0 0 1 0	34
α^{102}	$\alpha^6 + \alpha^2$	0 1 0 0 0 1 0 0	68
α^{103}	$\alpha^7 + \alpha^3$	1 0 0 0 1 0 0 0	136
α^{104}	$\alpha^3 + \alpha^2 + 1$	0 0 0 0 1 1 0 1	13
α^{105}	$\alpha^4 + \alpha^3 + \alpha$	0 0 0 1 1 0 1 0	26
α^{106}	$\alpha^5 + \alpha^4 + \alpha^2$	0 0 1 1 0 1 0 0	52
α^{107}	$\alpha^6 + \alpha^5 + \alpha^3$	0 1 1 0 1 0 0 0	104
α^{108}	$\alpha^7 + \alpha^6 + \alpha^4$	1 1 0 1 0 0 0 0	208
α^{109}	$\alpha^7 + \alpha^5 + \alpha^4 + \alpha^3 + \alpha^2 + 1$	1 0 1 1 1 1 0 1	189
α^{110}	$\alpha^6 + \alpha^5 + \alpha^2 + \alpha + 1$	0 1 1 0 0 1 1 1	103
α^{111}	$\alpha^7 + \alpha^6 + \alpha^3 + \alpha^2 + \alpha$	1 1 0 0 1 1 1 0	206
α^{112}	$\alpha^7 + 1$	1 0 0 0 0 0 0 1	129
α^{113}	$\alpha^4 + \alpha^3 + \alpha^2 + \alpha + 1$	0 0 0 1 1 1 1 1	31
α^{114}	$\alpha^5 + \alpha^4 + \alpha^3 + \alpha^2 + \alpha$	0 0 1 1 1 1 1 0	62
α^{115}	$\alpha^6 + \alpha^5 + \alpha^4 + \alpha^3 + \alpha^2$	0 1 1 1 1 1 0 0	124
α^{116}	$\alpha^7 + \alpha^6 + \alpha^5 + \alpha^4 + \alpha^3$	1 1 1 1 1 0 0 0	248
α^{117}	$\alpha^7 + \alpha^6 + \alpha^5 + \alpha^3 + \alpha^2 + 1$	1 1 1 0 1 1 0 1	237
α^{118}	$\alpha^7 + \alpha^6 + \alpha^2 + \alpha + 1$	1 1 0 0 0 1 1 1	199
α^{119}	$\alpha^7 + \alpha^4 + \alpha + 1$	1 0 0 1 0 0 1 1	147
α^{120}	$\alpha^5 + \alpha^4 + \alpha^3 + \alpha + 1$	0 0 1 1 1 0 1 1	59
α^{121}	$\alpha^6 + \alpha^5 + \alpha^4 + \alpha^2 + \alpha$	0 1 1 1 0 1 1 0	118
α^{122}	$\alpha^7 + \alpha^6 + \alpha^5 + \alpha^3 + \alpha^2$	1 1 1 0 1 1 0 0	236
α^{123}	$\alpha^7 + \alpha^6 + \alpha^2 + 1$	1 1 0 0 0 1 0 1	197
α^{124}	$\alpha^7 + \alpha^4 + \alpha^2 + \alpha + 1$	1 0 0 1 0 1 1 1	151
α^{125}	$\alpha^5 + \alpha^4 + \alpha + 1$	0 0 1 1 0 0 1 1	51
α^{126}	$\alpha^6 + \alpha^5 + \alpha^2 + \alpha$	0 1 1 0 0 1 1 0	102
α^{127}	$\alpha^7 + \alpha^6 + \alpha^3 + \alpha^2$	1 1 0 0 1 1 0 0	204
α^{128}	$\alpha^7 + \alpha^2 + 1$	1 0 0 0 0 1 0 1	133
α^{129}	$\alpha^4 + \alpha^2 + \alpha + 1$	0 0 0 1 0 1 1 1	23
α^{130}	$\alpha^5 + \alpha^3 + \alpha^2 + \alpha$	0 0 1 0 1 1 1 0	46
α^{131}	$\alpha^6 + \alpha^4 + \alpha^3 + \alpha^2$	0 1 0 1 1 1 0 0	92
α^{132}	$\alpha^7 + \alpha^5 + \alpha^4 + \alpha^3$	1 0 1 1 1 0 0 0	184
α^{133}	$\alpha^6 + \alpha^5 + \alpha^3 + \alpha^2 + 1$	0 1 1 0 1 1 0 1	109
α^{134}	$\alpha^7 + \alpha^6 + \alpha^4 + \alpha^3 + \alpha$	1 1 0 1 1 0 1 0	218
α^{135}	$\alpha^7 + \alpha^5 + \alpha^3 + 1$	1 0 1 0 1 0 0 1	169
α^{136}	$\alpha^6 + \alpha^3 + \alpha^2 + \alpha + 1$	0 1 0 0 1 1 1 1	79
α^{137}	$\alpha^7 + \alpha^4 + \alpha^3 + \alpha^2 + \alpha$	1 0 0 1 1 1 1 0	158
α^{138}	$\alpha^5 + 1$	0 0 1 0 0 0 0 1	33
α^{139}	$\alpha^6 + \alpha$	0 1 0 0 0 0 1 0	66
α^{140}	$\alpha^7 + \alpha^2$	1 0 0 0 0 1 0 0	132
α^{141}	$\alpha^4 + \alpha^2 + 1$	0 0 0 1 0 1 0 1	21
α^{142}	$\alpha^5 + \alpha^3 + \alpha$	0 0 1 0 1 0 1 0	42
α^{143}	$\alpha^6 + \alpha^4 + \alpha^2$	0 1 0 1 0 1 0 0	84
α^{144}	$\alpha^7 + \alpha^5 + \alpha^3$	1 0 1 0 1 0 0 0	168

Tableau A.2: Corps de Galois $GF[q] = GF[256]$ (partie 4)

élément de $GF[256]$	polynôme	forme binaire	forme décimale
α^{145}	$\alpha^6 + \alpha^3 + \alpha^2 + 1$	0 1 0 0 1 1 0 1	77
α^{146}	$\alpha^7 + \alpha^4 + \alpha^3 + \alpha$	1 0 0 1 1 0 1 0	154
α^{147}	$\alpha^5 + \alpha^3 + 1$	0 0 1 0 1 0 0 1	41
α^{148}	$\alpha^6 + \alpha^4 + \alpha$	0 1 0 1 0 0 1 0	82
α^{149}	$\alpha^7 + \alpha^5 + \alpha^2$	1 0 1 0 0 1 0 0	164
α^{150}	$\alpha^6 + \alpha^4 + \alpha^2 + 1$	0 1 0 1 0 1 0 1	85
α^{151}	$\alpha^7 + \alpha^5 + \alpha^3 + \alpha$	1 0 1 0 1 0 1 0	170
α^{152}	$\alpha^6 + \alpha^3 + 1$	0 1 0 0 1 0 0 1	73
α^{153}	$\alpha^7 + \alpha^4 + \alpha$	1 0 0 1 0 0 1 0	146
α^{154}	$\alpha^5 + \alpha^4 + \alpha^3 + 1$	0 0 1 1 1 0 0 1	57
α^{155}	$\alpha^6 + \alpha^5 + \alpha^4 + \alpha$	0 1 1 1 0 0 1 0	114
α^{156}	$\alpha^7 + \alpha^6 + \alpha^5 + \alpha^2$	1 1 1 0 0 1 0 0	228
α^{157}	$\alpha^7 + \alpha^6 + \alpha^4 + \alpha^2 + 1$	1 1 0 1 0 1 0 1	213
α^{158}	$\alpha^7 + \alpha^5 + \alpha^4 + \alpha^2 + \alpha + 1$	1 0 1 1 0 1 1 1	183
α^{159}	$\alpha^6 + \alpha^5 + \alpha^4 + \alpha + 1$	0 1 1 1 0 0 1 1	115
α^{160}	$\alpha^7 + \alpha^6 + \alpha^5 + \alpha^2 + \alpha$	1 1 1 0 0 1 1 0	230
α^{161}	$\alpha^7 + \alpha^6 + \alpha^4 + 1$	1 1 0 1 0 0 0 1	209
α^{162}	$\alpha^7 + \alpha^5 + \alpha^4 + \alpha^3 + \alpha^2 + \alpha + 1$	1 0 1 1 1 1 1 1	191
α^{163}	$\alpha^6 + \alpha^5 + \alpha + 1$	0 1 1 0 0 0 1 1	99
α^{164}	$\alpha^7 + \alpha^6 + \alpha^2 + \alpha$	1 1 0 0 0 1 1 0	198
α^{165}	$\alpha^7 + \alpha^4 + 1$	1 0 0 1 0 0 0 1	145
α^{166}	$\alpha^5 + \alpha^4 + \alpha^3 + \alpha^2 + \alpha + 1$	0 0 1 1 1 1 1 1	63
α^{167}	$\alpha^6 + \alpha^5 + \alpha^4 + \alpha^3 + \alpha^2 + \alpha$	0 1 1 1 1 1 1 0	126
α^{168}	$\alpha^7 + \alpha^6 + \alpha^5 + \alpha^4 + \alpha^3 + \alpha^2$	1 1 1 1 1 1 0 0	252
α^{169}	$\alpha^7 + \alpha^6 + \alpha^5 + \alpha^2 + 1$	1 1 1 0 0 1 0 1	229
α^{170}	$\alpha^7 + \alpha^6 + \alpha^4 + \alpha^2 + \alpha + 1$	1 1 0 1 0 1 1 1	215
α^{171}	$\alpha^7 + \alpha^5 + \alpha^4 + \alpha + 1$	1 0 1 1 0 0 1 1	179
α^{172}	$\alpha^6 + \alpha^5 + \alpha^4 + \alpha^3 + \alpha + 1$	0 1 1 1 1 0 1 1	123
α^{173}	$\alpha^7 + \alpha^6 + \alpha^5 + \alpha^4 + \alpha^2 + \alpha$	1 1 1 1 0 1 1 0	246
α^{174}	$\alpha^7 + \alpha^6 + \alpha^5 + \alpha^4 + 1$	1 1 1 1 0 0 0 1	241
α^{175}	$\alpha^7 + \alpha^6 + \alpha^5 + \alpha^4 + \alpha^3 + \alpha^2 + \alpha + 1$	1 1 1 1 1 1 1 1	255
α^{176}	$\alpha^7 + \alpha^6 + \alpha^5 + \alpha + 1$	1 1 1 0 0 0 1 1	227
α^{177}	$\alpha^7 + \alpha^6 + \alpha^4 + \alpha^3 + \alpha + 1$	1 1 0 1 1 0 1 1	219
α^{178}	$\alpha^7 + \alpha^5 + \alpha^3 + \alpha + 1$	1 0 1 0 1 0 1 1	171
α^{179}	$\alpha^6 + \alpha^3 + \alpha + 1$	0 1 0 0 1 0 1 1	75
α^{180}	$\alpha^7 + \alpha^4 + \alpha^2 + \alpha$	1 0 0 1 0 1 1 0	150
α^{181}	$\alpha^5 + \alpha^4 + 1$	0 0 1 1 0 0 0 1	49
α^{182}	$\alpha^6 + \alpha^5 + \alpha$	0 1 1 0 0 0 1 0	98
α^{183}	$\alpha^7 + \alpha^6 + \alpha^2$	1 1 0 0 0 1 0 0	196
α^{184}	$\alpha^7 + \alpha^4 + \alpha^2 + 1$	1 0 0 1 0 1 0 1	149
α^{185}	$\alpha^5 + \alpha^4 + \alpha^2 + \alpha + 1$	0 0 1 1 0 1 1 1	55
α^{186}	$\alpha^6 + \alpha^5 + \alpha^3 + \alpha^2 + \alpha$	0 1 1 0 1 1 1 0	110
α^{187}	$\alpha^7 + \alpha^6 + \alpha^4 + \alpha^3 + \alpha^2$	1 1 0 1 1 1 0 0	220
α^{188}	$\alpha^7 + \alpha^5 + \alpha^2 + 1$	1 0 1 0 0 1 0 1	165
α^{189}	$\alpha^6 + \alpha^4 + \alpha^2 + \alpha + 1$	0 1 0 1 0 1 1 1	87
α^{190}	$\alpha^7 + \alpha^5 + \alpha^3 + \alpha^2 + \alpha$	1 0 1 0 1 1 1 0	174
α^{191}	$\alpha^6 + 1$	0 1 0 0 0 0 0 1	65
α^{192}	$\alpha^7 + \alpha$	1 0 0 0 0 0 1 0	130
α^{193}	$\alpha^4 + \alpha^3 + 1$	0 0 0 1 1 0 0 1	25
α^{194}	$\alpha^5 + \alpha^4 + \alpha$	0 0 1 1 0 0 1 0	50

Tableau A.2: Corps de Galois $GF[q] = GF[256]$ (partie 5)

élément de $GF[256]$	polynôme	forme binaire	forme décimale
α^{195}	$\alpha^6 + \alpha^5 + \alpha^2$	0 1 1 0 0 1 0 0	100
α^{196}	$\alpha^7 + \alpha^6 + \alpha^3$	1 1 0 0 1 0 0 0	200
α^{197}	$\alpha^7 + \alpha^3 + \alpha^2 + 1$	1 0 0 0 1 1 0 1	141
α^{198}	$\alpha^2 + \alpha + 1$	0 0 0 0 0 1 1 1	7
α^{199}	$\alpha^3 + \alpha^2 + \alpha$	0 0 0 0 1 1 1 0	14
α^{200}	$\alpha^4 + \alpha^3 + \alpha^2$	0 0 0 1 1 1 0 0	28
α^{201}	$\alpha^5 + \alpha^4 + \alpha^3$	0 0 1 1 1 0 0 0	56
α^{202}	$\alpha^6 + \alpha^5 + \alpha^4$	0 1 1 1 0 0 0 0	112
α^{203}	$\alpha^7 + \alpha^6 + \alpha^5$	1 1 1 0 0 0 0 0	224
α^{204}	$\alpha^7 + \alpha^6 + \alpha^4 + \alpha^3 + \alpha^2 + 1$	1 1 0 1 1 1 0 1	221
α^{205}	$\alpha^7 + \alpha^5 + \alpha^2 + \alpha + 1$	1 0 1 0 0 1 1 1	167
α^{206}	$\alpha^6 + \alpha^4 + \alpha + 1$	0 1 0 1 0 0 1 1	83
α^{207}	$\alpha^7 + \alpha^5 + \alpha^2 + \alpha$	1 0 1 0 0 1 1 0	166
α^{208}	$\alpha^6 + \alpha^4 + 1$	0 1 0 1 0 0 0 1	81
α^{209}	$\alpha^7 + \alpha^5 + \alpha$	1 0 1 0 0 0 1 0	162
α^{210}	$\alpha^6 + \alpha^4 + \alpha^3 + 1$	0 1 0 1 1 0 0 1	89
α^{211}	$\alpha^7 + \alpha^5 + \alpha^4 + \alpha$	1 0 1 1 0 0 1 0	178
α^{212}	$\alpha^6 + \alpha^5 + \alpha^4 + \alpha^3 + 1$	0 1 1 1 1 0 0 1	121
α^{213}	$\alpha^7 + \alpha^6 + \alpha^5 + \alpha^4 + \alpha$	1 1 1 1 0 0 1 0	242
α^{214}	$\alpha^7 + \alpha^6 + \alpha^5 + \alpha^4 + \alpha^3 + 1$	1 1 1 1 1 0 0 1	249
α^{215}	$\alpha^7 + \alpha^6 + \alpha^5 + \alpha^3 + \alpha^2 + \alpha + 1$	1 1 1 0 1 1 1 1	239
α^{216}	$\alpha^7 + \alpha^6 + \alpha + 1$	1 1 0 0 0 0 1 1	195
α^{217}	$\alpha^7 + \alpha^4 + \alpha^3 + \alpha + 1$	1 0 0 1 1 0 1 1	155
α^{218}	$\alpha^5 + \alpha^3 + \alpha + 1$	0 0 1 0 1 0 1 1	43
α^{219}	$\alpha^6 + \alpha^4 + \alpha^2 + \alpha$	0 1 0 1 0 1 1 0	86
α^{220}	$\alpha^7 + \alpha^5 + \alpha^3 + \alpha^2$	1 0 1 0 1 1 0 0	172
α^{221}	$\alpha^6 + \alpha^2 + 1$	0 1 0 0 0 1 0 1	69
α^{222}	$\alpha^7 + \alpha^3 + \alpha$	1 0 0 0 1 0 1 0	138
α^{223}	$\alpha^3 + 1$	0 0 0 0 1 0 0 1	9
α^{224}	$\alpha^4 + \alpha$	0 0 0 1 0 0 1 0	18
α^{225}	$\alpha^5 + \alpha^2$	0 0 1 0 0 1 0 0	36
α^{226}	$\alpha^6 + \alpha^3$	0 1 0 0 1 0 0 0	72
α^{227}	$\alpha^7 + \alpha^4$	1 0 0 1 0 0 0 0	144
α^{228}	$\alpha^5 + \alpha^4 + \alpha^3 + \alpha^2 + 1$	0 0 1 1 1 1 0 1	61
α^{229}	$\alpha^6 + \alpha^5 + \alpha^4 + \alpha^3 + \alpha$	0 1 1 1 1 0 1 0	122
α^{230}	$\alpha^7 + \alpha^6 + \alpha^5 + \alpha^4 + \alpha^2$	1 1 1 1 0 1 0 0	244
α^{231}	$\alpha^7 + \alpha^6 + \alpha^5 + \alpha^4 + \alpha^2 + 1$	1 1 1 1 0 1 0 1	245
α^{232}	$\alpha^7 + \alpha^6 + \alpha^5 + \alpha^4 + \alpha^2 + \alpha + 1$	1 1 1 1 0 1 1 1	247
α^{233}	$\alpha^7 + \alpha^6 + \alpha^5 + \alpha^4 + \alpha + 1$	1 1 1 1 0 0 1 1	243
α^{234}	$\alpha^7 + \alpha^6 + \alpha^5 + \alpha^4 + \alpha^3 + \alpha + 1$	1 1 1 1 1 0 1 1	251
α^{235}	$\alpha^7 + \alpha^6 + \alpha^5 + \alpha^3 + \alpha + 1$	1 1 1 0 1 0 1 1	235
α^{236}	$\alpha^7 + \alpha^6 + \alpha^3 + \alpha + 1$	1 1 0 0 1 0 1 1	203
α^{237}	$\alpha^7 + \alpha^3 + \alpha + 1$	1 0 0 0 1 0 1 1	139
α^{238}	$\alpha^3 + \alpha + 1$	0 0 0 0 1 0 1 1	11
α^{239}	$\alpha^4 + \alpha^2 + \alpha$	0 0 0 1 0 1 1 0	22
α^{240}	$\alpha^5 + \alpha^3 + \alpha^2$	0 0 1 0 1 1 0 0	44
α^{241}	$\alpha^6 + \alpha^4 + \alpha^3$	0 1 0 1 1 0 0 0	88
α^{242}	$\alpha^7 + \alpha^5 + \alpha^4$	1 0 1 1 0 0 0 0	176
α^{243}	$\alpha^6 + \alpha^5 + \alpha^4 + \alpha^3 + \alpha^2 + 1$	0 1 1 1 1 1 0 1	125
α^{244}	$\alpha^7 + \alpha^6 + \alpha^5 + \alpha^4 + \alpha^3 + \alpha$	1 1 1 1 1 0 1 0	250

Tableau A.2: Corps de Galois $GF[q] = GF[256]$ (partie 6)

élément de $GF[256]$	polynôme								forme binaire								forme décimale
α^{245}	α^7	$+\alpha^6$	$+\alpha^5$		$+\alpha^3$			$+1$	1	1	1	0	1	0	0	1	233
α^{246}	α^7	$+\alpha^6$			$+\alpha^3$	$+\alpha^2$	$+\alpha$	$+1$	1	1	0	0	1	1	1	1	207
α^{247}	α^7						$+\alpha$	$+1$	1	0	0	0	0	0	1	1	131
α^{248}				α^4	$+\alpha^3$		$+\alpha$	$+1$	0	0	0	1	1	0	1	1	27
α^{249}			α^5	$+\alpha^4$		$+\alpha^2$	$+\alpha$		0	0	1	1	0	1	1	0	54
α^{250}		α^6	$+\alpha^5$		$+\alpha^3$	$+\alpha^2$			0	1	1	0	1	1	0	0	108
α^{251}	α^7	$+\alpha^6$		$+\alpha^4$	$+\alpha^3$				1	1	0	1	1	0	0	0	216
α^{252}	α^7		$+\alpha^5$		$+\alpha^3$	$+\alpha^2$		$+1$	1	0	1	0	1	1	0	1	173
α^{253}		α^6				$+\alpha^2$	$+\alpha$	$+1$	0	1	0	0	0	1	1	1	71
α^{254}	α^7				$+\alpha^3$	$+\alpha^2$	$+\alpha$		1	0	0	0	1	1	1	0	142

Bibliography

[BDMS91] E. Biglieri, D. Divsalar, P.J. McLane, and M.K. Simon. *Introduction to Trellis-Coded Modulation with Applications.* Macmillan Publishing Company, New-York, 1991.

[Bla84] R.E. Blahut. *Theory and Practice of Error Control Codes.* Addison-Wesley, Reading, Mass., 1984.

[CDG92] G. Cohen, J.-L. Dornstetter, and P. Godlewski. *Codes correcteurs d'erreurs: une introduction au codage algébrique.* Masson, Paris, 1992. Collection Technique et Scientifique des Télécommunications *CNET-ENST.*

[CETM95] A. Chini, M. El-Tanany, and S. Mahmoud. Application of a Filtered Decision Feedback Channel Estimator in Multi Carrier Digital TV Broadcasting. Technical Report 67CRC-4-0577, Carleton University, Ottawa, Ontario, Canada, mars 1995.

[CT95] J.-Y. Chouinard and M. Trichard. Codes de correction d'erreurs pour la transmission de la télévision numérique - phase 1. Technical Report 67CRC-4-0423, Université d'Ottawa, Ottawa, Ontario, Canada, mars 1995.

[For70] G.D. Forney. Convolutional Codes I: Algebraic Structure. *IEEE Transactions on Information Theory*, IT-16(6):720–738, novembre 1970.

[Hay94] Simon Haykin. *Communication Systems.* John Wiley and Sons, New-York, second edition, 1994.

[HLP93] C. Heegard, S.A. Lery, and W.H. Paik. Practical Coding for QAM Transmission of HDTV. *IEEE Journal on Selected Areas in Communications*, JSAC-11(1):111–118, janvier 1993.

[JBS92] Michel C. Jeruchim, Philip Balaban, and K. Sam Shanmugan. *Simulation of Communication Systems.* Plenum Press, New-York, 1992.

[KSL94] Thomas P. Krauss, Loren Shure, and John Little. *Signal Processing Toolbox for Use with Matlab.* The MathWorks, Inc., Natick, Massachusetts, 1994.

[LC83] S. Lin and D. Costello. *Error Control Coding: Fundamentals and Applications.* Prentice-Hall, New-Jersey, 1983.

[Lee77] L.-N. Lee. Concatenated Coding Systems Employing a Unit-Memory Convolutional Code and a Byte-Oriented Decoding Algorithm. *IEEE Transactions on Communications*, COM-25(10):1064–1074, octobre 1977.

[Mat92] MathWorks. *Simulink A Program for Simulating Dynamic Systems.* The MathWorks, Inc., Natick, Massachusetts, 1992.

[Mat94] MathWorks. *DSP Blockset for Use with Simulink.* The MathWorks, Inc., Natick, Massachusetts, 1994.

[MMV95] V. Mignone, A. Morello, and M. Visintin. CD3-OFDM: A New Channel Estimation Method to Improve the Spectrum Efficiency in Digital Terrestrial Television Systems. In *International Broadcasting Convention (IBC'95)*, number 413 in IEE Conference Publication, pages 122–128, Amsterdam, 14 au 18 septembre 1995. Institution of Electrical Engineers (IEE).

[Ple89] V. Pless. *Introduction to the Theory of Error-Correcting Codes.* Wiley-Interscience Series, New-York, 1989.

[RHG93] S.A. Raghavan, Y. Hebron, and I. Gurantz. On the Application of Appropriate APP Decoding to Digital Video Transmission. *IEEE Journal on Selected Areas in Communications*, JSAC-11(1):136–145, janvier 1993.

[Rze95] T.S. Rzeszewski. *Digital Video: Concepts and Applications Across Industries.* IEEE Press, Piscataway, 1995.

[Tri95] M. Trichard. Study of Trellis Coded Modulation and Error Control Coding. Master's thesis, Université d'Ottawa, Ottawa, Ontario, Canada, octobre 1995.

[Ung82] G. Ungerboeck. Channel Coding with Multilevel/Phase Signals. *IEEE Transactions on Information Theory*, IT-28(1):55–67, janvier 1982.

[Ung87a] G. Ungerboeck. Trellis-Coded Modulation with Redundant Signal Sets; Part I: Introduction. *IEEE Communications Magazine*, 25(2):5–11, février 1987.

[Ung87b] G. Ungerboeck. Trellis-Coded Modulation with Redundant Signal Sets; Part II: State of the Art. *IEEE Communications Magazine*, 25(2):12–21, février 1987.

[WB94] S.B. Wicker and V.K. Bhargava. *Reed-Solomon Codes and Their Applications*, chapter 11, pages 242–271. IEEE Press, Piscataway, 1994. Chapitre rédigé par J. Hagenauer, E. Offer et L. Papke et intitulé *Matching Viterbi Decoders and Reed-Solomon Decoders in a Concatenated System.*

[WC94] Y. Wu and B. Caron. Digital Television Terrestrial Broadcasting. *IEEE Communications Magazine*, 32(5):46–52, mai 1994.

[WGCT95] Y. Wu, M. Guillet, J.-Y. Chouinard, and M. Trichard. COFDM for Digital ATV Terrestrial Distribution over 6MHz Channels. In *International Broadcasting Convention (IBC'95)*, number 413 in IEE Conference Publication, pages 29–34, Amsterdam, 14 au 18 septembre 1995. Institution of Electrical Engineers (IEE).

[WLC94] Y. Wu, B. Ledoux, and B. Caron. Evaluation of Multi-Carrier Transmission Systems for Advanced Television in North America. In *NAB 1994 Broadcast Engineer Conference Proceedings*, mars 1994.