



Faculté de génie  
Département de génie électrique et génie informatique

ALGORITHME DE CONTRÔLE D'ERREURS  
APPLIQUÉ AU PHÉNOMÈNE DE DISPERSION  
CHROMATIQUE

Mémoire de maîtrise en sciences appliquées  
Spécialité : génie électrique

Manon BÉLANGER

---

Sherbrooke (Québec), Canada

Janvier 2008

IV-1859



Library and  
Archives Canada

Bibliothèque et  
Archives Canada

Published Heritage  
Branch

Direction du  
Patrimoine de l'édition

395 Wellington Street  
Ottawa ON K1A 0N4  
Canada

395, rue Wellington  
Ottawa ON K1A 0N4  
Canada

*Your file* *Votre référence*  
*ISBN: 978-0-494-42929-7*  
*Our file* *Notre référence*  
*ISBN: 978-0-494-42929-7*

**NOTICE:**

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

**AVIS:**

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

---

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

  
**Canada**

**UNIVERSITÉ DE SHERBROOKE**

**RÉSUMÉ**

**ALGORITHME DE CONTRÔLE D'ERREURS  
APPLIQUÉ AU PHÉNOMÈNE DE DISPERSION  
CHROMATIQUE**

par Manon Bélanger

Directeur de thèse : Professeur Alain Houle

Département de génie électrique et génie informatique

Un algorithme de contrôle d'erreurs, utilisant les statistiques des valeurs à la sortie d'un canal de communication par fibre optique pour minimiser les erreurs induites par la dispersion chromatique, est présenté. Utilisé conjointement avec un code correcteur d'erreurs, l'algorithme permet d'améliorer considérablement les performances du décodeur. En obtenant la distribution de la puissance moyenne à la sortie du canal en fonction de la séquence de bits à l'entrée, il est possible de calculer la probabilité d'appartenance de chaque bit à chacune des séquences possibles et ainsi déterminer quel bit est le plus probable d'avoir été émis. Cet algorithme produit de très bons résultats lorsque la dispersion chromatique accumulée est élevée. En cela, il est complémentaire aux codes correcteurs d'erreurs, car ces derniers sont efficaces uniquement si le taux d'erreur initial est en deçà d'un certain seuil, seuil qui est généralement excédé lorsque la dispersion chromatique accumulée est élevée.

## REMERCIEMENTS

L'auteure voudrait avant tout remercier son directeur de recherche, Alain Houle, pour ses précieux conseils, ainsi que le CRSNG pour le support financier. L'Office québécois de la langue française mérite également une mention, car son Grand dictionnaire terminologique m'a été très utile pour traduire les termes techniques de l'anglais au français. Merci également aux autres membres de l'équipe de recherche d'Alain Houle et à tous les autres étudiants qui ont permis de rendre mon séjour à Sherbrooke plus agréable. Finalement, j'adresse un merci spécial à toute ma famille et à Yannick Solari, pour leur support moral inconditionnel et constant.

# TABLE DES MATIÈRES

<b>1. INTRODUCTION .....</b>	<b>1</b>
1.1 Mise en situation .....	1
1.2 Explication de la dispersion chromatique .....	2
1.2.1 Causes physiques .....	2
1.2.2 Impact sur le signal .....	3
1.2.3 Variation du taux de dispersion chromatique .....	4
1.2.4 Autres caractéristiques de la fibre .....	5
1.2.5 Piaillage .....	8
1.3 Solution 1 : Pré-compensation .....	9
1.3.1 Pré-piaillage .....	10
1.3.2 Format de modulation .....	10
1.4 Solution 2 : Compensation optique .....	11
1.4.1 Fibre DCF .....	11
1.4.2 Réseau de Bragg .....	12
1.5 Solution 3 : Égalisation .....	13
1.5.1 Seuil de décision adaptatif .....	14
1.5.2 <i>Feed-Forward Equalizer (FFE)</i> .....	15
1.5.3 <i>Decision-Feedback Equalizer (DFE)</i> .....	16
1.5.4 <i>Maximum Likelihood Sequence Estimation (MLSE)</i> .....	17
1.6 Solution 4 : Correction d'erreurs .....	18
1.6.1 Codes correcteurs d'erreurs .....	18
1.6.2 Turbo codes .....	20
1.7 Bilan .....	22
1.8 Projet de recherche .....	23
1.8.1 Intérêt d'une nouvelle enquête .....	23
1.8.2 Contributions .....	24
1.8.3 Organisation du mémoire .....	24
<b>2. MODELISATION DU CANAL .....</b>	<b>25</b>
2.1 Construction du signal .....	25
2.1.1 Format de signalisation .....	25
2.1.2 Modulation .....	27
2.1.3 Problème pratique : la porteuse .....	28
2.2 Modélisation du canal .....	29
2.2.1 Idée générale .....	29
2.2.2 Numérisation .....	30
2.2.3 Théorème d'échantillonnage .....	31
2.2.4 Équation de Sellmeier .....	33
2.3 Résultats de simulation .....	35
2.4 Mesure de l'interférence inter-symbole .....	40
2.4.1 Proposition .....	40
2.4.2 Méthode de calcul .....	41
2.4.3 Résultats .....	42

2.5	Temps de simulation .....	47
2.5.1	Problématique du temps de simulation.....	47
2.5.2	Réduction du temps de simulation.....	47
2.5.3	Méthode alternative .....	47
2.6	Résumé .....	50
<b>3.</b>	<b>CALCUL DU TAUX D'ERREUR.....</b>	<b>52</b>
3.1	Prise de décision .....	52
3.1.1	Intégration au récepteur.....	52
3.1.2	Seuil de décision.....	53
3.2	Calcul du taux d'erreur.....	55
3.2.1	Méthode de Monte Carlo .....	56
3.2.2	Optimisation du seuil de décision .....	57
3.2.3	Variabilité du résultat .....	59
3.2.4	Impact du nombre d'échantillons par bit.....	61
3.2.5	Taux d'erreur minimal en fonction de la distance .....	62
3.2.6	Limite de propagation .....	63
3.2.7	Dispersion chromatique accumulée.....	64
3.3	Ajout du bruit .....	65
3.3.1	Sources de bruit.....	65
3.4	Simulation des résultats.....	66
3.4.1	Courbes à obtenir .....	66
3.4.2	Résultats.....	68
3.5	Résumé .....	70
<b>4.</b>	<b>DÉCODEUR INITIAL .....</b>	<b>71</b>
4.1	Encodage .....	71
4.1.1	Choix du code.....	72
4.1.2	Impact du surdébit .....	73
4.1.3	Caractéristiques des codes BCH.....	74
4.1.4	BCH(256,239).....	75
4.1.5	BCH(144,128).....	75
4.2	Exemple de décodage d'un turbo code .....	75
4.3	Algorithme de Chase .....	78
4.3.1	Définitions préalables .....	78
4.3.2	Algorithme du décodeur.....	79
4.3.3	Génération des patrons de test .....	81
4.4	Calcul de la fiabilité.....	82
4.5	Résultats de simulation .....	85
4.5.1	Méthode de génération des patrons de test à privilégier .....	85
4.5.2	Nombre d'itérations nécessaire .....	87
4.5.3	Ordre du décodage.....	87
4.5.4	Résultats.....	90
4.6	Résumé .....	91

<b>5. ALGORITHME DE CONTRÔLE D'ERREURS .....</b>	<b>93</b>
5.1 Sortie du canal en fonction de la séquence de bits .....	93
5.2 Explication de l'algorithme de prétraitement .....	95
5.3 Résultats de simulation .....	98
5.4 Performances suite à un changement de condition .....	101
5.4.1 Impact d'une variation du rapport signal sur bruit.....	101
5.4.2 Impact d'une variation de la dispersion chromatique accumulée.....	103
5.5 Aperçu d'ensemble .....	105
5.6 Résumé.....	106
<b>CONCLUSION.....</b>	<b>107</b>
<b>ANNEXE I.....</b>	<b>110</b>
<b>ANNEXE II.....</b>	<b>111</b>
<b>ANNEXE III.....</b>	<b>113</b>
<b>ANNEXE IV.....</b>	<b>115</b>
<b>ANNEXE V.....</b>	<b>120</b>
<b>BIBLIOGRAPHIE .....</b>	<b>123</b>

## LISTE DES FIGURES

Figure 1.1 – Diagramme d’œil a) sain b) avec dispersion chromatique .....	3
Figure 1.2 – Taux de dispersion chromatique .....	5
Figure 1.3 – Signal ayant subi du piaillement [ISSA 2007].....	8
Figure 1.4 – Principe du pré-piaillement.....	10
Figure 1.5 – Réseau de Bragg à pas variable .....	13
Figure 1.6 – Système avec seuil de décision adaptatif .....	14
Figure 1.7 – Performances d’un égalisateur FFE .....	15
Figure 1.8 – Schéma d’un égalisateur FFE + DFE.....	16
Figure 1.9 – Performances d’un égalisateur FFE + DFE.....	17
Figure 1.10 – Système de compensation MLSE .....	17
Figure 1.11 – Exemple d’un code en bloc BCH(7,4) .....	18
Figure 1.12 – Exemple d’encodeur convolutif.....	19
Figure 1.13 – Détermination du gain de codage .....	20
Figure 1.14 – Schéma bloc général d’un encodeur turbo code.....	21
Figure 1.15 – Schéma bloc général d’un décodeur turbo code.....	21
Figure 2.1 – Approximation d’un système d’ordre 1 .....	26
Figure 2.2 – Allure temporelle d’un signal binaire formaté en NRZ.....	27
Figure 2.3 – Allure temporelle d’un signal modulé AM .....	28
Figure 2.4 – Représentation dans le domaine fréquentiel de l’impact de la modulation AM..	29
Figure 2.5 – Schéma du système de transmission.....	30
Figure 2.6 – Schéma du système de transmission pour les simulations.....	31
Figure 2.7 – Signal pour le calcul de la transformée de Fourier .....	31
Figure 2.8 – Contenu fréquentiel d’un signal.....	32
Figure 2.9 – Diagrammes d’œil obtenus avec l’algorithme présenté .....	39
Figure 2.10 – Signal proposé pour la mesure de l’interférence inter-symbole .....	41
Figure 2.11 – Modification de la période du bit.....	43
Figure 2.12 – Confirmation du centrage de la période d’un bit.....	43
Figure 2.13 – Ajustement des paramètres de simulation .....	44
Figure 2.14 – Impact du nombre de points par bit.....	45
Figure 2.15 – Pourcentage d’énergie dans les bits adjacents en fonction de la distance .....	45
Figure 2.16 – Optimisation de la longueur du filtre pour simuler la dispersion chromatique ..	49
Figure 2.17 – Optimisation de la longueur du filtre d’ordre 1 au récepteur.....	50
Figure 3.1 – Visualisation de la période de décision .....	52
Figure 3.2 – Distribution des valeurs à la sortie du canal .....	54
Figure 3.3 – Test pour le seuil de décision.....	55
Figure 3.4 – Exemple d’un système simulé.....	56
Figure 3.5 – Taux d’erreur en fonction du seuil après 70 km de fibre .....	57
Figure 3.6 – Nombre de bits en erreur en fonction du seuil de décision .....	58
Figure 3.7 – Distribution des valeurs à la sortie du canal après 70 km de fibre .....	59
Figure 3.8 – Variabilité des résultats, séquences de 10 000 bits .....	60
Figure 3.9 – Seuil de décision optimal en fonction de la distance.....	62
Figure 3.10 – Taux d’erreur minimal en fonction de la distance .....	63
Figure 3.11 – Puissance du signal au récepteur .....	67
Figure 3.12 – BER en fonction du OSNR suivant la dispersion chromatique accumulée .....	69

Figure 3.13 – BER en fonction de la dispersion chromatique accumulée.....	69
Figure 4.1 – Structure du code proposé [MIZUOCHI et coll., 2004] .....	72
Figure 4.2 – Structure d'une trame G.709 [BARLOW 2003] .....	73
Figure 4.3 – Code BCH(7,4) x BCH(7,4) à transmettre .....	76
Figure 4.4 – Code BCH(7,4) x BCH(7,4) au récepteur .....	76
Figure 4.5 – Code BCH(7,4) x BCH(7,4) après une itération sur les lignes .....	77
Figure 4.6 – Code BCH(7,4) x BCH(7,4) après une itération sur les colonnes .....	77
Figure 4.7 – Schéma du système de communication .....	78
Figure 4.8 – Algorithme du décodeur .....	80
Figure 4.9 – Étage d'un décodeur à décision pondérée.....	84
Figure 4.10 – Entrelacement des colonnes.....	88
Figure 4.11 – Entrelacement complet .....	89
Figure 4.12 – Performances du décodeur.....	90
Figure 5.1 – Distribution statistique des '1' et des '0' – 1 bit avant et après .....	94
Figure 5.2 – Distribution statistique des '1' et des '0' - 2 bits avant et après .....	95
Figure 5.3 – Courbe d'appartenance à la séquence $k$ .....	96
Figure 5.4 – BER en fonction du OSNR, algorithme de prétraitement employé seul .....	99
Figure 5.5 – BER en fonction de la distance, algorithme de prétraitement employé seul .....	99
Figure 5.6 – Comparaison des performances.....	100
Figure 5.7 – BER lorsque l'OSNR simulé diffère de l'OSNR estimé.....	101
Figure 5.8 – Impact s'une surestimation de l'OSNR .....	102
Figure 5.9 – Impact de l'estimation de la dispersion chromatique accumulée .....	104
Figure 5.10 – Schéma du système .....	105
Figure AII.1 – Signal pour le calcul de la transformée de Fourier .....	111
Figure AII.2 – Signal triangulaire .....	111
Figure AII.3 – Reconstitution du signal à partir du signal triangulaire .....	112
Figure AIV.1 – Différence entre la simulation de 256 bits et deux séquences de 128 bits ...	116
Figure AIV.2 – Effet de l'ajout de bits sur l'erreur aux extrémités .....	117
Figure AIV.3 – Maximum sur 3 essais pour l'optimisation de la combinaison.....	118
Figure AIV.4 – Mise en évidence des plus courts temps de simulation .....	119

## LISTE DES TABLEAUX

Tableau 4.1 – Comparaison des méthodes de génération des patrons de test .....	86
Tableau 4.2 – Résultats en fonction de l'ordre du décodage .....	87
Tableau 4.3 – Résultats en utilisant différents types d'entrelacement.....	89
Tableau AIV.1 – Liste des facteurs de 36 864.....	115

## LISTE DES ACRONYMES

<b>AM</b>	:	<b>A</b> mplitude <b>M</b> odulation
<b>ARQ</b>	:	<b>A</b> utomatic <b>R</b> epeat <b>R</b> equest
<b>ASE</b>	:	<b>A</b> mplified <b>S</b> pontaneous <b>E</b> mission
<b>BCH</b>	:	<b>B</b> ose- <b>C</b> haudhuri- <b>H</b> ocquenghem
<b>BER</b>	:	<b>B</b> it <b>E</b> rror <b>R</b> ate
<b>CG</b>	:	<b>C</b> oding <b>G</b> ain
<b>DB</b>	:	<b>D</b> uobinary
<b>DCF</b>	:	<b>D</b> ispersion <b>C</b> ompensating <b>F</b> iber
<b>DFE</b>	:	<b>D</b> ecision- <b>F</b> eedback <b>E</b> qualizer
<b>DPSK</b>	:	<b>D</b> ifferential <b>P</b> hase- <b>S</b> hift <b>K</b> eying
<b>EDFA</b>	:	<b>E</b> rbium <b>D</b> oped <b>F</b> iber <b>A</b> mplifier
<b>FEC</b>	:	<b>F</b> orward <b>E</b> rror <b>C</b> orrection
<b>FFE</b>	:	<b>F</b> eed- <b>F</b> orward <b>E</b> qualizer
<b>FOM</b>	:	<b>F</b> igure of <b>M</b> erit
<b>ITU</b>	:	<b>I</b> nternational <b>T</b> elecommunication <b>U</b> nion
<b>ITU-T</b>	:	<b>I</b> TU <b>T</b> elecommunication <b>S</b> tandardization <b>S</b> ector
<b>MLSE</b>	:	<b>M</b> aximum <b>L</b> ikelihood <b>S</b> equence <b>E</b> stimation
<b>NCG</b>	:	<b>N</b> et <b>C</b> oding <b>G</b> ain
<b>NRZ</b>	:	<b>N</b> on <b>R</b> eturn to <b>Z</b> ero
<b>OOK</b>	:	<b>O</b> n- <b>O</b> ff <b>K</b> eying
<b>OSNR</b>	:	<b>O</b> ptical <b>S</b> ignal-to- <b>N</b> oise <b>R</b> atio
<b>OTN</b>	:	<b>O</b> ptical <b>T</b> ransport <b>N</b> etwork
<b>PMD</b>	:	<b>P</b> olarization <b>M</b> ode <b>D</b> ispersion
<b>RS</b>	:	<b>R</b> eed- <b>S</b> olomon
<b>RZ</b>	:	<b>R</b> eturn to <b>Z</b> ero
<b>SISO</b>	:	<b>S</b> oft <b>I</b> nput – <b>S</b> oft <b>O</b> utput
<b>WDM</b>	:	<b>W</b> avelength <b>D</b> ivision <b>M</b> ultiplexing

# CHAPITRE 1

## INTRODUCTION

### 1.1 Mise en situation

Depuis quelques années déjà, nous entrons de plein pied dans la société de l'information. Libéralisation des télécommunications, développement spectaculaire de l'Internet, mise en réseau des entreprises et de la société : l'information et l'accès à celle-ci sont devenus des produits à vendre, des moteurs de l'économie. Pour transmettre toutes ces informations, il faut nécessairement de plus en plus de bande passante, et cette bande passante doit bien sûr être de haute qualité (peu d'erreurs de transmission), fiable (peu de pannes) et disponible à faible coût.

Les réseaux de communication par fibre optique semblent les candidats idéaux pour satisfaire ces exigences, notamment à cause de leur grande capacité de transmission qui continue d'évoluer. Désireux d'augmenter la capacité de leurs réseaux, plusieurs opérateurs remplacent actuellement leurs équipements à 2,5 Gb/s par des équipements à 10 Gb/s, et à 40 Gb/s dans un futur rapproché. Malheureusement, plusieurs phénomènes physiques compliquent cette migration, le principal étant la dispersion chromatique. À 2,5 Gb/s, dans des conditions de modulation idéales, la dispersion chromatique devient problématique après 1000 km de fibre, tandis qu'à 10 Gb/s, cette distance descend à 80 km seulement et elle dégringole à 5 km à 40 Gb/s [HOULE 2006]. Pour les réseaux longue distance (entre  $\approx 2\ 000$  km et  $\approx 10\ 000$  km par lien), la dispersion chromatique était déjà un problème à 2,5 Gb/s, alors les équipements nécessaires pour la compensation sont déjà intégrés dans les réseaux. Toutefois, pour les réseaux métropolitains et régionaux (entre  $\approx 20$  km et  $\approx 600$  km par lien), le problème de la dispersion chromatique apparaît seulement à partir de 10 Gb/s.

Les solutions traditionnelles pour la compensation de la dispersion chromatique agissent au niveau optique. Elles nécessitent donc de modifier l'ingénierie de la couche physique du réseau, ce qui engendre des coûts substantiels. Pour éviter ces coûts, il faudrait corriger électroniquement les erreurs et les problèmes de transmission, comme c'est le cas actuellement dans la plupart des systèmes de télécommunications modernes. Quelques

solutions électroniques commencent à apparaître sur le marché, mais plusieurs recherches sont en cours pour améliorer les algorithmes utilisés ou en inventer de nouveau.

Toutes ces considérations mènent à l'objectif du présent projet : la conception d'un compensateur de dispersion électronique performant et à faible coût pour un canal à 10Gb/s dans le contexte d'un réseau métropolitain. Mais avant d'élaborer sur cet objectif, il est préférable de mieux comprendre le phénomène de la dispersion chromatique et d'explorer les solutions déjà existantes à ce problème.

## **1.2 Explication de la dispersion chromatique**

À 10 Gb/s, outre l'atténuation linéique, la principale limite quant à la distance maximale de transmission provient de la dispersion chromatique. Pourquoi ? Parce que les caractéristiques de la fibre font en sorte que la vitesse de propagation de la lumière dépend de sa longueur d'onde. Or, un signal modulé par une information n'a pas une largeur spectrale nulle. Pour un signal à 10 Gb/s, la largeur du spectre est typiquement d'environ 25 GHz. Cela signifie que l'information n'est pas transmise sur une seule longueur d'onde, mais bien sur un continuum de longueurs d'onde différentes. Ces longueurs d'onde voyageant à des vitesses différentes, elles ne prennent pas le même temps pour parcourir la même distance. Cela cause donc un élargissement des impulsions et éventuellement de l'interférence inter-symbole.

Quelles sont les causes physiques de ce phénomène ? Comment mesurer et visualiser l'impact sur le signal ? Existe-t-il d'autres phénomènes, d'autres caractéristiques propres à la fibre optique qui peuvent également poser problème ? Les sections qui suivent tenteront de répondre à ces questions.

### **1.2.1 Causes physiques**

Dans la fibre optique, il existe deux causes à la dispersion chromatique [PUREUR et coll., 2004]. D'une part, l'indice de réfraction de la silice, le matériau dont on se sert pour fabriquer la fibre optique, varie en fonction de la longueur d'onde. Or, l'indice de réfraction détermine la vitesse de propagation de l'onde dans le matériau. Par conséquent, à cause de la silice, la vitesse de propagation dans la fibre optique dépend de la longueur d'onde. C'est ce qu'on

appelle la dispersion du matériau. D'autre part, l'indice de réfraction effectif du mode de polarisation fondamental dépend des dimensions du guide par rapport à la longueur d'onde. La forme de la fibre optique fait donc elle aussi en sorte que la vitesse de propagation dépend de la longueur d'onde. C'est ce qu'on appelle la dispersion de forme. Le terme dispersion chromatique représente donc le résultat de la dispersion du matériau et de la dispersion de forme.

### 1.2.2 Impact sur le signal

Une façon pratique de visualiser l'effet de la dispersion chromatique est le diagramme d'œil. Celui-ci est une superposition des signaux reçus sur une période d'un bit. La figure 1.1, tirée de [WINZER et coll., 2005], présente deux diagrammes d'œil, l'un sans et l'autre avec dispersion chromatique. On remarque que le diagramme d'œil avec dispersion chromatique est plus fermé et qu'il est donc plus difficile de distinguer les '1' des '0'.

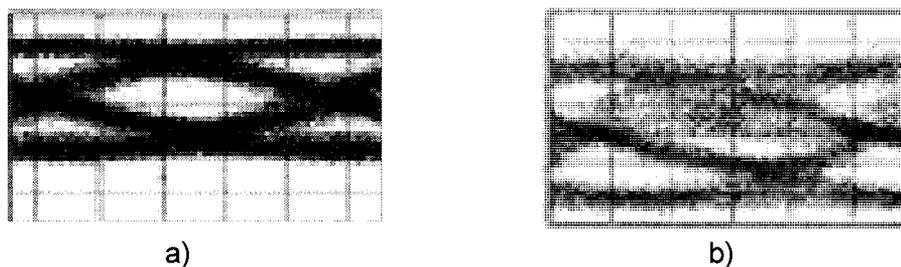


Figure 1.1 – Diagramme d'œil a) sain b) avec dispersion chromatique

Une métrique importante à connaître est le taux de dispersion chromatique qui caractérise la fibre transportant la lumière. Exprimée en  $\text{ps}/(\text{nm}\cdot\text{km})$ , cette métrique indique le retard de propagation entre deux ondes selon la différence entre leur longueur d'onde et selon la distance parcourue. Cela permet d'estimer l'élargissement temporel d'un signal en connaissant sa largeur de bande et la longueur totale du lien. Pour limiter le taux d'erreur, il ne faut pas que cet élargissement temporel dépasse une certaine fraction de la période d'un bit, sinon l'impulsion d'un bit déborde sur le bit suivant et il y a interférence.

Dans un système de communication par fibre optique sans compensation de la dispersion chromatique, il est possible, jusqu'à une certaine limite, d'augmenter la puissance du signal pour contrer l'impact de la dispersion chromatique. Cela correspond à une pénalité de

puissance dans le budget de puissance du lien de communication concerné, puisqu'il faut réserver dans ce but une partie de la puissance disponible, puissance qui pourrait plutôt, par exemple, permettre d'augmenter la longueur du lien ou de réduire le besoin d'amplification. Pour déterminer la pénalité de puissance causée par la dispersion chromatique, l'équation de base est la suivante [RAMASWAMI et coll., 2001] :

$$|D(\lambda)|LB(\Delta\lambda) < \varepsilon \quad (1.1)$$

où  $|D(\lambda)|$  : taux de dispersion chromatique de la fibre (en s/(nm.km))

$L$  : longueur de la fibre (en km)

$B$  : débit de l'information transportée (en b/s)

$\Delta\lambda$  : largeur spectrale du signal transporté (en nm)

$\varepsilon$  : fraction de la période d'un bit sur laquelle l'énergie est dispersée par la dispersion chromatique

Selon la norme européenne ITU G.957,  $\varepsilon < 0,306$  correspond à une pénalité de puissance de 1 dB, tandis que selon la norme nord-américaine Telcordia GR-253,  $\varepsilon < 0,491$  correspond à une pénalité de puissance de 2 dB. Ainsi, pour une transmission à 10 Gb/s avec une fibre de transport typique, la distance limite de propagation pour une pénalité de puissance d'au plus 2 dB est de :

$$L < \frac{\varepsilon}{|D|B\Delta\lambda} = \frac{0,491}{17E-12 \cdot 10E9 \cdot 0,2} = 14,4 \quad (1.2)$$

Cela signifie qu'à 10 Gb/s, pour aller plus loin que 14,4 km et maintenir la pénalité de puissance à au plus 2 dB, il faut compenser la dispersion chromatique.

### 1.2.3 Variation du taux de dispersion chromatique

Le taux de dispersion chromatique  $D(\lambda)$  varie en fonction de la longueur d'onde  $\lambda$ . Pour le représenter, on utilise souvent l'équation empirique du modèle de Sellmeier [HOULE 2006] :

$$D(\lambda) = \frac{\lambda S_0}{4} \left( 1 - \left( \frac{\lambda_0}{\lambda} \right)^4 \right) \quad (1.3)$$

où  $\lambda_0$  : longueur d'onde de référence

$S_0$  : pente de la dispersion chromatique à la longueur d'onde  $\lambda_0$

Pour une fibre de transport typique, on a :

$$\lambda_0 = 1310 \text{ nm}$$

$$S_0 = 0,092 \text{ ps} / \text{nm}^2 \cdot \text{km}$$

La figure 1.2 présente le taux de dispersion chromatique résultant.

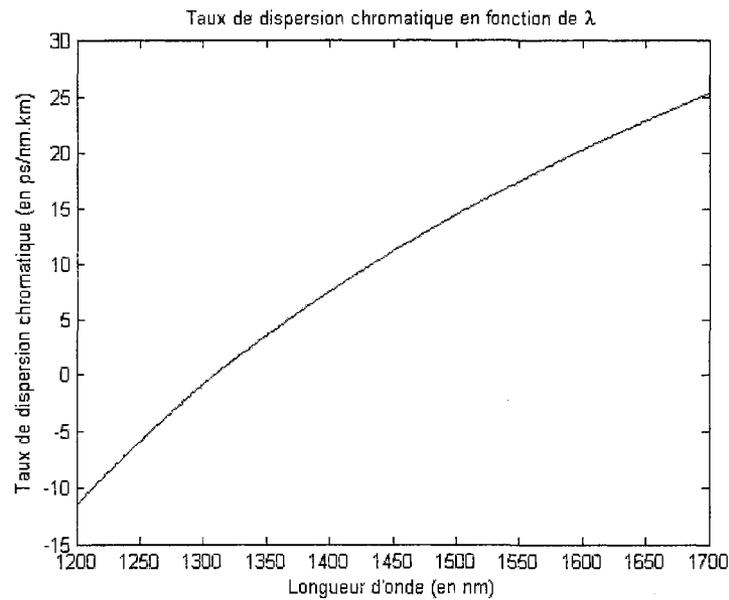


Figure 1.2 – Taux de dispersion chromatique

#### 1.2.4 Autres caractéristiques de la fibre

En plus de faire subir aux signaux l'effet néfaste de la dispersion chromatique, la fibre optique atténue la lumière qui la traverse, c'est-à-dire que la puissance optique du signal diminue avec la distance parcourue. L'atténuation est autour de 0,2 dB/km pour les longueurs d'onde dans la bande C (entre 1530 nm et 1565 nm), la principale bande de longueurs d'onde utilisée pour les télécommunications par fibre optique. Puisque le taux d'erreur de transmission dépend de la puissance reçue au récepteur, l'atténuation limite la longueur maximale d'un lien optique pour un taux d'erreur maximal donné. Pour contrer ce problème, il existe des amplificateurs optiques, qui permettent d'augmenter la puissance d'un signal optique sans avoir à le

régénérer. Dans les amplificateurs optiques dopés à l'erbium (EDFA – *Erbium Doped Fiber Amplifier*), l'amplification optique survient lorsque des ions  $\text{Er}^{3+}$  excités sont stimulés par un photon incident et retombent à leur état initial en émettant un nouveau photon ayant les mêmes caractéristiques que le photon incident. C'est l'émission stimulée. Un ion  $\text{Er}^{3+}$  peut demeurer excité pendant une période d'environ 10 ms. Au-delà de 10 ms, s'il n'a pas été excité par un photon incident, il retombe naturellement à son état initial, émettant tout de même un photon. C'est l'émission spontanée. Ce phénomène d'émission spontanée ajoute un bruit aléatoire au signal, ce qui peut éventuellement causer des erreurs à la réception.

D'autre part, la fibre optique utilisée dans les réseaux optiques métropolitains est généralement une fibre monomode. Par définition, une fibre monomode ne supporte qu'un seul mode de propagation, lequel se décompose en deux modes de polarisation. Dans une fibre monomode idéale, la parfaite symétrie du guide d'onde fait en sorte que ces deux modes ont les mêmes propriétés de propagation et apparaissent donc comme un seul. En pratique cependant, plusieurs causes viennent modifier ce comportement idéal.

Ces perturbations possibles peuvent être soit intrinsèques, comme par exemple le cœur de la fibre qui n'est pas parfaitement circulaire, soit extrinsèques, par exemple un pincement, une courbure ou une torsion de la fibre [AGRAWAL 1997]. Dans tous les cas, ces perturbations causent de la biréfringence, c'est-à-dire que chacun des modes de polarisation ne voit pas le même indice de réfraction effectif. En conséquence, les constantes de propagation des deux modes diffèrent et donc, il y a un délai entre la réception de la puissance contenue dans chacun des modes. Temporellement, l'effet cumulatif est un élargissement des impulsions. Ce phénomène est appelé la dispersion des modes de polarisation (PMD) et est caractérisé par le délai différentiel de groupe moyen  $\langle \Delta\tau \rangle$ , avec :

$$\langle \Delta\tau \rangle = D_{PMD} \sqrt{L} \quad (1.4)$$

où  $D_{PMD}$  : taux de dispersion des modes de polarisation de la fibre (en  $\text{ps}/\sqrt{\text{km}}$ )

$L$  : longueur de la fibre (en km)

La PMD peut être négligée pour autant que le délai différentiel de groupe moyen  $\langle \Delta\tau \rangle$  demeure sous  $0,1 T$ ,  $T$  étant la période d'un bit [RAMASWANI et coll., 2001].

Pour un réseau métropolitain à 10 Gb/s, la PMD était jusqu'à récemment un objet de préoccupation au même titre que la dispersion chromatique. La plupart sinon toutes les fibres installées avant 1995 ont des problèmes à transmettre à 40 Gb/s et parfois même à 10 Gb/s [NOÉ et coll., 2004]. En effet, pour 30% de ces fibres, l'accumulation de PMD dépasse la limite établie pour maintenir une faible probabilité d'erreur après 80 km [BÜLOW 2000].

Pour une fibre ayant plusieurs années d'existence, le taux  $D_{PMD}$  peut valoir autour de  $2\text{ps}/\sqrt{\text{km}}$ . Dans un tel cas, la distance limite de propagation vaut :

$$\langle \Delta\tau \rangle = D_{PMD} \sqrt{L} < 0,1T \rightarrow L < \left( \frac{0,1T}{D_{PMD}} \right)^2 = \left( \frac{0,1 \cdot 100\text{ps}}{2\text{ps}/\sqrt{\text{km}}} \right)^2 = 25 \text{ km} \quad (1.5)$$

Toutefois, la technologie s'est améliorée depuis et les fibres optiques déployées actuellement, de meilleure qualité, ont un taux  $D_{PMD}$  autour de  $0,1\text{ps}/\sqrt{\text{km}}$ . La distance limite de transmission sans avoir à se soucier de la PMD s'établit alors à :

$$L < \left( \frac{0,1T}{D_{PMD}} \right)^2 = \left( \frac{0,1 \cdot 100\text{ps}}{0,1\text{ps}/\sqrt{\text{km}}} \right)^2 = 10000 \text{ km} \quad (1.6)$$

À 10 Gb/s, sur une fibre contemporaine et à distance régionale/métropolitaine (jusqu'à 600 km environ), la PMD n'est donc pas un phénomène dont il faut absolument tenir compte. Par conséquent, le présent projet considère le cas où la PMD n'est pas un problème. Néanmoins, il est important de connaître l'existence de ce phénomène, car il est limitatif dans les systèmes de communication avec fibres dégradées. De plus, un jour les réseaux métropolitains voudront certainement migrer vers 40 Gb/s et à ce moment, la PMD devra impérativement être considérée.

En plus de la dispersion chromatique et de la dispersion des modes de polarisation, plusieurs autres phénomènes physiques peuvent affecter la transmission. C'est le cas de l'effet Brillouin, de l'effet Raman, de la « Self-Phase Modulation », de la « Cross-Phase Modulation » et du « Four-Wave Mixing ». Ces effets non-linéaires ne se manifestent pas si chaque longueur d'onde est exploitée à une puissance moyenne de 5 dBm ou moins dans une fibre dont le cœur a typiquement  $80 \mu\text{m}^2$  de surface (ce qui est le cas pour une SMF-28, une fibre de

transport typique) [KAMINOW et coll., 1997]. À cette puissance dans un tel cœur, la densité de puissance atteint  $40 \text{ MW/m}^2$ . Il est d'usage d'exploiter les réseaux métropolitains/régionaux à 5 dBm ou moins par longueur d'onde, de manière à éviter les effets non-linéaires. Par conséquent, dans le cadre de cette recherche, il n'est pas nécessaire de tenir compte de ces phénomènes non linéaires, puisqu'une opération dans le domaine linéaire seulement est considérée.

### 1.2.5 Paillement

Les phénomènes physiques décrits précédemment sont tous reliés à la fibre optique. Il existe un dernier phénomène physique qui affecte le signal mais qui provient de la modulation plutôt que de la fibre elle-même : le paillement. Un signal qui subit du paillement voit sa fréquence augmenter ou diminuer avec le temps. La figure 1.3 montre un tel signal. Le paillement cause un étalement du spectre fréquentiel du signal et augmente donc l'impact de la dispersion chromatique.

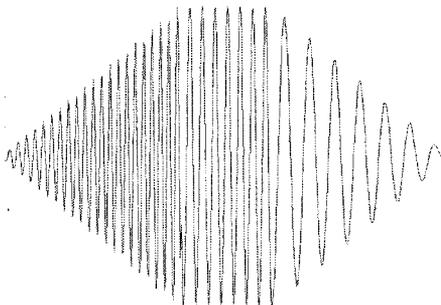


Figure 1.3 – Signal ayant subi du paillement [ISSA 2007]

Quelle est la cause physique du paillement ? D'abord, il faut savoir qu'il existe plusieurs méthodes de modulation. La modulation directe consiste à faire varier le courant qui alimente le laser. En fonction du courant, il y a ou non émission de lumière par le laser. Or, la variation de l'intensité du courant induit une variation de l'indice de réfraction du laser. Par conséquent, la phase du signal varie, d'où le paillement. La modulation directe n'est donc pas habituellement utilisée pour les transmissions à des débits supérieurs à 2,5 Gb/s à cause de l'importance du paillement.

Pour la télécommunication à haut débit (supérieur à 2,5 Gb/s), ce sont généralement des modulateurs externes qui sont utilisés. Ceux-ci modulent le faisceau émis par le laser. Le laser émet continuellement de la lumière et le modulateur externe, selon le bit à transmettre, laisse passer ou non cette lumière. Il existe deux types de modulateurs externes : les modulateurs à électro-absorption et les modulateurs Mach-Zender [HOULE 2006]. Les modulateurs à électro-absorption utilisent le phosphore d'indium, un matériau qui devient opaque lorsqu'un courant électrique le traverse. Toutefois, l'indice de réfraction est modulé en même temps que la transmittivité, ce qui cause du paillement.

Les modulateurs Mach-Zender misent plutôt sur l'alternance entre interférence constructive et destructive. Un guide optique comportant deux branches est construit sur substrat de  $\text{LiNbO}_3$ . L'indice de réfraction de ce matériau change en fonction du champ électrique auquel il est soumis. Ainsi, pour transmettre un '0', un champ électrique est appliqué sur l'une des branches afin de provoquer une différence de phase de  $180^\circ$ , provoquant de l'interférence destructive. S'il est conçu avec soin, un modulateur Mach-Zender ne cause pas de paillement.

Du point de vue du coût, la modulation directe est la méthode la moins chère. Un modulateur à électro-absorption est beaucoup plus coûteux qu'une diode laser, mais le modulateur Mach-Zender est de loin la solution la plus onéreuse.

Maintenant que les problèmes potentiels liés à la transmission d'informations par fibre optique sont connus, il est temps de regarder les solutions possibles, principalement pour le phénomène de la dispersion chromatique. Il existe typiquement trois familles de méthodes : la pré-compensation, la compensation optique et la compensation électronique. La prochaine section s'attarde à la première famille, la pré-compensation.

### **1.3 Solution 1 : Pré-compensation**

La pré-compensation, comme son nom l'indique, est effectuée avant que le signal ne soit transmis par la fibre optique. Les premières recherches en ce sens ont porté sur le paillement. Plus récemment, l'impact du format de modulation a été étudié.

### 1.3.1 Pré-piaillage

La technique du pré-piaillage tire profit du fait que dans la fibre optique standard, les hautes fréquences se propagent plus rapidement que les basses fréquences. Ainsi, en transmettant les composantes de basse fréquence au début de l'impulsion et celles de haute fréquence à la fin, la distance maximale de propagation est augmentée. La figure 1.4, tirée de [HENMI et coll., 1994], présente ce principe.

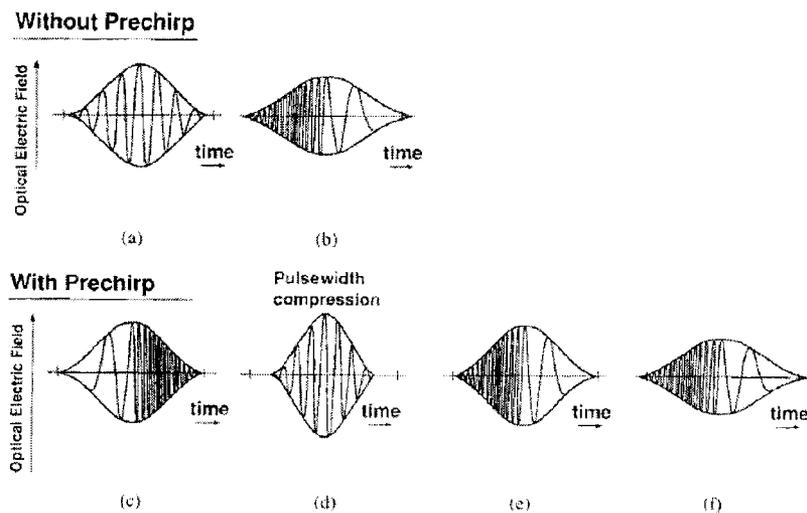


Fig. 1. Principle of prechirp. (a) Ideal intensity modulated waveform. (b) Received optical waveform. (c) Prechirped optical waveform. (d) and (e) Middle of a transmission fiber. (f) Received optical waveform.

Figure 1.4 – Principe du pré-piaillage

Le pré-piaillage permet en théorie de doubler la distance de propagation [HENMI et coll., 1994]. Toutefois, puisque le signal se comprime dans la première partie de la transmission, la densité de puissance dans la fibre augmente et peut dépasser le seuil d'apparition des effets non-linéaires. Les phénomènes non-linéaires viennent donc limiter les performances du pré-piaillage.

### 1.3.2 Format de modulation

Le format de modulation le plus fréquent est le NRZ-OOK, c'est-à-dire le « non return to zero », « on-off keying ». Le « on-off keying » (OOK) signifie que les '1' et les '0' de la suite de bits correspondent à la présence ou à l'absence de lumière émise dans la fibre. Quant au NRZ, il

faut le présenter en opposition au format RZ, « return to zero », pour lequel la lumière est émise durant seulement une fraction de la durée d'un bit lors de la transmission d'un '1'. À l'opposé, pour la modulation NRZ, la lumière est émise durant toute la durée du bit.

Or, certains formats de modulation sont moins sensibles à la dispersion chromatique que d'autres. Une étude a démontré que les formats DPSK (« differential phase-shift keying ») et DB (« duobinary ») peuvent être démodulés avec un faible taux d'erreur pour une plus grande dispersion accumulée que le NRZ-OOK [CURRI et coll., 2004]. Une autre étude montre également la supériorité de ces formats de modulation face à la dispersion chromatique [GNAUCK et coll., 2004].

Malgré ces avantages, la pré-distorsion du signal augmenterait la susceptibilité aux effets non linéaires [ESSIAMBRE et coll., 2006]. De plus, les formats de modulation avancés demandent l'utilisation d'un modulateur Mach-Zehnder, d'où un émetteur plus complexe et plus dispendieux.

## **1.4 Solution 2 : Compensation optique**

Une deuxième approche pour compenser la dispersion chromatique est de le faire dans le domaine optique. La première méthode présentée est la solution traditionnelle : l'utilisation de fibres compensatrices (DCF – « Dispersion Compensating Fiber »). La deuxième méthode, les réseaux de Bragg, est en développement.

### **1.4.1 Fibre DCF**

Le principe de la fibre compensatrice est simple : le profil d'indice de la fibre est modifié pour obtenir une fibre à dispersion inversée par rapport à la fibre de transmission. En plaçant un tronçon d'une longueur appropriée de cette fibre à la suite du tronçon de fibre standard, les effets de la dispersion chromatique s'opposent et s'annulent pour certaines longueurs d'onde. En effet, la valeur de la dispersion chromatique dépend de la longueur d'onde. Pour que la compensation soit totale pour toutes les longueurs d'onde, il faudrait produire une fibre avec une dispersion de pente et de courbure parfaitement inverse à celle de la fibre de transmission, ce qui est difficilement réalisable en pratique. Selon l'application, on peut soit s'intéresser à la

dispersion résiduelle à une longueur d'onde spécifique, soit à la largeur de bande utilisable, donc la largeur de bande pour laquelle la dispersion résiduelle est en deçà d'une certaine valeur.

En plus de la dispersion résiduelle, il faut s'intéresser à l'atténuation produite par la fibre compensatrice, puisque cela a un impact sur le budget de puissance du système. En effet, pour que la même puissance soit reçue, il faut soit réduire la longueur du lien, soit ajouter de l'amplification, ce qui réduit le rapport signal sur bruit. Pour comparer l'efficacité des DCF entre elles, on utilise fréquemment la figure de mérite (FOM), définie comme suit :

$$FOM = -\frac{D_{DCF}}{\alpha_{DCF}} \quad (1.7)$$

où  $D_{DCF}$  : dispersion de la fibre de compensation

$\alpha_{DCF}$  : atténuation de la fibre de compensation

La figure de mérite doit être aussi élevée que possible, puisque l'on désire compenser le plus de dispersion possible pour la plus faible atténuation possible. On retrouve sur le marché des fibres ayant une figure de mérite entre 220 et 420 ps/(nm·dB) [GRÜNER-NIELSEN et coll., 2005]. Finalement, la DCF a un cœur plus petit qu'une fibre de transport typique, occasionnant ainsi une plus grande densité de puissance. De ce fait, elle est plus sensible aux effets non-linéaires.

#### 1.4.2 Réseau de Bragg

Un réseau de Bragg est une fibre dont l'indice de réfraction change par pas dans le sens de la propagation. Cela crée une cavité résonante et cause la réflexion d'une longueur d'onde particulière. Cette longueur d'onde est déterminée par la condition de Bragg [HOULE 2006] :

$$\lambda_0 = 2n_{eff} \Lambda \quad (1.8)$$

où  $\lambda_0$  : longueur d'onde réfléchie

$n_{eff}$  : indice de réfraction effectif de la fibre

$\Lambda$  : pas du réseau

Dans la bande C de la fibre optique, les longueurs d'onde plus courtes se propagent plus rapidement dans la fibre que les longueurs d'onde plus longues. Pour compenser cette dispersion chromatique, l'idée est de faire varier le pas afin que les différentes longueurs d'onde ne parcourent pas la même distance dans le réseau avant d'être réfléchies. Ainsi, on peut parvenir à compenser le retard des basses fréquences (longueurs d'onde plus longues) en leur faisant parcourir une distance plus courte dans le réseau. Ce principe est illustré à la figure 1.5.

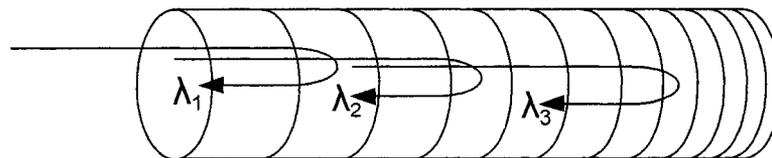


Figure 1.5 – Réseau de Bragg à pas variable

À un débit de 10 Gb/s, la compensation maximale est de l'ordre de -2000 ps/nm pour une compensation fixe pour un canal de transmission [PUREUR et coll., 2004]. Pour une longueur d'onde de 1550 nm dans une fibre de transport typique,  $D(\lambda) = 17,5$  ps/(nm.km). Un réseau de Bragg pourrait donc compenser la dispersion chromatique accumulée de 114 km de fibre.

### 1.5 Solution 3 : Égalisation

Les techniques dans le domaine optique doivent être mises en place physiquement avant le récepteur. Une approche a priori moins coûteuse et demandant moins de modifications à la couche physique est de compenser la dispersion chromatique électroniquement, c'est-à-dire une fois la conversion optique-électrique faite. Tout l'équipement nécessaire peut ainsi être inclus au récepteur, ce qui facilite la transition d'une technologie à une autre.

Il existe deux principales approches pour la compensation dans le domaine électrique : l'égalisation et la correction d'erreurs. L'égalisation est l'objet de la présente section, tandis que la correction d'erreurs est abordée à la prochaine section.

Le but de l'égalisation est de se servir des informations connues concernant la fibre de transport pour approximer avec le moins d'erreur possible la séquence de bits transmise à partir de la séquence reçue. C'est le principe à la base des quatre méthodes présentées dans cette section, soit le seuil de décision adaptatif, le *Feed-Forward Equalizer* (FFE), le *Decision-Feedback Equalizer* (DFE) et le *Maximum Likelihood Sequence Estimation* (MLSE).

### 1.5.1 Seuil de décision adaptatif

Les caractéristiques du bruit et des perturbations changent avec le temps et selon la séquence de bits transmise. Par conséquent, le seuil de décision optimal permettant de minimiser le taux d'erreur varie dans le temps. L'idée est donc d'ajuster le seuil de décision en fonction de la séquence de bits reçue pour minimiser le taux d'erreur. La figure 1.6, tirée de [MATSUMOTO et coll., 2001], présente le fonctionnement général d'un tel système.

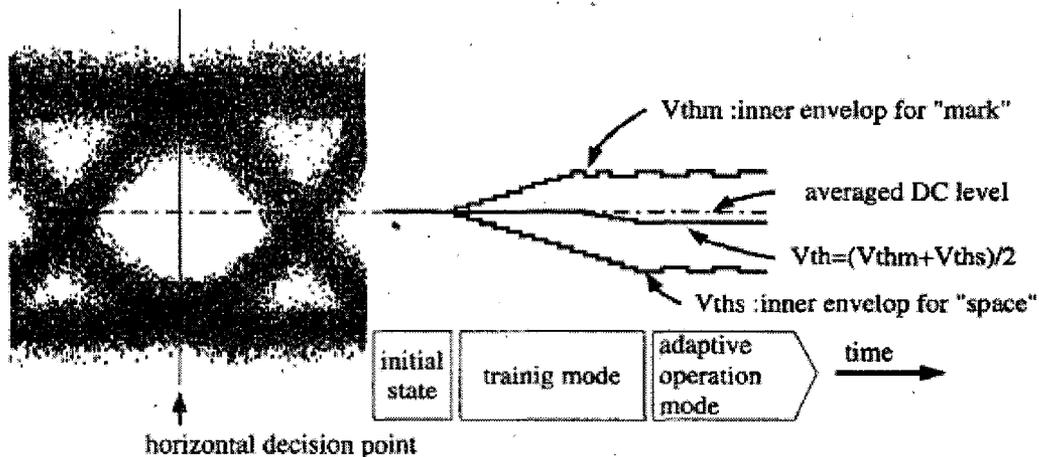


Figure 1.6 – Système avec seuil de décision adaptatif

Initialement, les seuils  $V_{thm}$ ,  $V_{th}$  et  $V_{ths}$  sont fixés à la valeur moyenne du signal. Puis, lors de la phase d'entraînement,  $V_{thm}$  est augmenté et  $V_{ths}$  est diminué jusqu'à ce qu'ils atteignent les limites de l'œil. Le seuil de décision  $V_{th}$  est fixé au milieu des deux.

Le système permet d'atteindre un taux d'erreur inférieur à  $10^{-12}$  après 600 km de fibre optique monomode à 2,5 Gb/s [MATSUMOTO et coll., 2001].

### 1.5.2 Feed-Forward Equalizer (FFE)

L'idée derrière le FFE est de concevoir un filtre dont la fonction de transfert est l'inverse de la fonction de transfert de la fibre de transport. En combinant en cascade la fibre de transport et sa fonction de transfert inverse, alors l'effet de la fibre est en théorie annulé.

Soit  $x(t)$  le signal reçu, alors la sortie du filtre vaut [CURRI et coll., 2004]:

$$y(t) = \sum_{i=0}^{N_{taps}-1} C_i \cdot x(t - i\Delta T) \quad (1.9)$$

- où
- $N_{taps}$  : nombre de valeurs gardées en mémoire
  - $\Delta T$  : délai entre les lectures
  - $C_i$  : coefficients du filtre

La figure 1.7, tirée de [CURRI et coll., 2004], présente le rapport signal sur bruit (OSNR) en fonction de la dispersion chromatique accumulée pour un taux d'erreur de  $10^{-12}$ , lorsque  $N_{taps}$ ,  $\Delta T$  et  $C_i$  sont optimisés. Les résultats montrent que l'amélioration apportée par l'égalisation dépend fortement du format de modulation.

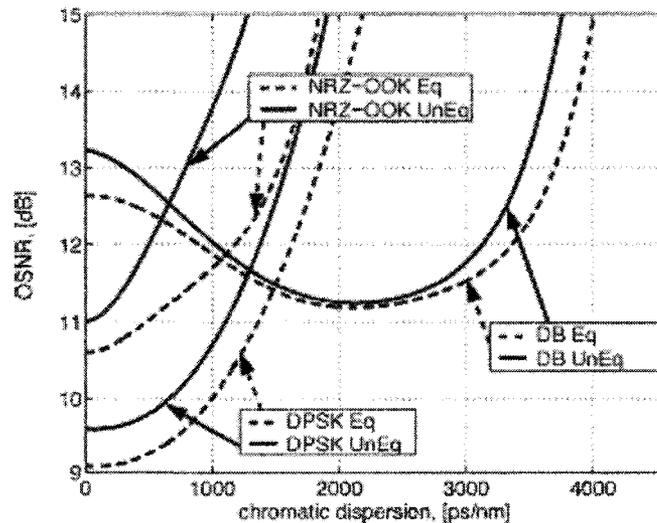


Figure 1.7 – Performances d'un égalisateur FFE

### 1.5.3 Decision-Feedback Equalizer (DFE)

Supposons qu'au récepteur, on reçoive la séquence '10'. Puisqu'il y a de l'interférence inter-symbole, le niveau de tension pour le '0' sera supérieur à ce qu'il devrait être à cause du '1' qui le précède. L'idée est ici de soustraire au signal reçu une certaine portion du niveau de tension associé au bit précédent. Ainsi, dans l'exemple précédent, on soustrairait à la tension reçue pour le '0' une partie de la tension d'un '1'. Par conséquent, le circuit de décision pourrait plus facilement détecter un '0', sans avoir à être modifié.

Le DFE est la plupart du temps utilisé conjointement avec le FFE pour améliorer ses performances. La figure 1.8, tirée de [WANG et coll., 2004], présente le schéma d'un tel système d'égalisation, où les coefficients des deux filtres sont adaptés en temps réel par l'algorithme du moindre carré.

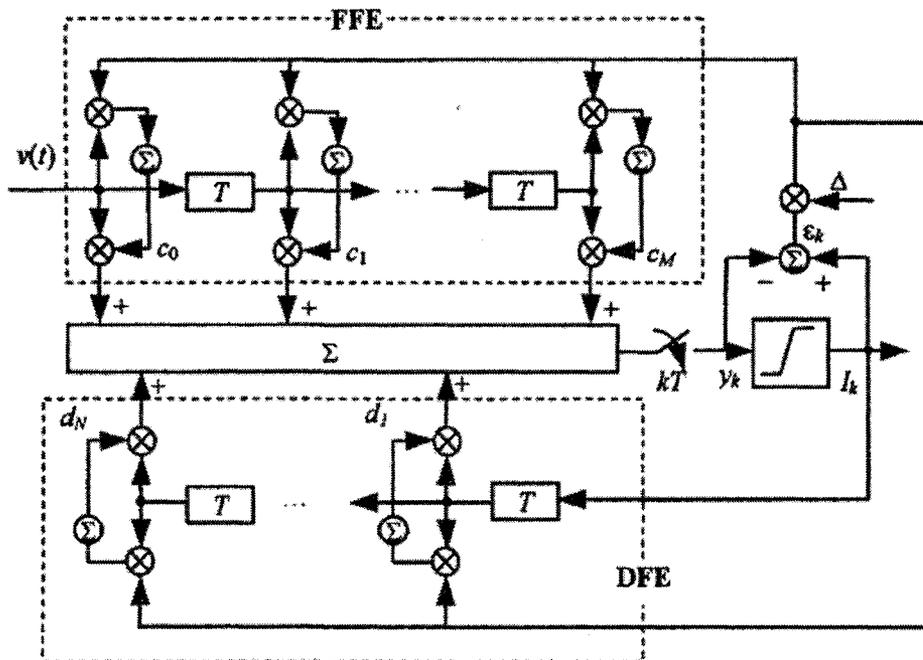


Figure 1.8 – Schéma d'un égalisateur FFE + DFE

La figure 1.9, également tirée de [WANG et coll., 2004], montre que l'ajout du DFE améliore très faiblement les performances du FFE.

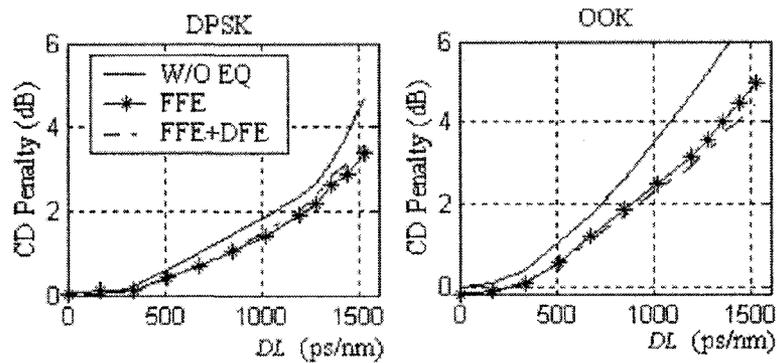


Figure 1.9 – Performances d'un égalisateur FFE + DFE

#### 1.5.4 Maximum Likelihood Sequence Estimation (MLSE)

Cette méthode a recours à un traitement probabiliste des niveaux de tension reçus. La décision entre '1' et '0' n'est pas prise au moment de la réception du signal, mais plutôt après la réception d'une séquence de bits d'une longueur déterminée. La figure 1.10, tirée de [AGAZZI et coll., 2005], montre le schéma général d'un tel système. Un premier système tient compte des caractéristiques estimées du canal pour produire les statistiques utiles au décodeur Viterbi. Celui-ci détermine alors, pour un nombre de bit préétabli, la séquence de bits qui est la plus probable d'avoir été émise parmi toutes les séquences de bits possibles. Une fois la décision prise, l'information est transmise à l'estimateur des caractéristiques du canal, pour que ces dernières soient mises à jour.

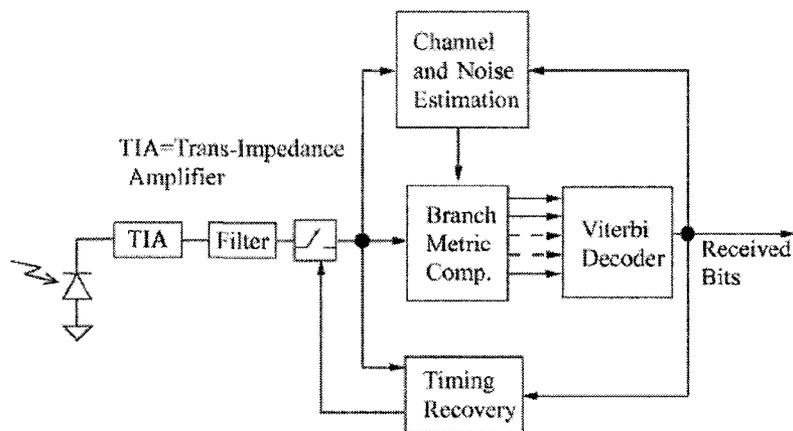


Figure 1.10 – Système de compensation MLSE

Cette méthode fournit les meilleures performances parmi toutes les techniques d'égalisation [SPINNLET et coll., 2006]. Par contre, la plus grande complexité algorithmique implique un circuit électronique plus complexe et de plus grande dimension, d'où une consommation de puissance plus élevée.

## 1.6 Solution 4 : Correction d'erreurs

L'égalisation n'est pas le seul type de compensation électronique. Il existe également la correction d'erreurs, qui est décrite dans la présente section. Une bonne source d'information est le livre *Error Control Coding : Fundamentals and Applications* [LIN et coll., 1983], d'où est tirée la majeure partie de l'information présentée ici.

Le but de la correction d'erreurs est de détecter quels bits reçus sont erronés et de les corriger dans la mesure du possible. Cela se fait en ajoutant de la redondance aux bits transmis. En fait, on pourrait diviser cette approche en deux méthodes : FEC (forward error correction) et ARQ (automatic repeat request). La FEC corrige automatiquement les erreurs détectées au récepteur tandis que l'ARQ envoie une requête de retransmission au transmetteur lorsqu'une erreur de transmission est détectée. Toutefois, pour des raisons pratiques, l'ARQ n'est pas envisagé pour les systèmes de communication par fibre optique.

### 1.6.1 Codes correcteurs d'erreurs

Il y a principalement deux types de codes correcteurs : les codes en bloc et les codes convolutifs. Pour les codes en bloc, la séquence d'information est segmentée en blocs messages de  $k$  bits. L'encodeur transforme, selon certaines règles, chaque message de  $k$  bits en un mot code de  $n$  bits, avec  $n > k$ . Il existe une grande quantité de sous-catégories et de familles de codes en bloc, qui ne seront pas présentées ici. Mentionnons toutefois les codes BCH (Bose-Chaudhuri-Hocquenghem) et les codes RS (Reed-Solomon), qui sont fréquemment employés en pratique. La figure 1.11 présente un exemple de code en bloc, utilisant le code BCH(7,4).

0 1 0 1 → 0 1 0 1 1 0 0

Figure 1.11 – Exemple d'un code en bloc BCH(7,4)

Les codes convolutifs ne séparent pas la séquence d'information en blocs mais codent plutôt celle-ci au fur et à mesure que les bits arrivent, de façon continue. Pour chaque groupe de  $k$  bits reçu, on émet  $n$  bits qui sont une combinaison linéaire des  $m$  derniers groupes de  $k$  bits reçus (incluant le groupe en cours). D'un point de vue pratique, on peut implémenter un encodeur bloc avec une multiplication matricielle, tandis qu'un encodeur convolutif s'implémente grâce à une machine à états finis. La figure 1.12 présente un exemple d'encodeur convolutif avec  $k = 1$ ,  $n = 3$  et  $m = 3$ .

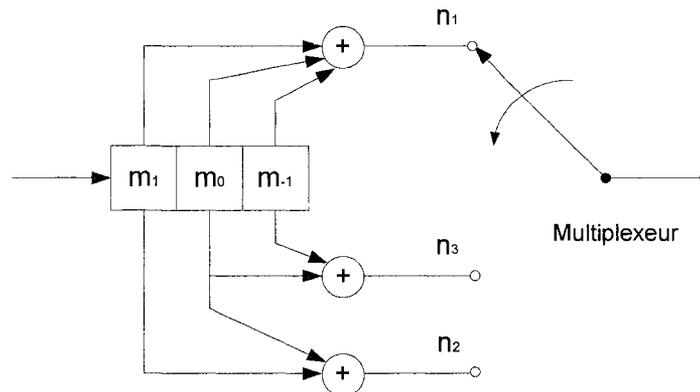


Figure 1.12 – Exemple d'encodeur convolutif

Pour comparer les performances des différents codes, la caractéristique la plus souvent employée est le gain de codage (CG, « coding gain » en anglais). Le CG est la diminution de puissance requise du signal codé par rapport au signal non codé pour obtenir le même taux d'erreur à la réception. Par exemple, si le gain de codage est de 3 dB, cela signifie que grâce au codage, on peut émettre un signal deux fois moins puissant pour obtenir le même taux d'erreur. Il ne faut pas oublier de tenir compte de l'augmentation de débit nécessaire pour transporter la même quantité d'information. En effet, à cause de l'ajout de bits de redondance, le débit binaire doit être augmenté pour transmettre le même débit d'information. Le gain de codage net (NCG, « net coding gain » en anglais) est donc la diminution de puissance requise pour le signal codé avec un débit binaire augmenté, par rapport au signal non codé au débit binaire initial. La figure 1.13, tirée de [BARLOW 2003], montre comment mesurer le gain de codage à partir du taux d'erreur en fonction de la puissance reçue.

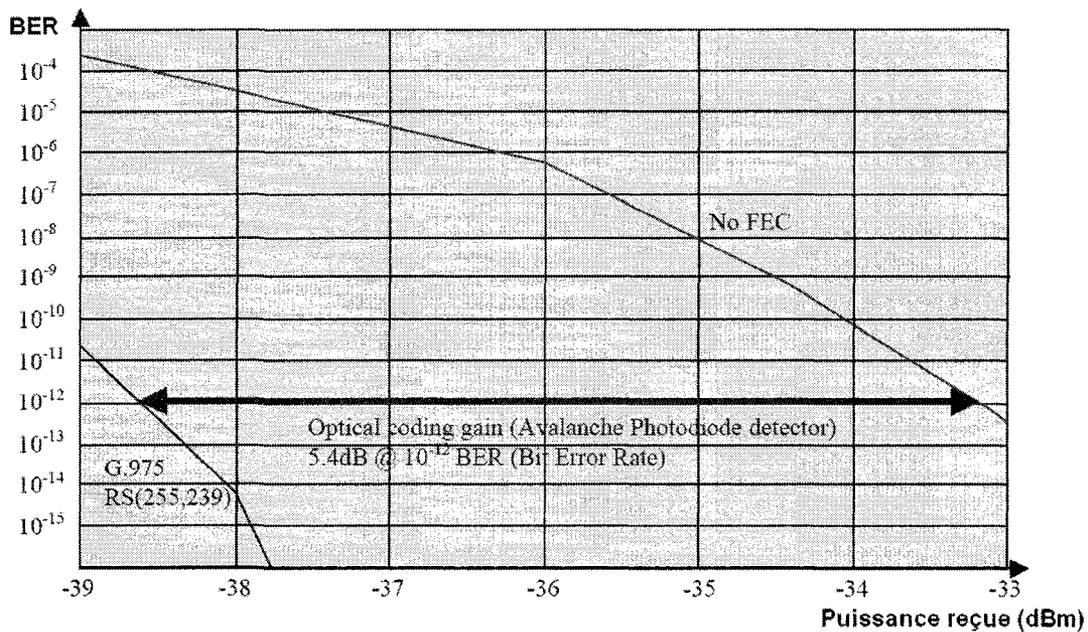


Figure 1.13 – Détermination du gain de codage

Actuellement, dans les réseaux de communication par fibre optique, ce sont principalement les codes en blocs qui sont utilisés. Un code FEC est intégré dans le protocole OTN, défini dans les normes ITU-T G.709 et G.975. Le code employé est un Reed-Solomon RS(255,239), qui peut corriger jusqu'à 8 erreurs par 255 bits transmis [BARLOW 2003] et qui a un NCG de 5,8 dB [OUCHI et coll., 2006].

### 1.6.2 Turbo codes

Selon le théorème de Shannon, si le débit d'information transmis par la source est inférieur à la capacité du canal, alors il est théoriquement pensable de concevoir un traitement du signal qui permette de transmettre le contenu de la source sur le canal avec une probabilité d'erreur aussi petite que voulue [BOUTIN 2005]. De ce théorème découle la limite de Shannon, une limite théorique souvent exprimée comme le rapport signal sur bruit minimum requis pour qu'un débit d'information donné soit transmis avec une probabilité d'erreur aussi faible que souhaitée.

Une façon simple de s'approcher de la limite de Shannon est d'allonger le code. Or, pour un code FEC standard, la complexité du décodeur s'accroît avec la longueur du code. Et plus un décodeur est complexe, plus il est difficile à réaliser en pratique. Les turbo codes permettent de contourner ce problème. L'idée : décomposer un problème complexe en sous-problèmes plus simples qu'on traite successivement avec un processus itératif.

Un encodeur turbo est donc construit comme l'association de deux encodeurs FEC standards (blocs ou convolutifs) liés par une fonction de permutation. Chacun des deux encodeurs reçoit donc à son entrée les mêmes bits, mais dans une séquence différente.

La figure 1.14 présente le schéma bloc général d'un tel système dans le cas d'un code turbo en bloc.

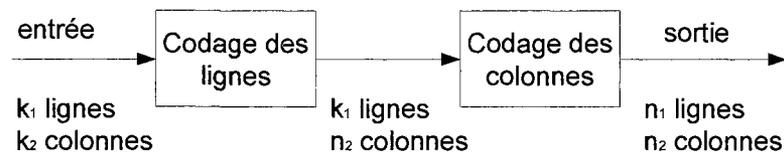


Figure 1.14 – Schéma bloc général d'un encodeur turbo code

Pour le décodage, la manière la plus simple de procéder est illustrée à la figure 1.15.

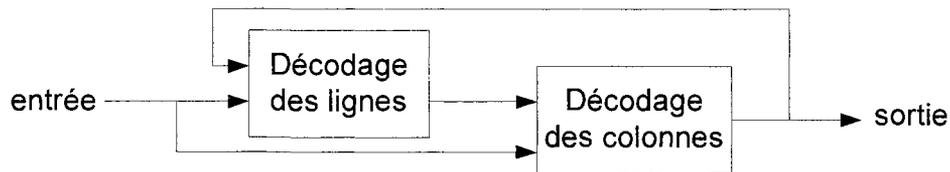


Figure 1.15 – Schéma bloc général d'un décodeur turbo code

Pour expliquer le fonctionnement des turbo codes, on utilise souvent l'analogie avec une grille de mots croisés. Dans une grille de mots croisés, un premier décodage horizontal permet de remplir certaines cases, puis le décodage vertical confirme ou remet en cause les cases remplies et permet aussi de remplir d'autres cases. On recommence la suite décodage

horizontal – décodage vertical jusqu'à ce que la solution cesse d'évoluer. Vu autrement, on peut dire qu'on utilise le résultat du décodage vertical pour améliorer le traitement horizontal et vice versa. Le principe de base est le même pour le décodeur de turbo code.

Bien sûr, l'analogie n'est pas parfaite. Elle laisse croire qu'à chaque étape, le décodeur prend une décision stricte quant à la valeur '1' ou '0' de chaque bit d'information. En réalité, les turbo codes utilisent plutôt un traitement probabiliste, c'est-à-dire qu'ils associent aux valeurs binaires '0' et '1' des poids de vraisemblance qui sont mis à jour tout au long des différents traitements dans le récepteur, jusqu'à la décision finale. Pour établir les poids de vraisemblance, l'algorithme de décodage utilise un modèle du bruit en présence.

Récemment, Mitsubishi Electric a réalisé un circuit avec turbo code qui s'approche à 2,2dB de la limite de Shannon avec un NCG de 10,1dB. La redondance est de 21% et le code utilisé est la concaténation d'un BCH(144,128) avec un BCH(256,239) [OUCHI et coll., 2006]. Comparativement, le RS(255,239) des normes ITU-T G.709 et G.975 a un NCG de 5.8 dB avec 7% de redondance. Une autre étude démontre que l'utilisation d'un modèle de bruit blanc gaussien dans l'algorithme de décodage dégrade de façon significative les performances des turbo codes en présence d'un bruit non gaussien [CAI et coll., 2003]. Les turbo codes sont donc des codes de correction d'erreur aux résultats intéressants qui peuvent probablement encore être améliorés.

## 1.7 Bilan

La compensation électronique semble être la solution retenue par l'industrie en remplacement de la compensation optique. Plusieurs compagnies se sont récemment associées pour tester expérimentalement leurs produits, dans le but de créer un standard [GHIASI et coll., 2006]. Toutes les puces soumises au test, qui alliaient FFE, DFE et MLSE, obtenaient après 140 km un taux d'erreur de  $10^{-12}$ . Toutefois, parmi l'ensemble des méthodes de compensation électronique, les turbo codes est celle qui présente le plus de potentiel. Les premiers résultats obtenus sont excellents et il semble être possible de les améliorer davantage.

La pré-compensation permet d'obtenir de bons résultats, mais ce n'est pas la meilleure solution lorsque le but est d'obtenir un compensateur à faible coût. En effet, la pré-compensation

nécessite l'utilisation d'un modulateur Mach-Zender, le modulateur le plus cher. De plus, l'utilisation de cette technique nécessite de remplacer à la fois l'émetteur et le récepteur, ce qui engendre des coûts supplémentaires. C'est pourquoi la pré-compensation n'est pas retenue pour ce projet.

## **1.8 Projet de recherche**

### **1.8.1 Intérêt d'une nouvelle enquête**

La compensation électronique de la dispersion chromatique est une technologie en pleine émergence. Le but d'un système de communication est à la base de transporter des bits d'informations de la source à la destination de manière fiable, c'est-à-dire avec le moins d'erreurs possibles. Mais les impératifs économiques ajoutent une contrainte supplémentaire, le meilleur coût possible, et à cet égard, l'approche électronique semble en effet la plus prometteuse. C'est pourquoi plusieurs recherches sont en cours pour tenter d'améliorer les algorithmes existants ou pour identifier de nouvelles techniques plus performantes. La présente recherche s'inscrit dans cette mouvance.

En examinant les solutions actuellement exploitées pour corriger la dispersion chromatique, l'auteure s'est rendue compte que l'une d'elle pourrait potentiellement être améliorée. En effet, les codes de correction d'erreurs sont tous conçus pour corriger du bruit blanc gaussien. Or, la dispersion chromatique ne cause pas du bruit blanc gaussien, mais plutôt de l'interférence inter-symbole, un phénomène déterministe. De plus, son impact sur la distribution statistique du bruit ne semble pas avoir encore fait l'objet d'études. Cela mène à l'hypothèse de travail suivante : tenir compte de l'effet de la dispersion chromatique sur la distribution statistique du bruit augmente l'efficacité du FEC en présence de dispersion chromatique.

Pour tenter de valider cette hypothèse, il faut comparer les taux d'erreur obtenus grâce à un FEC modifié avec ceux obtenus par un code correcteur d'erreur standard. Faute de moyens, un test expérimental n'est pas possible dans le cadre de la présente recherche. Une simulation Matlab est la solution retenue. Le code de Vivian Issa [ISSA 2007] est utilisé pour recréer virtuellement l'impact de la dispersion chromatique.

### 1.8.2 Contributions

- Établissement d'une métrique pour évaluer l'interférence inter-symbole : le pourcentage d'énergie dans les bits adjacents.
- Proposition d'une méthode alternative pour la simulation du canal, afin de réduire le temps requis pour les simulations.
- Étude de l'influence de la séquence de bits sur la distribution statistique du bruit.
- Proposition d'un ajout à un code correcteur d'erreurs pour tenir compte du caractère déterministe de la dispersion chromatique.

### 1.8.3 Organisation du mémoire

Le deuxième chapitre présente la simulation du canal. L'algorithme utilisé pour créer la dispersion chromatique est expliqué, ainsi que les calculs pour déterminer le pourcentage d'énergie dans les bits adjacents. La présentation d'une méthode alternative pour obtenir une simulation acceptable en un temps raisonnable termine le chapitre.

Le troisième chapitre s'attaque à une étape importante dans l'obtention de résultats : le calcul du taux d'erreur en fonction du rapport signal sur bruit et en fonction de la dispersion chromatique accumulée. Il faut donc ajouter le bruit au modèle de simulation et prendre une décision quant au bit reçu à la sortie du canal, sujets qui seront traités lors de ce chapitre.

Le quatrième chapitre touche quant à lui au décodeur à décision pondérée (SISO, « soft input – soft output) que la présente recherche tente d'améliorer. Le code choisi ainsi que les algorithmes de base seront expliqués. Suivent les résultats de simulation montrant les performances de cet algorithme.

Le cinquième et dernier chapitre porte sur les modifications proposées pour combler les lacunes de l'algorithme de base. Dans un premier temps, une étude permet de constater l'influence de la séquence de bits sur les valeurs à la sortie du canal. Puis, ces résultats sont utilisés pour concevoir un prétraitement améliorant l'efficacité du décodeur.

L'ensemble de ces résultats permettra de conclure quant à la possibilité d'améliorer les codes correcteurs d'erreur en tenant compte du caractère déterministe de la dispersion chromatique et permettra aussi d'énoncer de nouvelles perspectives de recherche.

## CHAPITRE 2

### MODÉLISATION DU CANAL

Le but de ce projet de maîtrise est de proposer une amélioration à un algorithme déjà existant permettant de combattre l'effet de la dispersion chromatique. Pour valider la proposition émise, il faut bien entendu procéder à des simulations. Pour ce faire, il faut d'abord être en mesure de simuler l'effet de la dispersion chromatique sur un signal à transmettre. Le présent chapitre porte donc sur la modélisation du canal qu'est la fibre optique.

#### 2.1 Construction du signal

##### 2.1.1 Format de signalisation

Pour tester l'algorithme proposé, il est d'abord nécessaire d'établir un modèle du signal à l'émetteur. Les systèmes de communication par fibre optique servent généralement à transmettre des signaux numériques binaires, c'est-à-dire des suites de '1' et de '0'. Pour atteindre un certain débit binaire, il faut émettre les bits à intervalle régulier, chacun pendant une période de temps  $T$ . À un débit de 10 Gb/s, la période  $T$  est de 100 ps.

Dans les réseaux de communication par fibre optique, ce sont les formats de modulation RZ (Retour à Zéro) ou NRZ (Non Retour à Zéro) qui sont généralement employés [RAMASWAMI et coll., 2001]. Dans le cas particulier des réseaux de communication régionaux/métropolitains où l'on désire faire migrer le débit de transmission de 2,5 Gb/s à 10 Gb/s, c'est le format de modulation NRZ qui est employé en raison de sa moindre complexité d'implémentation. Dans ce format, les « 1 » sont représentés par un signal de niveau haut constant durant toute la durée du bit alors que les « 0 » sont représentés par un signal de niveau bas durant toute la durée du bit.

Un ajustement est toutefois fait au modèle pour tenir compte d'un aspect pratique : le signal émis par le transmetteur ne peut pas changer instantanément. Le temps de réponse d'un circuit est limité par son inertie due à la mobilité finie des électrons. Il y a une vitesse finie pour la propagation d'un signal. Dans le domaine fréquentiel, cela se caractérise par une fréquence

de coupure plus ou moins haute. Une approximation valable est de représenter l'effet de l'électronique par un filtre passe-bas d'ordre 1. D'après l'expérience pratique d'Alain Houle, directeur de recherche, le temps de montée entre 10% et 90% de la valeur finale correspond à 25% de la période d'un bit, soit 25 ps. Cela correspond à une fréquence de coupure  $f_c$  de 14 GHz. Les calculs nécessaires pour obtenir ce résultat sont présentés à l'annexe I.

La figure 2.1 montre la différence entre la réponse à l'échelon d'un passe-bas d'ordre 1 ayant une fréquence de coupure de 14 GHz et une approximation linéaire où la réponse varie de 0 à 1 en 25 ps. Compte tenu de l'écart relativement faible entre les deux courbes, l'approximation linéaire est retenue par souci de simplicité du modèle.

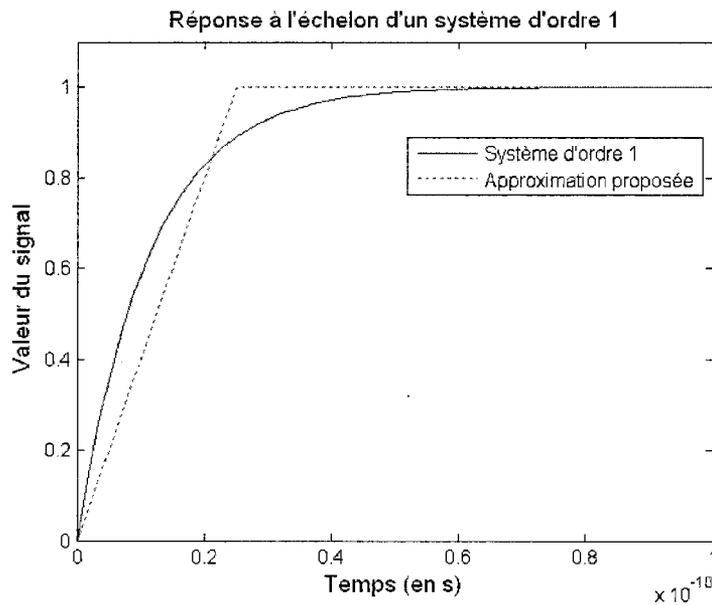


Figure 2.1 – Approximation d'un système d'ordre 1

Ainsi, la période d'un bit est divisée en deux parties. La première partie, établie à 25% de la période totale, est la période de transition. Par exemple, si le bit en transmission est un '1' mais que le bit précédent est un '0', c'est dans la période de changement que la valeur du signal passe graduellement de '0' à '1'. La seconde partie du bit, qui représente 75% du temps total, est la période de stabilité. Dans cette partie, le signal est constant, au niveau du bit à émettre. La figure 2.2 permet de visualiser l'allure temporelle d'un tel signal.

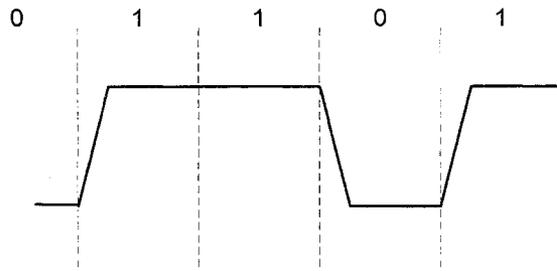


Figure 2.2 – Allure temporelle d'un signal binaire formaté en NRZ

### 2.1.2 Modulation

Une fois le message construit, il faut se demander comment le transmettre. Or, pour pouvoir se propager à l'intérieur de la fibre optique, un signal doit être sous la forme d'une onde lumineuse, préférablement dans la bande C, c'est-à-dire avec une longueur d'onde se situant entre 1530 et 1565 nm. La façon la plus simple est d'utiliser le format de modulation « On-Off Keying » (OOK). Comme le nom l'indique, soit il y a émission de lumière (« On »), soit il n'y en a pas (« Off »). Jumelé avec le format de signalisation NRZ, on obtient le format NRZ-OOK : pour un « 1 », la lumière est injectée dans la fibre et pour un « 0 », aucune lumière n'est injectée dans la fibre.

D'un point de vue mathématique, la lumière émise par le laser est une onde sinusoïdale de fréquence  $f_p$  et d'amplitude  $A$ . Conventionnellement, l'équation  $c = \lambda f$  est utilisée pour établir la relation entre la longueur d'onde et la fréquence dans la fibre. On obtient donc que  $f_p$  varie entre 191,7 THz et 196,1 THz dans la bande C. Toutefois, en réalité, la longueur d'onde  $\lambda$  n'est pas la même dans le vide que dans la fibre, puisque la vitesse de propagation dans la fibre est d'environ  $2/3 c$ . L'emploi de  $c = \lambda f$  est donc une convention.

La modulation OOK est une modulation d'amplitude, une modulation AM. Soit  $m(t)$  le message à transmettre et  $s(t)$  le signal résultant de la modulation, on a :

$$s(t) = m(t) \cdot A \cos(2\pi f_p t + \phi) \quad (2.1)$$

L'allure temporelle d'un tel signal est reproduite à la figure 2.3, considérant la séquence de bits de la figure 2.2. La sinusoïdale porteuse est confinée à l'intérieur d'une enveloppe formée par le message et son négatif. Il est à noter qu'à la figure 2.3,  $f_p$  vaut 50 GHz, comparativement à 193,5 THz pour une longueur d'onde typique de 1550 nm. Cela soulève un problème pratique pour les simulations. À 193,5 THz, le cycle de l'onde porteuse se répète 19350 fois à l'intérieur de la période d'un bit. Puisque les simulations se font dans un univers numérique et non analogique, chaque période doit être divisée en un certain nombre  $L$  d'échantillons. Pour que les résultats soient significatifs, il faudrait donc avoir  $L$  beaucoup plus grand que 19350. Une alternative est nécessaire afin d'amener le temps de simulation à une valeur raisonnable.

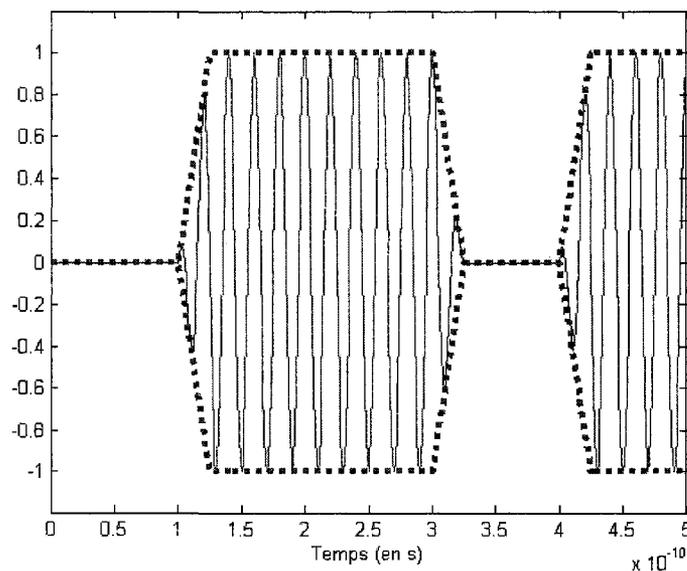


Figure 2.3 – Allure temporelle d'un signal modulé AM

### 2.1.3 Problème pratique : la porteuse

Il faut d'abord se demander s'il est vraiment nécessaire d'inclure la porteuse dans les simulations. En effet, au détecteur à la fin de la fibre optique, c'est l'enveloppe qui est importante, car c'est le message  $m(t)$  qui doit être retrouvé pour ensuite déterminer quelle suite de bits l'a produit. Si la porteuse est supprimée au détecteur, peut-on l'ignorer avant, lors de la simulation du canal ?

Le théorème de la modulation [CORINTHIOS 2001] permet de comprendre l'impact sur le contenu fréquentiel du signal.

Soit  $g(t) = f(t) \cos \omega_p t$  avec  $\omega_p = 2\pi f_p$  la porteuse

alors 
$$G(\omega) = \frac{1}{2} [F(\omega + \omega_p) + F(\omega - \omega_p)] \quad (2.2)$$

La figure 2.4 présente visuellement ce résultat.

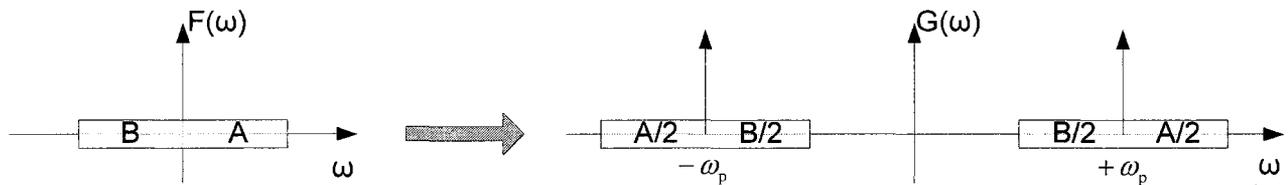


Figure 2.4 – Représentation dans le domaine fréquentiel de l'impact de la modulation AM

Le but est de simuler l'effet de la dispersion chromatique, qui induit un déphasage différent pour chaque composante fréquentielle. Il est donc possible d'utiliser le message  $m(t)$  plutôt que le signal modulé  $s(t)$  si l'on tient compte du décalage en fréquence lors du calcul de l'effet de la dispersion chromatique. Cette possibilité a d'ailleurs déjà été testée et les résultats obtenus sont identiques, que les simulations s'effectuent avec la porteuse ou avec l'enveloppe seulement [ISSA 2007].

## 2.2 Modélisation du canal

### 2.2.1 Idée générale

L'algorithme présenté ci-après permettant de simuler l'effet de la dispersion chromatique est tiré d'un projet de maîtrise réalisé au sein de notre groupe de recherche [ISSA 2007]. L'idée est de modifier le signal transmis dans le domaine fréquentiel, pour rendre compte du fait que les différentes longueurs d'onde ne se propagent pas toutes à la même vitesse dans la fibre optique. La transformée de Fourier et la transformée de Fourier inverse permettent de passer

du domaine temporel au domaine fréquentiel et vice-versa. La figure 2.5 présente un schéma du système.

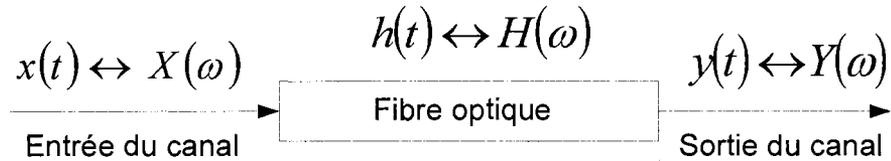


Figure 2.5 – Schéma du système de transmission

$$\text{Pour un tel système, on a : } Y(\omega) = X(\omega)H(\omega) \Leftrightarrow y(t) = x(t)*h(t) \quad (2.3)$$

où  $H(\omega)$  exprime le déphasage à appliquer à chaque composante fréquentielle de l'entrée  $X(\omega)$  en fonction de l'effet de la dispersion chromatique de la fibre optique. Pour obtenir  $y(t)$ , le signal à la sortie du canal, il faut donc faire la transformée de Fourier de  $x(t)$ , multiplier par la fonction de transfert de la fibre optique, puis effectuer la transformée de Fourier inverse du résultat.

### 2.2.2 Numérisation

Or, comme il l'a été mentionné précédemment, les simulations se font dans un univers numérique et non analogique. Chaque période  $T$  est divisée en un certain nombre  $L$  d'échantillons. La fonction  $x(t)$  est donc échantillonnée à toutes les  $T/L$  secondes. Soit  $x_e(n)$  la séquence obtenue suite à l'échantillonnage, alors on a :

$$x_e(n) = x\left(n \frac{T}{L}\right) \quad n \in [1, N] \quad (2.4)$$

où  $N$  : nombre total d'échantillons, ici  $L \times$  nombre de bits transmis simulés

Au lieu de la transformée de Fourier, on utilise la transformée discrète de Fourier, qui se calcule comme suit :

$$X(k) = \sum_{n=0}^{N-1} x_e(n) e^{-j \frac{2\pi}{N} nk} \quad 0 \leq k \leq N-1 \quad (2.5)$$

La figure 2.6 reprend le schéma du système de transmission, mais tel que simulé plutôt que réel. Ainsi, on y retrouve les séquences échantillonnées et les transformées discrètes de Fourier, plutôt que les signaux continus analogiques et les transformées de Fourier.

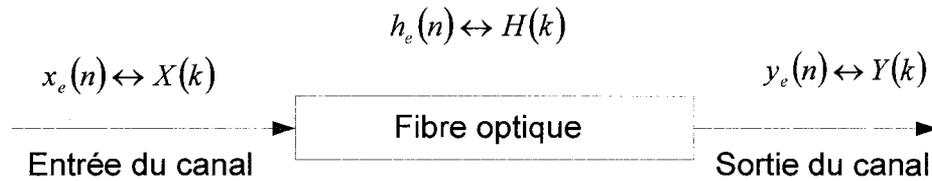


Figure 2.6 – Schéma du système de transmission pour les simulations

### 2.2.3 Théorème d'échantillonnage

Il faut savoir s'il est possible de récupérer de façon intacte le message analogique en utilisant sa version échantillonnée. En d'autres mots, est-ce qu'en remplaçant l'équation  $Y(\omega) = X(\omega)H(\omega)$  par  $Y(k) = X(k)H(k)$ , la séquence  $y_e(n)$  sera équivalente à  $y(t)$  ? Le théorème d'échantillonnage permet de répondre à cette question. Selon ce théorème, pour un signal  $f(t)$  analogique occupant une bande de fréquence comprise dans l'intervalle  $[-\omega_{max}, +\omega_{max}]$ , il est possible de le récupérer de façon intacte en partant de sa version échantillonnée  $f_s(t)$  à la condition que la fréquence d'échantillonnage  $\omega_s$  soit supérieure ou égale à deux fois  $\omega_{max}$ . [CORINTHIOS 2001].

Que vaut ce  $\omega_{max}$  pour un message  $m(t)$  à transmettre par fibre optique ? Plus un signal est rapide dans le temps, plus son spectre est étendu en fréquence. Par conséquent, une suite alternée de '0' et de '1' a le plus large contenu fréquentiel. Le signal analysé, non périodique, est décrit à la figure 2.7.

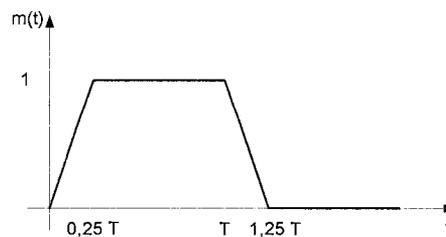


Figure 2.7 – Signal pour le calcul de la transformée de Fourier

Le calcul de la transformée de Fourier est présenté à l'annexe II. Le résultat est le

$$\text{suivant : } M(\omega) = 0.25T \left( \text{sinc} \left( \frac{\omega T}{8} \right) \right)^2 \left[ e^{-j\omega 0.25T} + e^{-j\omega 0.5T} + e^{-j\omega 0.75T} + e^{-j\omega T} \right] \quad (2.6)$$

La figure 2.8 présente la valeur absolue de  $M(\omega)$  pour  $\omega$  entre  $-8\pi/T$  et  $+8\pi/T$ . On remarque que la puissance du signal devient pratiquement nulle pour  $|\omega| > 2,5 \times 10^{11} \text{ rad/s}$ .

Avec  $\omega_{\max} = 2,5 \times 10^{11} \text{ rad/s}$ , prendre  $\omega_s \geq 5 \times 10^{11} \text{ rad/s}$  permet de s'assurer que le message demeure intact.  $f_s = \omega_s / 2\pi = 7,958 \times 10^{10} \text{ Hz}$  et puisqu'il y a  $10^{10}$  bits par seconde, il faut au minimum 7,958 points par bit, donc 8 points ou plus.

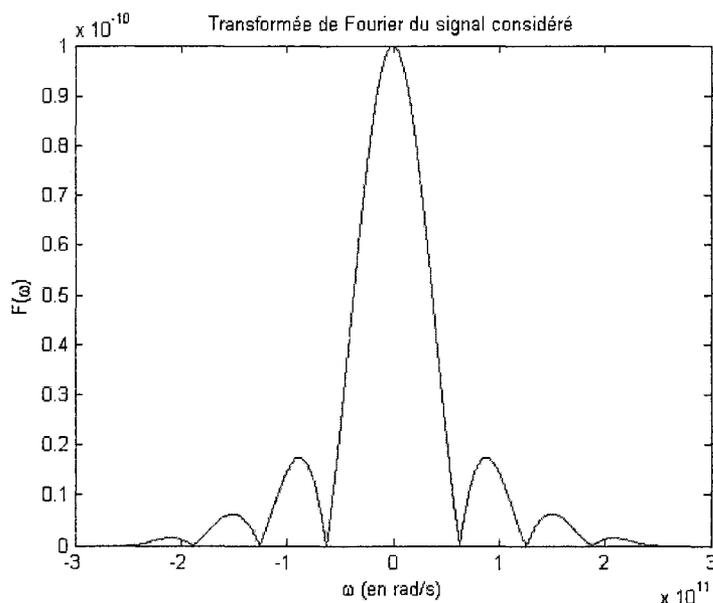


Figure 2.8 – Contenu fréquentiel d'un signal

Pour résumer, en prenant un minimum de 8 échantillons par bit, le théorème d'échantillonnage est respecté en ce qui concerne la partie du spectre où l'on retrouve l'essentiel de la puissance du signal, donc le fait d'utiliser une version discrète du signal plutôt que la version continue n'a pas d'impact significatif sur les résultats. Ainsi, l'équation  $Y(k) = X(k)H(k)$  peut être utilisée.

Bien sûr, le fait de prendre le nombre minimum d'échantillons par bit a un impact sur l'aspect visuel du signal simulé. Il faut comprendre qu'à  $\omega_s = 2\omega_{\max}$ , une sinusoïde de fréquence  $\omega_{\max}$

serait représentée par seulement deux échantillons, donc aurait l'allure d'une onde triangulaire, en faisant une interpolation linéaire entre les points lors de l'affichage. Mais il faut aussi comprendre que chacun des échantillons aurait bel et bien la valeur de la sinusoïde au moment de l'échantillonnage. Et puisque l'on s'intéresse surtout à la valeur des échantillons dans les simulations numériques à effectuer, il est raisonnable de ne prendre seulement que huit échantillons par bit.

#### 2.2.4 Équation de Sellmeier

Une fois cette précision établie, il reste une dernière étape à franchir avant de pouvoir simuler le canal : établir  $H(k)$ , la fonction de transfert de la dispersion chromatique. Il est connu que les différentes longueurs d'onde ne se propagent pas toutes à la même vitesse dans la fibre optique. L'équation de Sellmeier [HOULE 2006] permet d'obtenir le taux de dispersion chromatique pour chacune des longueurs d'onde :

$$D(\lambda) = \frac{\lambda S_0}{4} \left( 1 - \left( \frac{\lambda_0}{\lambda} \right)^4 \right) \quad (2.7)$$

où  $\lambda_0$  : longueur d'onde de référence

$S_0$  : pente de la dispersion chromatique à la longueur d'onde  $\lambda_0$

Pour une fibre de transport typique, ces constantes valent :

$$\lambda_0 = 1310 \text{ nm}$$

$$S_0 = 0,092 \text{ ps} / \text{nm}^2 \cdot \text{km}$$

Le déphasage  $D(\lambda)$  est donné en ps/nm·km. Ainsi, selon le nombre de kilomètres parcourus dans la fibre, l'équation donne, pour chaque longueur d'onde, le déphasage en ps par rapport à la porteuse. Mais comment construire  $H(k)$  à partir de  $D(\lambda)$  ? Puisque  $k$  est un entier entre 0 et  $N-1$ , comment savoir quelle longueur d'onde  $\lambda$  associer à quelle valeur de  $k$  ?

Il est connu que la transformée discrète de Fourier  $X(k)$  est un échantillonnage de la transformée de Fourier  $X(\omega)$ . La formule pour le calcul de la transformée de Fourier est la suivante :

$$X(\omega) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt \quad (2.8)$$

Si l'on calcule la transformée de Fourier de la fonction échantillonnée  $x_e(n)$ , on obtient :

$$X(\omega) = \sum_{n=0}^{N-1} x\left(n \frac{T}{L}\right) e^{-j\omega n \frac{T}{L}} \quad (2.9)$$

puisqu'on sait que  $x_e(n) = x\left(n \frac{T}{L}\right)$  pour  $n$  entier valant entre 0 et  $N-1$  et vaut 0 autrement.

Voici plutôt la formule pour le calcul de la transformée discrète de Fourier :

$$X(k) = \sum_{n=0}^{N-1} x_e(n) e^{-j \frac{2\pi}{N} nk} \quad 0 \leq k \leq N-1 \quad (2.10)$$

Puisque à la fois  $X(k)$  et  $X(\omega)$  doivent représenter le même contenu fréquentiel, il faut avoir :

$$\omega \cdot n \frac{T}{L} = \frac{2\pi}{N} nk \Rightarrow \omega = \frac{2\pi L}{NT} k \quad (2.11)$$

Une fois la relation entre  $k$  et  $\omega$  établie, deux équations simples permettent d'obtenir la relation entre  $k$  et  $\lambda$  :

$$\begin{aligned} \omega &= 2\pi f \\ c &= \lambda f \end{aligned} \quad (2.12)$$

Une dernière transformation s'impose avant de pouvoir construire  $H(k)$ . En effet, le déphasage obtenu par l'équation de Sellmeier est en picosecondes. Or, une fonction de transfert est normalement définie par l'amplitude et la phase en radians. Il serait possible de calculer l'équivalence entre picosecondes et radians, mais ce qui suit explique plutôt comment transformer l'équation de la transformée inverse pour utiliser les picosecondes.

Voici l'équation de la transformée de Fourier discrète inverse :

$$x(n) = \frac{1}{N} \sum_{k=1}^N X(k) e^{-j \frac{2\pi kn}{N}} \quad 0 \leq n \leq N-1 \quad (2.13)$$

Pour calculer la valeur du signal à la sortie du canal, il faut modifier cette équation ainsi :

$$y(n) = \frac{1}{N} \sum_{k=1}^N H(k) M(k) e^{-j \frac{2\pi kn}{N}} \quad 0 \leq n \leq N-1 \quad (2.14)$$

Cette équation ne permet pas d'introduire directement un déphasage en ps. La phase, qui est l'exposant de l'exponentielle, n'a pas d'unité :  $k$  indique le numéro de la fréquence par rapport aux  $N$  fréquences obtenues, et  $n$  indique le numéro de l'échantillon temporel. Mais puisque

$\omega = \frac{2\pi L}{NT} k$  et que  $\omega = 2\pi f$ , on peut remplacer  $\frac{k}{N}$  par  $f(k) \frac{T}{L}$ , où  $f(k)$  est une fonction associant  $k$  à une fréquence. L'unité de  $f(k)$  est  $s^{-1}$ , donc il faut multiplier par une valeur en secondes. Or, on sait qu'aux instants d'échantillonnage,  $t = n \frac{T}{L}$ . On peut donc obtenir  $t(n)$ , le temps associé à chaque numéro d'échantillon.

Voici donc l'équation finale de l'algorithme proposé :

$$y(n) = \frac{1}{N} \sum_{k=1}^N M(k) e^{-j 2\pi f(k) \phi(k)} e^{j 2\pi f(k) t(n)} = \frac{1}{N} \sum_{k=1}^N M(k) e^{j 2\pi f(k) (t(n) - \phi(k))} \quad (2.15)$$

où  $y(n)$  : version échantillonnée du signal à la sortie du canal

$M(k)$  : composantes fréquentielles du message à l'entrée du canal

$\phi(k)$  : déphasage (en secondes) calculé par l'équation de Sellmeier

$f(k)$  : fonction associant  $k$  à une fréquence

$t(n)$  : fonction associant à l'échantillon  $n$  le temps correspondant

### 2.3 Résultats de simulation

Il faut maintenant voir quels résultats produit l'algorithme en utilisant le modèle présenté pour le signal. Mais d'abord, il faut se poser quelques questions. Le signal simulé représente-t-il la

puissance de la lumière ou plutôt son champ électrique ? Quel est l'impact du récepteur sur le signal ? Et dans l'équation de la transformée discrète de Fourier, l'un des paramètres est le nombre de bits utilisés pour les calculs. Quel est l'impact de ce paramètre sur les résultats ?

Physiquement, la lumière est une onde électromagnétique, et c'est sur cette onde électromagnétique que la dispersion chromatique a un impact. Le signal simulé est donc l'enveloppe du champ électrique produit par le laser. Mais au photodétecteur, à la sortie du canal, c'est plutôt la puissance de la lumière qui est détectée et transformée en signal électronique. Pour obtenir, à une constante près, la puissance à partir du champ électrique  $\vec{E}$ , il faut multiplier  $\vec{E}$  par son complexe conjugué. Et puisque c'est à partir du signal électronique que les décisions sont prises, il est important d'en tenir compte lors de la présentation des résultats de simulation.

Parlant de signal électronique, il ne faut pas oublier que tout comme le transmetteur, le récepteur est constitué de composantes électroniques et que par conséquent, sa bande passante est limitée. Une bonne approximation est de représenter l'électronique du récepteur par un passe-bas d'ordre 1 ayant une fréquence de coupure de 7 GHz. Ce choix n'est pas arbitraire : c'est la bande passante minimale que l'on retrouve dans plusieurs récepteurs sur le marché [AC6539], [ERM568RLC].

Il serait techniquement possible d'obtenir une bande passante plus large, mais cela ne serait pas nécessairement souhaitable. Le but visé est que la bande passante soit suffisamment large pour laisser passer le signal mais assez restreinte pour couper le plus de bruit possible. Or, une suite infinie de « 1 » et de « 0 » constitue le signal binaire possédant le plus grand contenu fréquentiel. À 10 Gb/s, cette suite infinie correspond à une onde carrée dont la fréquence est de 5 GHz. Par conséquent, la majeure partie de l'énergie du signal se trouvera à 5 GHz, ce qui est correctement restitué par un filtre à 7 GHz. La partie restante de l'énergie se trouve aux harmoniques impaires et sera atténuée. Toutefois, puisque la contribution principale des harmoniques est de diminuer la période de transition du signal et que la décision sur la séquence de bits se fait plutôt dans la période de stabilité du signal, l'impact sur la décision sera minime.

Pour simuler numériquement un tel filtre, une approche possible consiste à regarder sa réponse en fréquence. Soit  $|H(j\omega)|$  le gain et  $\angle H(j\omega)$  la phase à la fréquence  $\omega$ . Alors pour

$$\text{un filtre passe-bas d'ordre 1, on a : } |H(j\omega)| = \frac{1}{\sqrt{1 + \left(\frac{\omega}{\omega_c}\right)^2}} \quad (2.16)$$

Pour  $\angle H(j\omega)$ , il est important que le choix n'entraîne pas de la dispersion. Pour que ce soit le cas, le filtre doit être linéaire en phase. Un filtre linéaire en phase introduit un délai qui est proportionnel à la pente de la fonction linéaire. Idéalement, pour faciliter la synchronisation au récepteur, le délai doit être nul. C'est le cas pour un filtre linéaire particulier : le filtre à phase nulle. Ainsi, il faut prendre :

$$\angle H(j\omega) = 0 \quad (2.17)$$

Soit  $X(\omega)$  le contenu fréquentiel à l'entrée du filtre et  $Y(\omega)$  à la sortie, alors :

$$Y(\omega) = X(\omega) |H(j\omega)| e^{j\angle H(j\omega)} \quad (2.18)$$

Pour ce qui est de l'impact du nombre de bits nb utilisés pour les calculs, il faut regarder à nouveau la relation entre  $\omega$  et  $k$  :

$$\omega = \frac{2\pi L}{NT} k \quad (2.19)$$

Puisque  $k$  est un entier,  $2\pi L/NT$  est l'incrément entre chaque fréquence du spectre obtenu. Or,  $N = L \times \text{nb bits}$ . En augmentant  $L$ , le nombre d'échantillons par bit, l'incrément n'est pas touché, mais en augmentant le nombre de bits, l'incrément diminue. Cela signifie que plus la séquence simulée contient de bits, plus les fréquences considérées pour les calculs sont rapprochées les unes des autres. Il faut donc optimiser le paramètre  $L$  pour avoir la meilleure précision possible en un temps de simulation raisonnable.

Mais pour le moment, il n'est pas nécessaire d'effectuer cette optimisation, puisqu'il s'agit de vérifier si les résultats de l'algorithme sont valides. Ainsi, le temps de simulation n'est pas encore important. Une bonne façon de visualiser l'impact de la dispersion chromatique est le

diagramme d'œil, une superposition du signal reçu sur une période d'un bit. La figure 2.9 présente différents diagrammes d'œil obtenus grâce à l'algorithme présenté, en utilisant 8 échantillons par bit avec une simulation de 200 bits. Pour ces simulations, l'impact du récepteur a été considéré (multiplication par le complexe conjugué et filtre passe-bas 7 GHz).

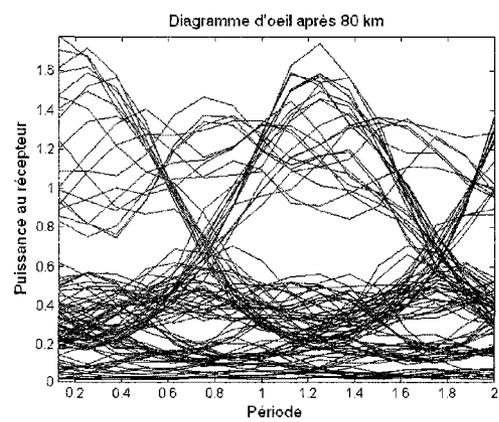
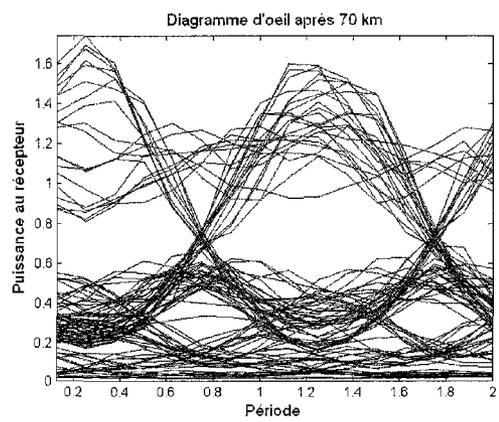
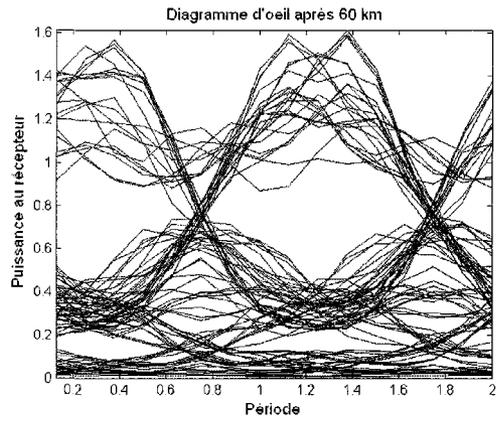
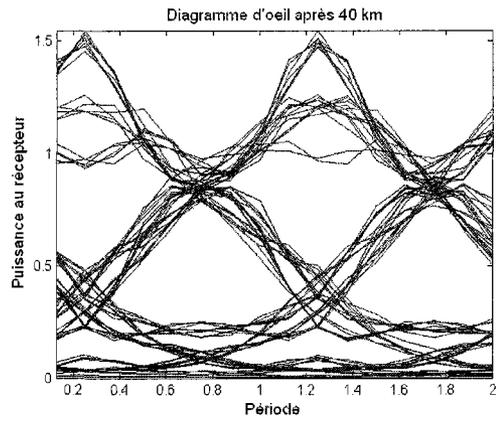
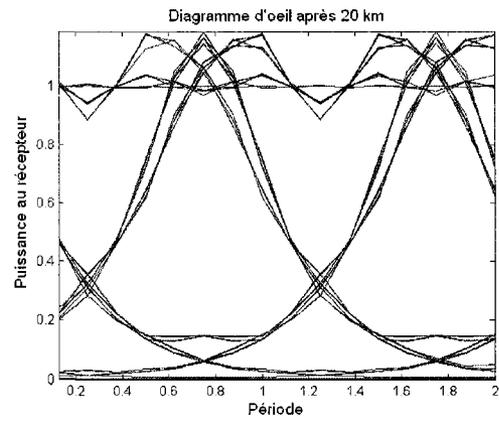
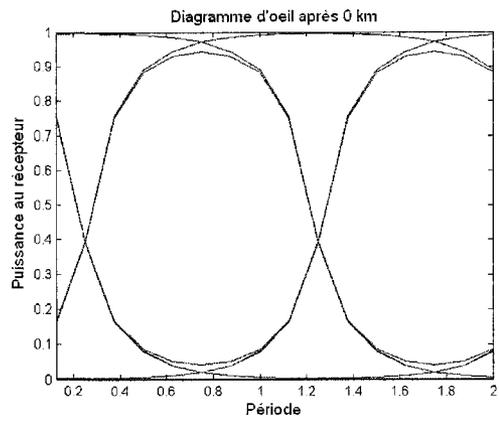


Figure 2.9 – Diagrammes d'oeil obtenus avec l'algorithme présenté

Il est clair en observant les diagrammes d'œil de la figure 2.9 que les signaux simulés ont subi de la dispersion, c'est-à-dire un élargissement temporel des impulsions. De plus, la dispersion paraît plus importante après 80 km qu'après 60 ou 40 km, ce qui concorde avec le phénomène de la dispersion chromatique. Mais peut-on affirmer que ces simulations fournissent une approximation valable de la réalité ? Comment savoir si les résultats obtenus après 60 km en simulation sont représentatifs de ce qu'on obtient après 60 km de véritable fibre optique ? En pratique, le signal obtenu dépend de nombreux facteurs : caractéristiques du modulateur, de la fibre, du laser, ... Ainsi, d'une expérimentation à l'autre, le diagramme d'œil n'est jamais exactement le même. Mais typiquement, la limite de propagation est de 1000 km à 2,5 Gb/s, de 80 km à 10 Gb/s et serait de 5 km à 40 Gb/s. Obtient-on les mêmes limites avec le modèle présenté ? La prochaine section tentera de répondre à cette question grâce au calcul du pourcentage d'énergie dans les bits adjacents, qui servira aussi à mesurer l'interférence inter-symbole.

## **2.4 Mesure de l'interférence inter-symbole**

L'interférence inter-symbole causée par la dispersion chromatique est visible dans les diagrammes d'œil de la section précédente. Mais comment la mesurer ? Comment chiffrer l'effet produit par la dispersion chromatique ? La mesure proposée peut-elle prédire la limite de propagation dans la fibre ? C'est à ces questions que s'intéresse la présente section.

### **2.4.1 Proposition**

Par définition, l'interférence inter-symbole signifie que le signal reçu pendant la période d'un bit donné dépend des autres bits émis. Cela signifie aussi qu'un bit transmis influence la valeur reçue pour les bits environnants, bref, qu'une partie de l'énergie émise pour la transmission du bit déborde de la période de temps  $T$  qui lui est normalement allouée. Cela mène à la proposition suivante : mesurer l'interférence inter-symbole en calculant le pourcentage d'énergie dans les bits adjacents. En d'autres mots, par rapport à l'énergie totale d'un bit, calculer quelle proportion se retrouve à l'extérieur de la période  $T$ .

La façon la plus simple de faire le calcul est de considérer l'envoi d'un '1' logique précédé et suivi de '0'. Puisque le format de modulation est OOK, l'énergie présente peut uniquement provenir du '1', puisqu'un '0' correspond à l'absence de lumière dans la fibre optique. Ainsi, mesurer la répartition temporelle de l'énergie à la sortie du canal permet de calculer le pourcentage d'énergie dans les bits adjacents. La figure 2.10 présente l'aspect temporel du signal proposé. Il est à noter que même sans dispersion chromatique, le pourcentage d'énergie dans les bits adjacents n'est pas nul, à cause de la période de changement au début de chaque bit.

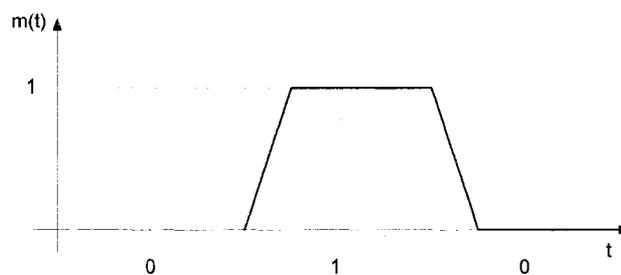


Figure 2.10 – Signal proposé pour la mesure de l'interférence inter-symbole

Lorsque la moitié de l'énergie se retrouve dans les bits adjacents, comment déterminer de quel bit provient cette énergie ? Cette question mène à l'hypothèse suivante : la distance limite de propagation est atteinte lorsque le pourcentage d'énergie dans les bits adjacents vaut 50%.

#### 2.4.2 Méthode de calcul

Par définition, l'énergie est l'intégrale de la puissance par rapport au temps. Quant à la puissance, elle vaut, à une constante près, le champ électrique multiplié par son complexe conjugué. Cette constante s'annule d'ailleurs dans les calculs, puisque l'on recherche le rapport entre l'énergie dans les bits adjacents et l'énergie totale.

Soit  $v_e(n)$  le champ électrique obtenu à la sortie du canal. Chacun des échantillons de  $v_e(n)$  est un nombre complexe, noté  $v_e(n) = a_n + b_n i$ . En multipliant chaque échantillon par son complexe conjugué, on obtient la puissance instantanée à chacun des instants :

$$P(n) = (a_n + b_n i)(a_n - b_n i) = a_n^2 + b_n^2 \quad (2.20)$$

Il faut aussi considérer l'impact de l'électronique du récepteur, qui limite la bande passante du signal à 7 GHz. Ainsi, appelons  $P_f(n)$  la puissance instantanée à la sortie du filtre d'ordre 1.

L'énergie est l'intégrale de la puissance par rapport au temps. Une interpolation linéaire entre les échantillons de puissance instantanée permet d'obtenir une fonction temporelle. Pour un test sur N échantillons, l'énergie totale  $E_{tot}$  vaut donc :

$$E_{tot} = \sum_{n=0}^{N-1} \left( \frac{P_f(n) + P_f(n+1)}{2} \right) T \quad (2.21)$$

L'énergie contenue à l'intérieur de la période réservée au bit ( $E_{bit}$ ) se calcule comme suit :

$$E_{bit} = \sum_{n=n_1}^{n_2} \left( \frac{P_f(n) + P_f(n+1)}{2} \right) T \quad (2.22)$$

Finalement, le but est d'obtenir le pourcentage d'énergie dans les bits adjacents :

$$\%E_{adj} = \left( 1 - \frac{E_{bit}}{E_{tot}} \right) \times 100\% \quad (2.23)$$

Le détail des calculs est présenté à l'annexe III.

### 2.4.3 Résultats

Avant d'effectuer les premiers tests, il faut d'abord déterminer la valeur de  $n_1$  et  $n_2$ , les échantillons qui correspondent au début et à la fin de la période réservée au bit '1'. D'après le format proposé, le temps réservé à un bit commence au début de la période de changement et se termine à la fin de la période de stabilité. Or, ce découpage n'est pas optimal au moment de la décision. Il est préférable qu'un maximum d'énergie soit initialement à l'intérieur de la période. Pour ce faire, le début de la période réservée au bit est fixé au milieu de la période de changement. La figure 2.11 illustre ce changement.

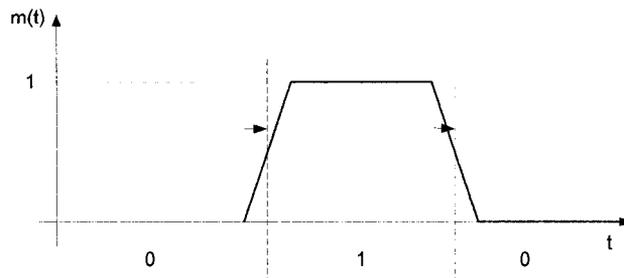


Figure 2.11 – Modification de la période du bit

La figure 2.12 montre que cet ajustement permet effectivement de centrer la période pour que le maximum d'énergie soit à l'intérieur du temps réservé à un bit. La partie en trait gras correspond à la période réservée au bit '1'.

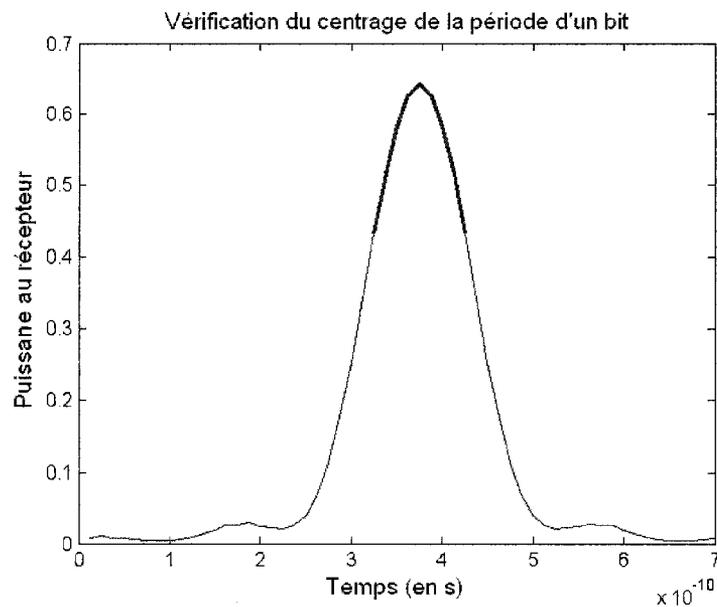


Figure 2.12 – Confirmation du centrage de la période d'un bit

Deux autres paramètres doivent aussi être ajustés : le nombre de points par bit ( $L$ ) et le nombre de '0' avant et après le '1'. L'impact du nombre de '0' est double. D'abord, puisque l'interférence inter-symbole fait déborder l'énergie d'un bit sur les bits adjacents, il faut s'assurer qu'il y a suffisamment de '0' pour absorber ce débordement. Puis, comme il en a été discuté précédemment, plus la séquence simulée contient de bits, plus les fréquences

considérées pour les calculs sont rapprochées les unes des autres. Par conséquent, plus il y a de '0', plus le résultat de simulation est précis. Quant au nombre de points par bit, il faut vérifier son influence à cause de l'interpolation linéaire effectuée entre les points. À la fréquence limite du théorème d'échantillonnage, une sinusoïde a l'allure d'une onde triangulaire. Il faut donc vérifier si 8 points par bits sont suffisants. La figure 2.13 permet de visualiser l'effet conjoint de ces deux paramètres. Chaque courbe présente, pour un certain nombre de '0' ajoutés avant et après, le pourcentage d'énergie dans les bits adjacents en fonction du nombre de points par bit. Les simulations ont été effectuées pour une distance de transmission de 70 km.

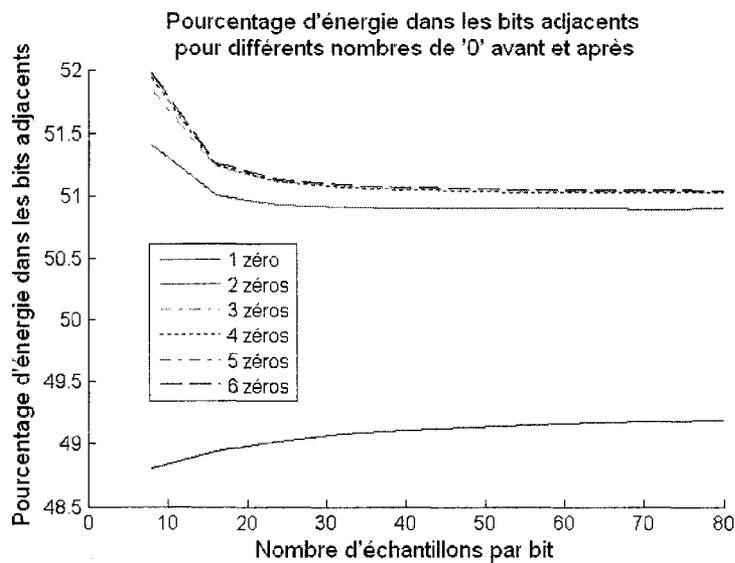


Figure 2.13 – Ajustement des paramètres de simulation

Puisque, d'après la figure 2.13, les courbes se confondent à partir de trois '0' ajoutés, un bon choix semble être d'ajouter trois '0' avant et après. Pour ce qui est de l'impact du nombre de points par bit, il est faible (de l'ordre de 1,5%), mais il est intéressant de regarder pourquoi. La figure 2.14 superpose les signaux obtenus à 8 points par bit et à 32 points par bit. Les signaux sont très similaires, sauf aux extrémités, où l'on remarque que le signal à 32 points par bit a une valeur légèrement plus faible que celui à 8 points par bit. Cela explique pourquoi le pourcentage d'énergie dans les bits adjacents est plus élevé à 8 points par bit. Ainsi, pour obtenir les résultats les plus précis possibles, les simulations pour le calcul du pourcentage d'énergie dans les bits adjacents seront exceptionnellement effectuées à 32 points par bits.

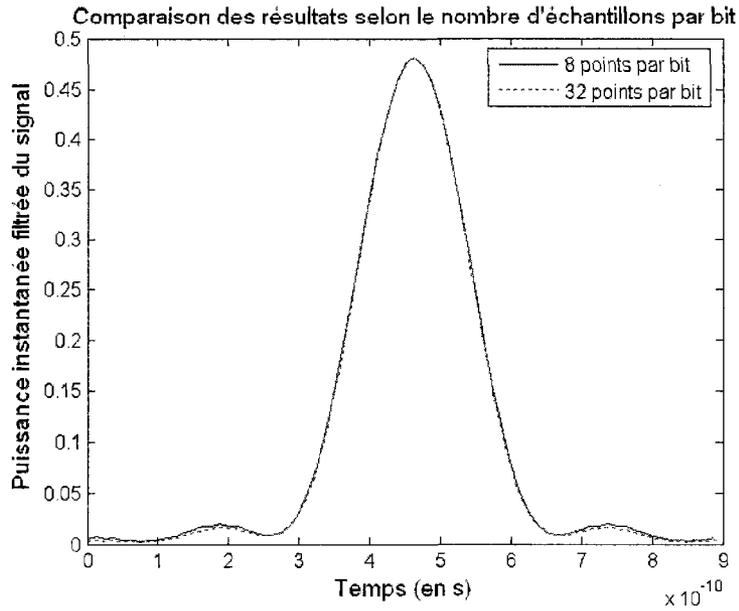


Figure 2.14 – Impact du nombre de points par bit

Une fois ces paramètres ajustés, il est possible d'effectuer le test désiré : le calcul du pourcentage d'énergie dans les bits adjacents en fonction de la distance parcourue dans la fibre. La figure 2.15 présente les résultats obtenus pour un débit de 10 Gb/s.

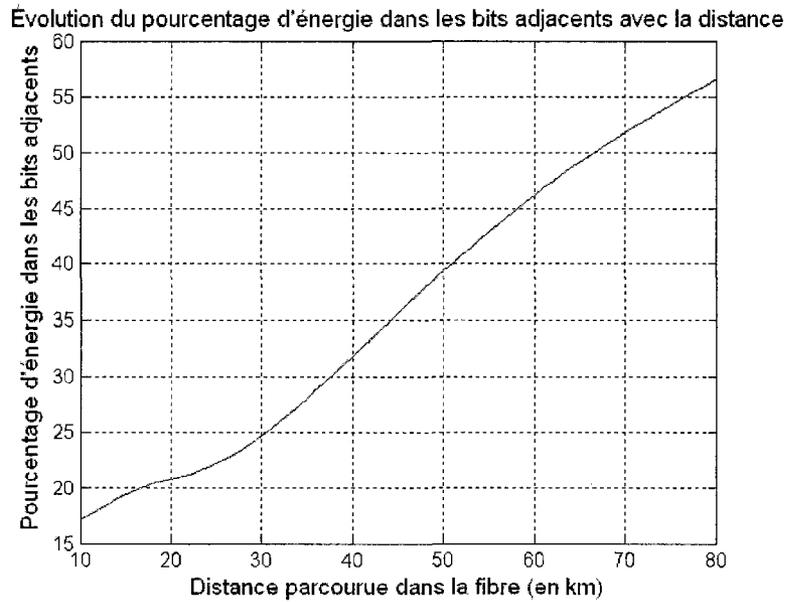


Figure 2.15 – Pourcentage d'énergie dans les bits adjacents en fonction de la distance

Le seuil de 50% d'énergie dans les bits adjacents est atteint après environ 66 km dans la fibre, ce qui est légèrement inférieur à la limite de propagation sans compensation à 10 Gb/s. Pour effectuer le même test à 2,5 Gb/s et à 40 Gb/s, il est nécessaire d'ajuster la bande passante du filtre passe-bas au récepteur. Pour garder la même proportion, la fréquence de coupure est fixée à 0,7 fois le débit binaire. Ainsi, la bande passante est de 1,75 GHz pour la transmission à 2,5 Gb/s et de 28 GHz pour la transmission à 40 Gb/s.

Suite à cet ajustement, le seuil de 50% est atteint après 1065 km à 2,5 Gb/s tandis qu'à 40 Gb/s, il est atteint après 4,1 km. Seul le test à 2,5 Gb/s semble correspondre à la limite de propagation atteinte en pratique. Toutefois, c'est généralement le piaillement (*chirp* en anglais) qui ramène la limite de propagation à 1000 km à 2,5 Gb/s. Or, le modèle de simulation utilisé ici implique une modulation sans piaillement, ce qu'on obtient en utilisant un modulateur Mach-Zender. À 2,5 Gb/s, le modulateur Mach-Zender est rarement utilisé vu son coût plus élevé. Ainsi, la limite de propagation devrait être supérieure à 1000 km en simulation. D'un autre côté, les résultats montrent qu'il y a bel et bien un effet dispersif, puisqu'ils sont inversement proportionnels au carré du débit de transmission. En effet, la distance où le seuil est atteint est 16 fois plus grande à 10 Gb/s qu'à 40 Gb/s et est également 16 fois plus grande à 2,5 Gb/s qu'à 10 Gb/s.

Deux conclusions sont possibles pour expliquer la différence entre les distances obtenues et les limites standards de propagation : soit la limite n'est pas atteinte à 50% d'énergie dans les bits adjacents, soit le modèle du canal est plus dispersif que les fibres optiques déployées. Cela pourra être confirmé lors du calcul du taux d'erreur au prochain chapitre. Finalement, il ne faut pas oublier l'objectif du projet de recherche : proposer une amélioration à un code correcteur d'erreur dans le contexte d'un milieu de propagation dispersif. Ainsi, même si la seconde conclusion s'avérait vraie et que le modèle était effectivement plus dispersif qu'une fibre optique standard, cela n'affecterait pas la validité des résultats, à savoir si oui ou non, la modification proposée est efficace.

Avant de s'attaquer à la simulation du taux d'erreur, il faut s'intéresser à un dernier aspect pratique important : le temps de simulation. La prochaine section s'attarde à ce sujet.

## 2.5 Temps de simulation

Les résultats obtenus portent à croire que l'algorithme permet de bien simuler l'effet de la dispersion chromatique et qu'il peut donc être utilisé pour les simulations du canal. Or, puisque le code correcteur d'erreurs choisi est celui utilisé dans [OUCHI et coll., 2006], soit le BCH(144,128) x BCH(256,239), cela signifie que les simulations se font par blocs de  $144 \times 256 = 36864$  bits. Le temps de simulation devient alors un paramètre important à considérer.

### 2.5.1 Problématique du temps de simulation

L'algorithme proposé fonctionne bien, mais les calculs requis sont complexes et coûteux en temps de calcul. Par exemple, le calcul de la transformée de Fourier inverse implémenté a une complexité de l'ordre de  $2n^2$ . Il est donc impensable de simuler les 36864 bits en une seule séquence. Par exemple, à 8 échantillons par bit, une séquence de 128 bits demande 32s de temps de calcul, tandis qu'une séquence de 256 bits demande 242s. Il faut donc trouver une façon de réduire le temps de simulation.

### 2.5.2 Réduction du temps de simulation

La première façon de réduire le temps de simulation est de diviser la séquence de 36864 bits en plus petites séquences. En effet, il est plus rapide de simuler deux fois 128 bits qu'une fois 256 bits. Par contre, il faut s'assurer que les résultats de simulation demeurent valables malgré cette modification.

L'idée est de trouver le meilleur compromis entre rapidité d'exécution et exactitude des simulations. Plusieurs choix sont possibles pour la division des 36864 bits en plus petites séquences. La meilleure combinaison trouvée est de simuler 16 bits à la fois tout en considérant les 14 bits précédents et les 14 bits subséquents. La démarche suivie pour obtenir ce résultat est présentée à l'annexe 4. Pour 36864 bits, la simulation dure 2957,4 secondes avec ce choix, soit 49 minutes, et l'erreur est faible.

### 2.5.3 Méthode alternative

Le temps obtenu en divisant la longue séquence en plus petites est raisonnable, mais il est possible de réduire davantage le temps de calcul en modifiant légèrement la méthode de simulation de la dispersion chromatique.

L'idée proposée repose sur le théorème de convolution [CORINTHIOS 2001] :

$$\begin{aligned} \text{Si} \quad & f_1(t) \leftrightarrow F_1(\omega) \\ & f_2(t) \leftrightarrow F_2(\omega) \end{aligned} \quad (2.24)$$

$$\text{alors} \quad f_1(t) * f_2(t) \rightarrow F_1(\omega)F_2(\omega)$$

En d'autres mots, la convolution de deux fonctions dans le domaine temporel produit le même contenu fréquentiel que la multiplication de leur spectre respectif.

Pour le système à l'étude, soit la modélisation de la dispersion chromatique, la sortie vaut :  $Y(\omega) = X(\omega)H(\omega)$ . Donc d'après le théorème de convolution :

$$y(t) = F^{-1}[Y(\omega)] = x(t) * h(t) \quad (2.25)$$

À la section 2.2.3, il a été énoncé que si le théorème d'échantillonnage est respecté, alors la version numérisée peut se substituer à la version analogique sans perte d'information. Ainsi, il est possible de prendre l'équation suivante :  $y_e(n) = x_e(n) * h_e(n)$ .

L'avantage : la convolution, sur Matlab, est une opération plus rapide que la transformée de Fourier inverse. De plus, on évite d'avoir initialement à calculer la transformée de Fourier du signal. Il suffit de calculer une fois la transformée de Fourier inverse de la fonction de transfert  $H(k)$  et d'utiliser la réponse impulsionnelle  $h_e(n)$  obtenue comme un filtre.

Pour s'assurer de la validité de la méthode, un test a été effectué. Dans un premier temps, une séquence de 36864 bits est simulée en utilisant la combinaison retenue, soit 16 bits simulés à la fois en considérant les 14 bits précédents et les 14 bits suivants. Pour chaque section de 16 bits, la simulation utilise une fonction de transfert  $H(k)$  ayant 352 éléments (8 points par bit x (16+2x14)bits). En utilisant la réponse impulsionnelle  $h_e(n)$  équivalente plutôt que  $H(k)$  et en filtrant la séquence complète, la simulation dure seulement 2,7 secondes. La différence maximale entre les deux simulations est de 0,0084 et la différence moyenne est de

$8,1316 \times 10^{-4}$ . Vu le gain de temps considérable et la similitude entre les deux résultats, la méthode alternative sera utilisée pour les simulations subséquentes.

Il reste maintenant à déterminer la longueur optimale du filtre à utiliser. Combien de bits doit-on considérer simultanément pour obtenir le meilleur compromis entre la précision des résultats et le temps de simulation ? Pour le déterminer, le test précédent est repris, mais cette fois en variant la longueur du filtre, c'est-à-dire le nombre de bits utilisés dans le calcul de la fonction de transfert. La figure 2.16 présente les résultats d'un tel test.

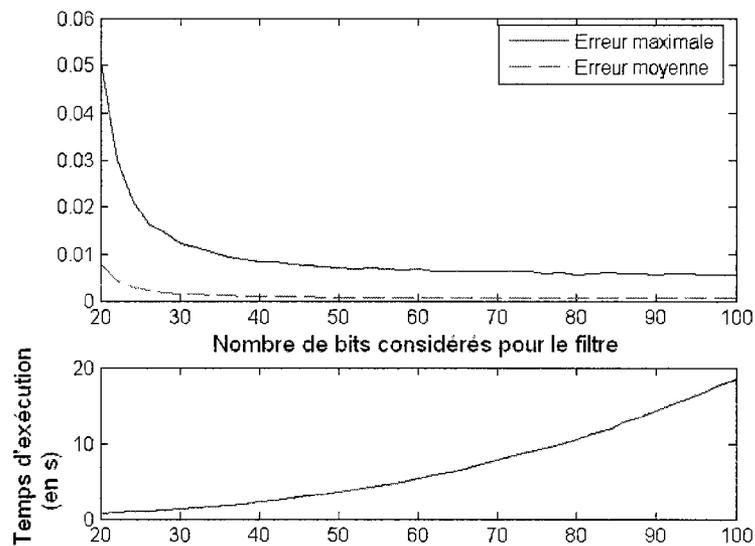


Figure 2.16 – Optimisation de la longueur du filtre pour simuler la dispersion chromatique

À la figure 2.16, on remarque qu'après environ 30-40 bits, l'erreur diminue peu mais le temps d'exécution de la simulation augmente de plus en plus. Si l'on décide que l'erreur maximale tolérable sur les résultats de simulation est de 0,01, alors toutes les simulations pour lesquelles l'erreur maximale est supérieure à 0,01 sont rejetées. La simulation où 36 bits sont considérés est la plus rapide parmi les possibilités restantes. Avec 36 bits, l'erreur maximale est de  $9,5 \times 10^{-3}$ , l'erreur moyenne est de  $1,05 \times 10^{-3}$  et le temps d'exécution est de 1,89 secondes. C'est donc cette option qui est retenue.

Il est également intéressant d'effectuer la même optimisation pour le filtre d'ordre 1 à la sortie du récepteur. La figure 2.17 présente les résultats de ce test. D'abord, on remarque que le temps de simulation est trop court pour être mesuré adéquatement. Pourquoi est-ce le cas ici, mais pas pour l'obtention du filtre pour la dispersion chromatique ? Tout simplement parce qu'ici, la fonction *ifft* de Matlab est utilisée plutôt que d'effectuer le calcul de la transformée de Fourier inverse par une double boucle *for*. Cela réduit considérablement le temps de calcul.

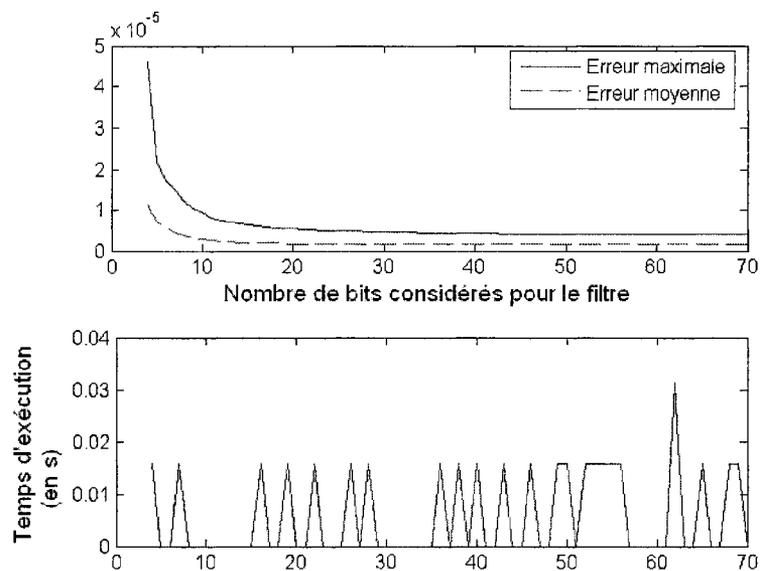


Figure 2.17 – Optimisation de la longueur du filtre d'ordre 1 au récepteur

Deuxième observation d'importance : dans tous les cas, l'erreur maximale et l'erreur moyenne sont très faibles, de l'ordre de  $10^{-5}$ . Ainsi, n'importe quel choix testé est satisfaisant. Toutefois, puisqu'il faut prendre une décision, un filtre de 12 bits sera choisi.

## 2.6 Résumé

La simulation du canal est la première étape à franchir pour pouvoir tester un algorithme de correction d'erreur. Le signal à transmettre est modélisé par deux périodes : la période de transition (25% de la période totale) où il transite linéairement d'une valeur à une autre, et la période de stabilité (75% de la période totale) où il maintient sa valeur. Ce signal est modulé

suivant le format NRZ-OOK, mais la modulation par la porteuse, la lumière émise par le laser, n'est pas incluse dans les simulations, car son omission n'affecte pas les résultats.

Les simulations se font dans un environnement numérique. D'après le théorème d'échantillonnage, huit échantillons par bit sont suffisants pour représenter le signal. L'algorithme reproduisant l'effet de la dispersion chromatique repose sur l'équation de Sellmeier, qui permet d'obtenir la réponse en fréquence à appliquer à la transformée de Fourier du signal. Le théorème de convolution amène une méthode alternative qui réduit grandement le temps de simulation nécessaire.

Il faut aussi se préoccuper de l'impact de l'électronique au récepteur. Le signal doit être multiplié par son complexe conjugué puisque c'est la puissance qui est détectée et non le champ électrique. La bande passante limitée de l'électronique est représentée par un filtre passe-bas d'ordre 1.

Une fois ces considérations réglées, il est possible d'obtenir les diagrammes d'œil à la sortie du canal, diagrammes montrant l'effet dispersif subi par le signal. Le calcul du pourcentage d'énergie dans les bits adjacents est proposé comme mesure de l'interférence inter-symbole causée par la dispersion chromatique. Une hypothèse est posée : la limite de propagation est atteinte lorsque 50% de l'énergie se retrouve dans les bits adjacents. Pour le vérifier, il faut s'intéresser au taux d'erreur, sujet qui fait l'objet du prochain chapitre.

## CHAPITRE 3

### CALCUL DU TAUX D'ERREUR

Le précédent chapitre a permis de comprendre comment simuler numériquement l'effet de la dispersion chromatique sur un signal binaire. C'est une partie importante du travail de simulation, mais il ne faut pas en oublier le but : mesurer les performances du canal qu'est la fibre optique. Le présent chapitre expliquera donc comment obtenir le taux d'erreur en fonction du rapport signal sur bruit et de la distance.

Mais avant de parler d'erreur, il faut d'abord déterminer comment prendre la décision à la sortie du canal : le bit reçu est-il un '1' ou un '0' ? Puisque l'on souhaite obtenir le taux d'erreur en fonction du rapport signal sur bruit, il faut également se demander quelles sont les sources de bruit du système et comment les modéliser pour les ajouter au signal. Finalement, comme lors du chapitre 2, il faut se soucier du temps de simulation pour obtenir de bons résultats en un temps raisonnable.

#### 3.1 Prise de décision

##### 3.1.1 Intégration au récepteur

À la sortie du canal, au détecteur, il faut déterminer si le signal reçu correspond à un '1' ou à un '0'. La décision ne se prend toutefois pas en un instant précis. Typiquement, au récepteur, un système électronique calcule la moyenne du signal pendant une certaine période de temps. Selon l'expérience du professeur Alain Houle, cela correspond à 25% de la période, centré sur le milieu du bit. La figure 3.1 permet de voir que pour une période coupée en 8 échantillons, la décision se prend en considérant le signal entre les échantillons 5 et 7.

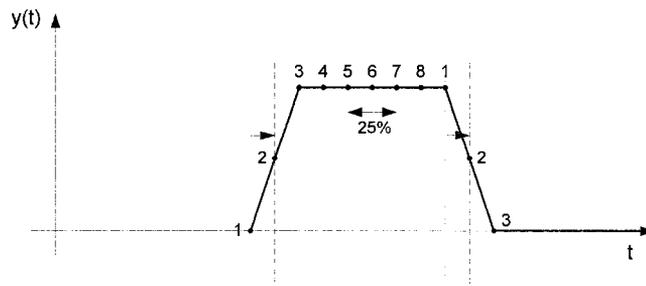


Figure 3.1 – Visualisation de la période de décision

Le calcul se fait suite à la conversion, au récepteur, de la puissance optique en courant électrique. Il ne faut donc pas oublier de multiplier le signal à la sortie du canal par son complexe conjugué, puis d'appliquer le filtre passe-bas d'ordre 1 à 7 GHz de fréquence de coupure. La variation du signal entre les points obtenus est considérée linéaire.

La moyenne se calcule en intégrant la puissance à l'intérieur de la période considérée, puis en divisant par l'intervalle de temps. Soit  $M(k)$  la moyenne pour le bit  $k$ , alors :

$$M(k) = \sum_{n=n_1}^{n_2-1} \frac{P_n(k) + P_{n+1}(k)}{2(n_2 - n_1)} \quad (3.1)$$

avec  $P_n(k)$  : puissance au récepteur pour l'échantillon  $n$  du bit  $k$   
 $n_1$  : échantillon correspondant au début de l'intégration  
 $n_2$  : échantillon correspondant à la fin de l'intégration

Pour le cas qui nous intéresse, soit l'intégration de 25% de la période, centré sur le milieu du bit avec 8 échantillons par bit, alors l'équation devient :

$$M(k) = \frac{0.5P_5(k) + P_6(k) + 0.5P_7(k)}{2} \quad (3.2)$$

### 3.1.2 Seuil de décision

Le calcul précédent permet d'établir la valeur moyenne du signal pour chacun des bits. Parmi ces bits, lesquels étaient un '1' à l'émission et lesquels étaient un '0' ? La façon la plus simple de le déterminer est de fixer un seuil de décision. Par exemple, si le seuil est fixé à 0,5, tous les bits dont la valeur moyenne est supérieure ou égale à 0,5 sont considérés des '1', et tous ceux dont la valeur moyenne est inférieure à 0,5 sont considérés des '0'.

Maintenant, comment déterminer la valeur du seuil de décision ? Le but est évidemment de fixer le seuil à une valeur qui minimise le taux d'erreur et qui n'introduit pas un biais. Il faut autant que possible éviter qu'un '1' à l'émission soit classé parmi les '0' à la réception, et vice-versa. Pour y arriver, plusieurs approches sont possibles.

D'abord, il est intéressant de regarder la distribution des valeurs à la sortie du canal. La figure 3.2 résulte d'une simulation de 500 000 bits, après transmission sur 50 km de fibre. Elle

présente la proportion de '1' ou la proportion de '0' ayant une certaine valeur à la sortie du canal. On remarque d'abord que la distribution n'est pas symétrique : les '1' n'ont pas la même distribution que les '0'. De plus, le système simulé est déterministe, donc la sortie dépend uniquement de la séquence de bits. C'est pourquoi certaines valeurs intermédiaires ne sont jamais présentes. Pour le cas particulier de la figure 3.2, si le seuil de décision est placé entre 0,36 et 0,6, aucun bit n'est en erreur, puisque tous les bits '0' ont une valeur moyenne inférieure et tous les '1' ont une valeur moyenne supérieure.

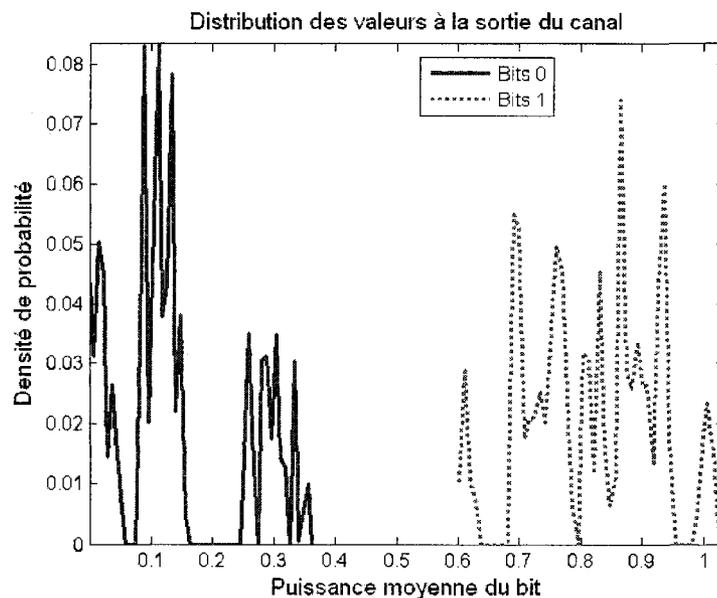


Figure 3.2 – Distribution des valeurs à la sortie du canal

Suivant cette constatation, il est intéressant de trouver, pour une séquence de bits test, la valeur minimale pour un '1' ( $M_{\min}^1$ ) et la valeur maximale pour un '0' ( $M_{\max}^0$ ). Si  $M_{\min}^1 > M_{\max}^0$ , alors tout seuil de décision entre  $M_{\min}^1$  et  $M_{\max}^0$  produit un taux d'erreur nul. La figure 3.3 présente les résultats d'un tel test pour une séquence de 50 000 bits, pour des distances allant jusqu'à 80 km.

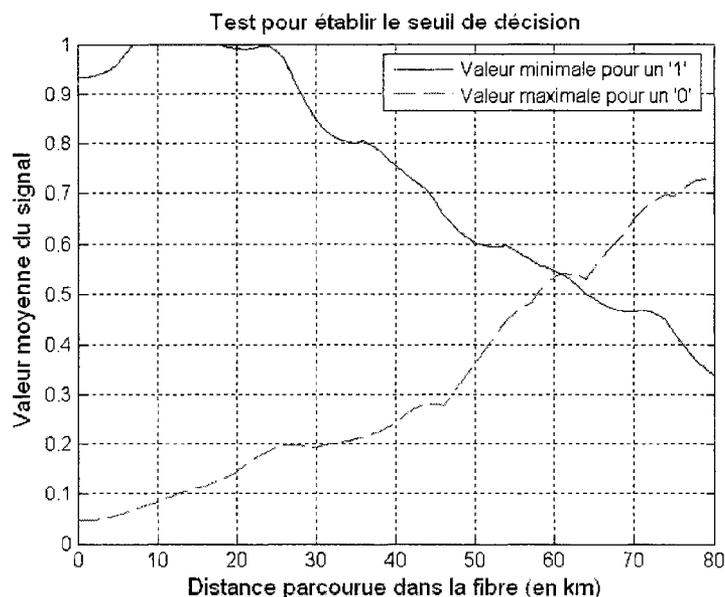


Figure 3.3 – Test pour le seuil de décision

La figure 3.3 fournit plusieurs renseignements utiles. D’abord, jusqu’à approximativement 60 km, la condition  $M_{\min}^1 > M_{\max}^0$  est respectée, donc il est possible de choisir un seuil de décision qui produit un taux d’erreur nul. Par exemple, après 50 km de fibre, le taux d’erreur devrait être nul si le seuil de décision est fixé entre 0,36 et 0,6. Par contre, après 70 km de fibre, si le seuil est fixé à 0,55, alors il existe au moins un ‘1’ à l’émission converti en ‘0’ à la réception et au moins un ‘0’ converti en ‘1’. Ainsi, d’après ces résultats, la limite de transmission dans la fibre semble se situer autour de 60 km pour nos simulations, et non autour de 66 km comme l’hypothèse sur le pourcentage d’énergie dans les bits adjacents le suggérait.

La figure 3.3 ne nous renseigne toutefois pas sur la fréquence des erreurs après 60 km ni sur le meilleur seuil à choisir. Pour ces questions, il faut s’intéresser au calcul du taux d’erreur, qui est présenté à la prochaine section.

### 3.2 Calcul du taux d’erreur

Il existe plusieurs méthodes pour calculer le taux d’erreur sur les bits (BER en anglais). Celle qui sera utilisée est la méthode de Monte Carlo. La prochaine sous-section explique en quoi consiste cette méthode et pourquoi elle a été choisie.

### 3.2.1 Méthode de Monte Carlo

La méthode de Monte Carlo est fréquemment utilisée dans les simulations numériques lorsqu'une ou des entrées du système sont des variables aléatoires et que le modèle du système à l'étude est complexe ou non linéaire. Soit, à la figure 3.4, un système qui prend en entrée  $x = \{x_1, x_2, \dots, x_n\}$  et qui produit en sortie la fonction  $g(x)$ . Pour le cas de l'évaluation du taux d'erreur,  $x$  correspond à la combinaison du bruit (considéré nul pour le moment) et de la séquence de bits transmise, et  $g(x)$  vaut 1 lorsqu'à la sortie, le bit est en erreur, et vaut 0 autrement.

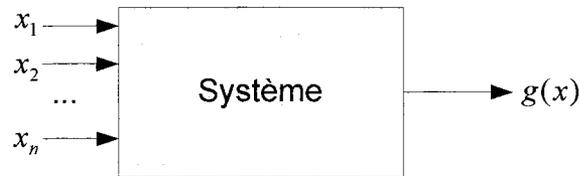


Figure 3.4 – Exemple d'un système simulé

Le taux d'erreur est l'espérance de  $g(x)$ , qui se calculerait ainsi :

$$E[g(x)] = \int_{\Omega} g(x) f_X(x) dx \quad (3.3)$$

où  $f_X(x)$  est la fonction de densité de probabilité de  $x$ .

Or, lorsque le modèle est complexe, il est difficile, voire impossible, d'obtenir une version analytique de  $g(x)$  afin de pouvoir évaluer l'intégrale. L'idée est donc de générer plusieurs échantillons  $x_i$  respectant la fonction de densité de probabilité  $f_X(x)$ , d'évaluer  $g(x)$  pour chacun des échantillons puis d'obtenir un estimateur du taux de probabilité en faisant la moyenne des valeurs trouvées :

$$\tilde{g}_n(x) = \frac{1}{n} \sum_{i=1}^n g(x_i) \quad (3.4)$$

$$\text{Ou encore : } BER = \frac{\text{Nombre d'erreurs}}{\text{Nombre de bits simulés}} \quad (3.5)$$

La précision de cet estimateur est proportionnelle au nombre d'échantillons générés.

Dans le cas de la simulation d'une fibre optique, il existe d'autres méthodes plus précises pour estimer le taux d'erreur, notamment estimer la fonction de densité de probabilité à la sortie du système et de là, calculer la proportion de '0' dont la valeur est supérieure au seuil et la proportion de '1' dont la valeur est inférieure. Or, une telle approche n'est pas applicable avec un décodeur turbo, car contrairement à l'application d'un seuil où l'on voit facilement quelles valeurs sont en erreur, il est difficile de savoir a priori ce que le décodeur turbo sera ou non en mesure de corriger. Pour fins de comparaison, il est préférable que toutes les simulations soient effectuées avec la même méthode, afin d'être certain que la manière de calculer les erreurs n'influence pas le résultat. Ainsi, il est préférable d'appliquer une méthode de type Monte Carlo à la fois pour les simulations sans décodeur et les simulations avec décodeur.

### 3.2.2 Optimisation du seuil de décision

Une façon d'optimiser le seuil de décision est de calculer le taux d'erreur pour différents seuils de décision, puis de choisir celui produisant le taux d'erreur le plus faible. La figure 3.5 présente le résultat d'un tel test après transmission sur 70 km de fibre, en simulant une séquence de 50 000 bits. D'après ce test, le seuil optimal serait de 0,475 pour un taux d'erreur de 5,77%.

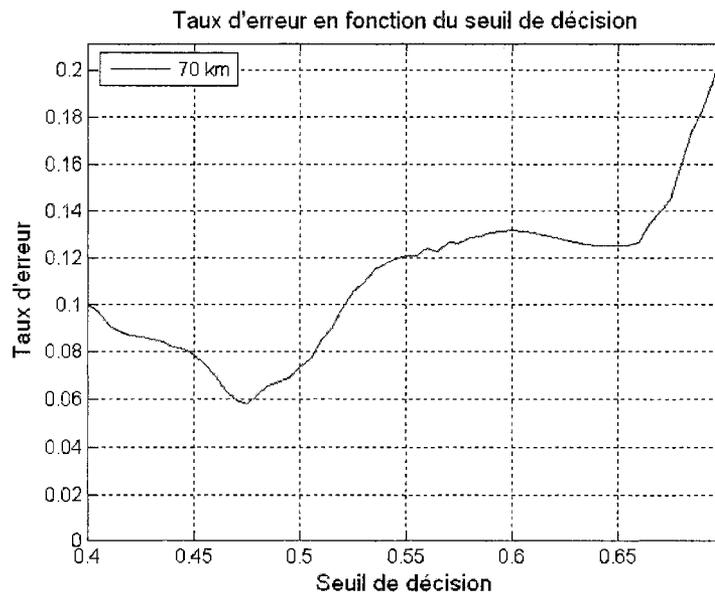


Figure 3.5 – Taux d'erreur en fonction du seuil après 70 km de fibre

Il est intéressant de savoir si, lorsque le seuil de décision permet de minimiser le nombre d'erreurs, les erreurs sur les '1' et les '0' sont équiprobables ou si l'un des deux bits est privilégié. Pour le déterminer, le test précédent est repris, mais cette fois-ci en distinguant les '1' et les '0' en erreur. À la figure 3.6, on peut voir clairement qu'après 70 km de fibre, en choisissant le seuil de décision optimal, la plupart des erreurs sont des '0' transformés en '1'.

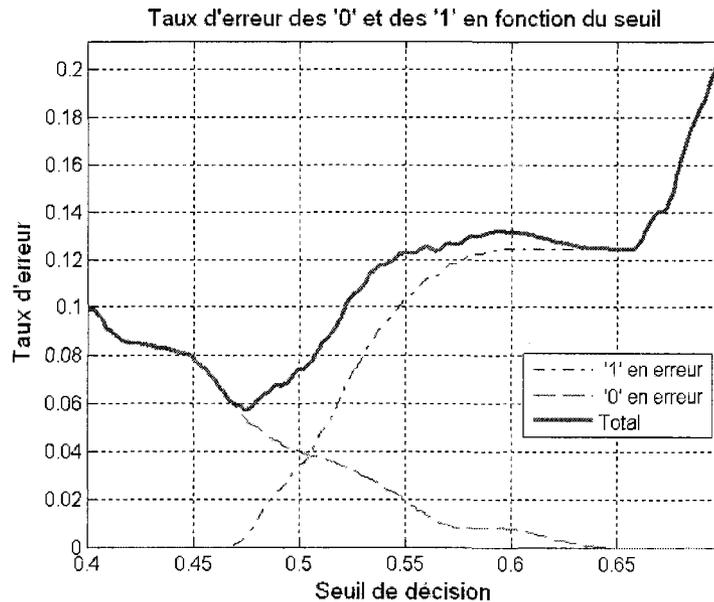


Figure 3.6 – Nombre de bits en erreur en fonction du seuil de décision

Pour comprendre, il faut aller voir du côté de la distribution des valeurs à la sortie du canal, à la figure 3.7. On remarque la forte proportion de '1' ayant une puissance moyenne de l'ordre du seuil optimal. Ainsi, en augmentant le seuil, le nombre de '1' en erreur augmente beaucoup plus rapidement que le nombre de '0' en erreur ne diminue. C'est pourquoi, du point de vue du taux d'erreur, il est préférable dans ce cas-ci que les '1' soient moins souvent en erreur que les '0'. La figure 3.7 permet aussi d'expliquer les plateaux à la figure 3.6, puisque certaines valeurs ne sont pas possibles.

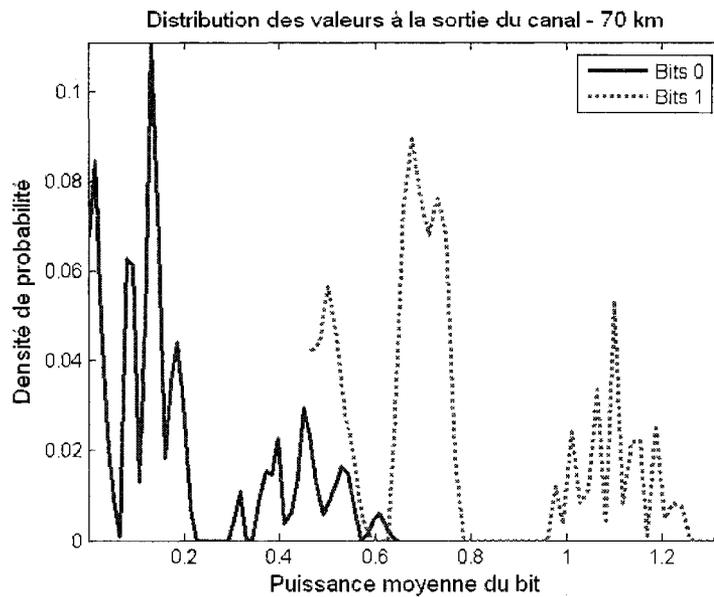


Figure 3.7 – Distribution des valeurs à la sortie du canal après 70 km de fibre

### 3.2.3 Variabilité du résultat

Les seuils obtenus jusqu'à maintenant sont optimaux pour une séquence particulière. Or, la précision de l'estimation du BER est proportionnelle au nombre d'échantillons générés. Le seuil optimal varie-t-il donc beaucoup d'une séquence à l'autre ? Comment déterminer le seuil qui, pour une distance donnée, sera satisfaisant peu importe la séquence ?

La première étape est d'étudier la variabilité du résultat. Le test de la section 3.2.2 a été effectué avec une séquence de 50 000 bits. Si le même test est effectué à 100 reprises, y a-t-il une grande différence entre les résultats ? Le seuil moyen trouvé est de 0,47455 et les valeurs oscillent toutes entre 0,473 et 0,475. Pour ce qui est du taux d'erreur minimal obtenu, la moyenne est de 0,057 et la variance est de  $9,36 \times 10^{-7}$ . Le résultat varie donc peu d'une séquence à l'autre. Le seuil moyen obtenu avec 100 simulations peut donc être considéré le seuil optimal à 70 km.

Il faut toutefois considérer le temps de simulation requis. Trouver le seuil optimal pour une séquence de 50 000 bits requiert un temps de calcul de 24s. Simuler 100 séquences demande donc environ 40 minutes. Est-ce possible d'obtenir d'aussi bons résultats en moins de temps ?

Deux pistes sont à explorer : réduire la longueur des séquences et réduire le nombre de séquences.

D'abord, en réduisant la séquence à 10 000 bits, le temps de calcul par séquence passe de 24 secondes à 2,3 secondes. Pour 100 séquences, le temps nécessaire est donc de 3 minutes 50 secondes. La figure 3.8 présente le seuil de décision et le taux d'erreur moyen après 100 séquences, pour 50 essais différents. Le seuil optimal varie entre 0,47412 et 0,47464 et le taux d'erreur varie entre 0,05645 et 0,057789. On peut donc conclure que malgré la réduction de la longueur des séquences, les résultats sont demeurés valables.

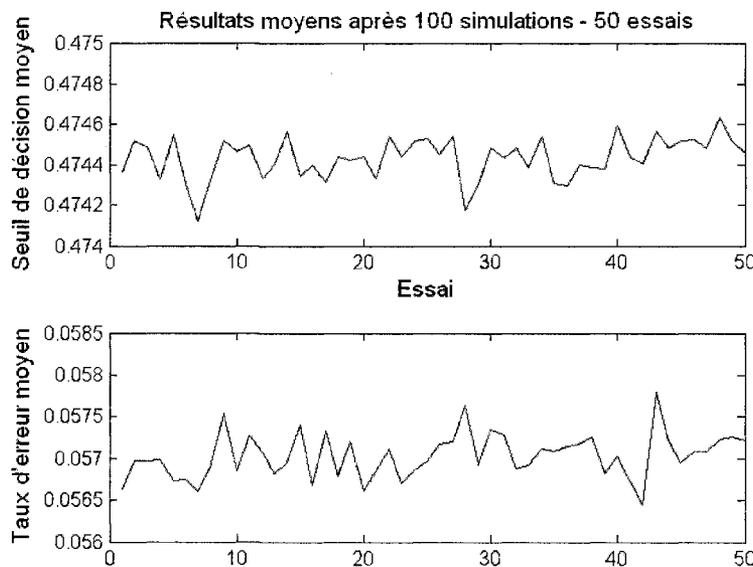


Figure 3.8 – Variabilité des résultats, séquences de 10 000 bits

En prenant la moyenne sur 50 séquences plutôt que 100, le temps de simulation est réduit de moitié et les résultats demeurent acceptables. Le seuil optimal varie alors entre 0,47403 et 0,47479. Pour les tests subséquents, le seuil de décision optimal sera donc choisi en calculant la moyenne du seuil optimal pour 50 simulations de 10 000 bits.

Bien sûr, il serait possible d'optimiser davantage les paramètres afin d'obtenir le meilleur compromis entre précision et temps de simulation. Toutefois, puisque ce seront principalement les mêmes distances qui seront simulées encore et encore, il est préférable d'opter pour une

plus grande précision et d'établir une table de valeurs, donnant pour chaque distance le seuil optimal correspondant.

#### 3.2.4 Impact du nombre d'échantillons par bit

Avant d'aller plus loin dans les simulations, il faut s'assurer que le nombre d'échantillons par bit choisi n'influence pas les résultats. Au chapitre 2, il a été prouvé que 8 échantillons par bit était le minimum requis pour respecter le théorème d'échantillonnage. Le fait de prendre le nombre minimum d'échantillons par bit a en théorie un impact sur l'aspect visuel du signal simulé, puisqu'une sinusoïde de fréquence  $\omega_{max}$  a l'allure d'une onde triangulaire. Cela devrait avoir un impact sur le résultat de l'intégration au récepteur, l'intégrale d'une sinusoïde ne donnant pas le même résultat que l'intégrale d'une onde triangulaire. Cet impact doit-il être pris en considération ou est-il négligeable?

Pour répondre à cette question, quelques tests ont été effectués. Dans un premier temps, une même séquence aléatoire de 50 000 bits est simulée avec 8 et 80 échantillons par bit, après 70 km de fibre. À 80 échantillons par bit, soit dix fois plus de points par bit que le minimum requis, on considère généralement que l'aspect visuel du signal n'est pas affecté. Or, après simulation, la différence après intégration au récepteur est relativement faible. La différence moyenne entre les simulations à 8 et 80 échantillons est de 0,0122 pour ce qui est de la puissance d'un bit, et la différence maximale est de 0,051. Cet écart peut-il éventuellement avoir un impact sur le taux d'erreur ?

Pour le savoir, il faut calculer le taux d'erreur moyen à 80 échantillons par bit. Pour ce faire, le test effectué sera celui retenu à la section 3.2.3 : la moyenne de 50 séquences de 10 000 bits, considérant pour chacune des séquences l'application du seuil de décision optimal. Ainsi, le seuil moyen trouvé est de 0,4705 pour un taux d'erreur moyen de 0,061. À 8 échantillons par bit, on trouve plutôt un taux d'erreur moyen de 0,057. La différence de 7% entre les deux valeurs peut sembler importante, mais il ne faut pas oublier que les taux d'erreur sont normalement exprimés sur une échelle logarithmique. Ainsi, graphiquement, il y a peu de différence entre  $5,7 \times 10^{-2}$  et  $6,1 \times 10^{-2}$ . Vu la faible différence au niveau des résultats et la grande différence au niveau des temps de simulation requis, il est donc raisonnable de continuer les simulations en utilisant seulement 8 échantillons par bit.

### 3.2.5 Taux d'erreur minimal en fonction de la distance

Les tests présentés jusqu'à maintenant permettent de déterminer le seuil de décision optimal et d'estimer le taux d'erreur minimal obtenu. Il serait maintenant intéressant de regarder comment ces deux variables évoluent en fonction de la distance. La figure 3.9 s'attarde à l'évolution du seuil de décision optimal. Rappel : le seuil de décision optimal est déterminé en prenant la moyenne sur 50 essais du seuil optimal pour une séquence de 10 000 bits. Pour une séquence donnée, le seuil optimal est celui minimisant le taux d'erreur. Si le taux d'erreur est identique pour une plage de seuil, par exemple pour le cas où plusieurs seuils différents permettent d'obtenir un taux d'erreur nul, alors le seuil optimal est choisi au milieu de cette plage.

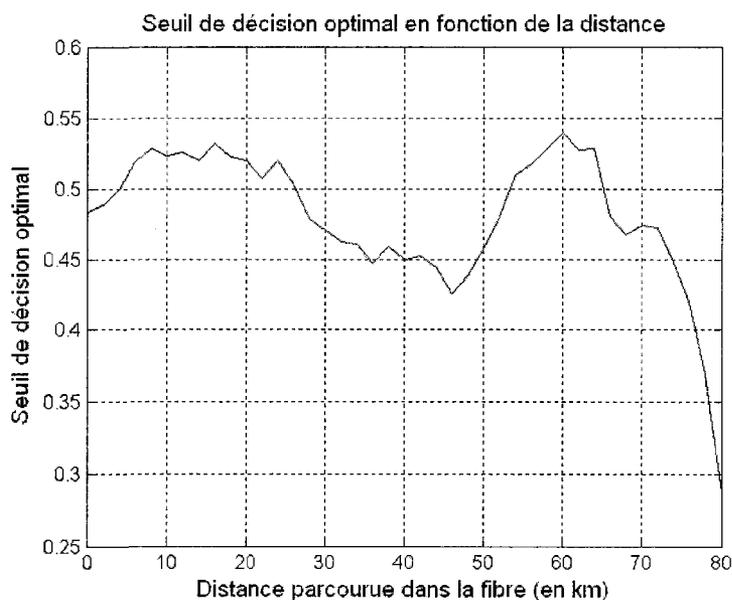


Figure 3.9 – Seuil de décision optimal en fonction de la distance

Comment expliquer l'évolution du seuil de décision optimal ? D'abord, il a été expliqué précédemment qu'à 70 km, les '1' sont privilégiés au détriment des '0', à cause de la distribution des valeurs à la sortie du canal. Il est logique de faire l'hypothèse qu'il en est également ainsi dès que le taux d'erreur n'est pas nul. La figure 3.10 présente le taux d'erreur moyen obtenu lors des simulations en fonction de la distance. Le taux d'erreur est nul jusqu'à 60 km. Or, à la figure 3.9, le seuil de décision optimal commence à diminuer à partir de 60 km. Plus la distance augmente, plus la valeur minimale pour un '1' diminue et donc plus le seuil de décision doit être bas pour limiter le nombre de '1' en erreur.

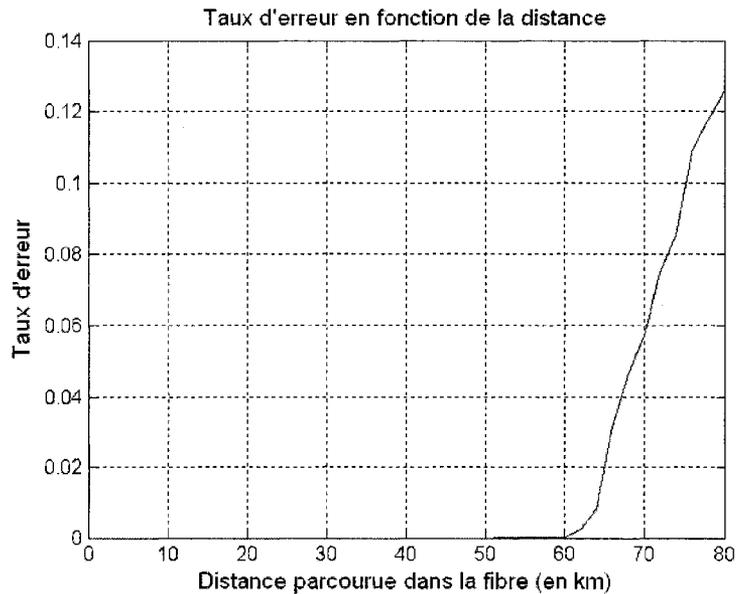


Figure 3.10 – Taux d'erreur minimal en fonction de la distance

Pour ce qui est de l'évolution du seuil de décision entre 0 km et 60 km, une manière de l'expliquer est de revoir le test présenté à la figure 3.3, soit l'évolution de la valeur minimale d'un '1' et de la valeur maximale d'un '0' en fonction de la distance. Jusqu'à 60 km, la condition  $M_{\min}^1 > M_{\max}^0$  est respectée, donc il est possible de choisir un seuil de décision qui produit un taux d'erreur nul. Dans un tel cas, le seuil optimal choisi vaut  $(M_{\min}^1 + M_{\max}^0)/2$ , soit le milieu de la plage des seuils ne produisant pas d'erreur. À la figure 3.3, on remarque qu'avant 25 km, la valeur minimale d'un '1' demeure relativement stable tandis que la valeur maximale d'un '0' augmente. Ainsi, le seuil optimal augmente avec la distance. Par contre, entre 25 km et 45 km, la valeur minimale du '1' diminue plus rapidement que la valeur maximale du '0', ce qui cause une diminution du seuil de décision optimal. Finalement, de 50 à 60 km, c'est l'inverse qui se produit, donc le seuil de décision augmente à nouveau. Ainsi, les seuils de décision obtenus à la figure 3.9 sont conséquents avec l'information de la figure 3.3.

### 3.2.6 Limite de propagation

D'après les tests effectués précédemment, le taux d'erreur est nul jusqu'à 60 km, puis augmente rapidement pour des distances plus grandes. Jusqu'à maintenant, aucune source de bruit n'a été considérée. Or, l'ajout de bruit ne peut qu'augmenter le taux d'erreur. Ainsi,

d'après les simulations, la limite de propagation dans la fibre, pour le modèle utilisé, est d'au maximum 60 km. Ce résultat diffère des 80 km normalement obtenus en pratique et des 66 km d'après l'hypothèse du 50% d'énergie dans les bits adjacents. Par conséquent, deux conclusions s'imposent. Premièrement, l'hypothèse que la limite de propagation est atteinte lorsque 50% de l'énergie d'un bit se retrouve dans les bits adjacents ne semble pas fondée. Deuxièmement, le modèle utilisé pour les simulations est, pour une même distance, plus dispersif qu'une fibre optique en pratique. Toutefois, comme il a été mentionné précédemment, le modèle peut tout de même être utilisé, puisque l'objectif est de déterminer si la modification proposée diminue le taux d'erreur en milieu dispersif et non de déterminer de manière absolue l'amélioration obtenue en fonction de la distance.

### 3.2.7 Dispersion chromatique accumulée

Puisque les résultats de simulation ne semblent pas concorder avec les distances en pratique, il serait préférable de ne plus parler de distance, mais plutôt de dispersion chromatique accumulée. La dispersion chromatique accumulée se mesure en ps/nm. Pour la calculer, il faut retourner à l'équation de Sellmeier étudiée dans le chapitre précédent :

$$D(\lambda) = \frac{\lambda S_0}{4} \left( 1 - \left( \frac{\lambda_0}{\lambda} \right)^4 \right) \quad (3.6)$$

avec :  $\lambda_0 = 1310 \text{ nm}$

$$S_0 = 0,092 \text{ ps} / \text{nm}^2 \cdot \text{km}$$

Pour les simulations, la longueur d'onde utilisée est de 1550 nm, d'où  $D(\lambda) = 17,5 \text{ ps}/(\text{nm} \cdot \text{km})$ .

La dispersion chromatique accumulée, en ps/nm, vaut donc :

$$DC_{\text{accumulée}} = 17,5 \times \text{distance}(\text{en km}) \quad (3.7)$$

Dans les simulations subséquentes, les distances seront remplacées par la dispersion chromatique accumulée.

### 3.3 Ajout du bruit

Les tests précédents ont tous été effectués sans considérer le bruit. Or, en pratique, le bruit existe et il est non négligeable. Ainsi, la présente section explique d'abord d'où provient le bruit, puis comment il est modélisé pour l'introduire dans les simulations.

#### 3.3.1 Sources de bruit

La principale source de bruit est l'émission spontanée amplifiée (ASE – *Amplified Spontaneous Emission*) produite dans les amplificateurs optiques dopés à l'erbium (EDFA – *Erbium Doped Fiber Amplifier*). Les deux autres principales sources de bruit proviennent de l'électronique. Le bruit thermique est généré par l'agitation thermique des porteurs de charge. Quant au bruit de grenaille (*shot noise*), il s'explique par le fait que le courant est constitué de porteurs de charge, les électrons. Ces porteurs de charge n'arrivent pas régulièrement, donc le nombre de porteurs mesurés sur un intervalle de temps est aléatoire.

#### 3.3.2 Modélisation du bruit

L'impact de ces sources de bruit doit être introduit dans le modèle de simulation. Dans les réseaux multiplexés par répartition en longueur d'onde (WDM – *Wavelength-division multiplexing*), les éléments de multiplexage/démultiplexage introduisent une forte perte d'insertion. Pour compenser ces pertes d'insertion, il faut utiliser l'amplification optique par EDFA. Par le fait même, le bruit ASE devient dominant et c'est pourquoi le bruit électronique est généralement négligé lors de la modélisation du bruit.

Le bruit ASE est typiquement modélisé par un bruit blanc gaussien. Or, d'après [CAI et coll., 2003], il serait plutôt de type chi-carré. De plus, toujours selon cet article, le fait d'utiliser une approximation gaussienne lors du décodage à décision pondérée (*soft decoding*) peut significativement dégrader les performances du décodeur.

Encore une fois, puisque l'objectif du présent projet de recherche est d'abord et avant tout de déterminer si la modification proposée permet d'améliorer les performances de décodage des erreurs après transmission dans un milieu dispersif, le fait d'utiliser un bruit gaussien plutôt qu'un bruit chi-carré ne devrait pas fausser les résultats. Bien sûr, de manière absolue, l'amélioration ne serait pas la même, mais il est raisonnable de supposer qu'une modification

améliorant les performances avec un bruit gaussien améliorerait aussi les performances pour un bruit chi-carré. C'est pourquoi, pour simplifier la modélisation du problème, le bruit est considéré gaussien et est introduit à la fin du canal, juste avant le récepteur.

### 3.4 Simulation des résultats

#### 3.4.1 Courbes à obtenir

Typiquement, lors de l'étude des performances d'un canal de communication par fibre optique, deux graphiques différents sont présentés. Le premier est le taux d'erreur (BER) en fonction du rapport signal sur bruit (OSNR : *Optical Signal-to-Noise Ratio*) pour une quantité donnée de dispersion chromatique accumulée. Le second est le BER en fonction de la dispersion chromatique accumulée pour un OSNR donné. Mais pour obtenir ces graphiques, il faut d'abord savoir comment calculer l'OSNR.

Comme son nom l'indique, l'OSNR est le rapport signal sur bruit, c'est-à-dire un rapport entre la puissance du signal et la puissance du bruit. La plupart du temps, il est exprimé en décibel. Il se calcule de la façon suivante :

$$OSNR(dB) = 10 \log \left( \frac{P_{signal}}{P_{bruit}} \right) \quad (3.8)$$

Puisque le bruit ajouté est un bruit blanc gaussien, sa puissance est équivalente à sa variance car sa moyenne est nulle :

$$P_{bruit} = \sigma_{bruit}^2 \quad (3.9)$$

Pour ce qui est de la puissance du signal, il faut considérer la puissance moyenne au récepteur. Puisque le signal au récepteur varie en fonction de la dispersion chromatique accumulée, il est important de calculer  $P_{signal}$  indépendamment pour chaque cas. Ce calcul peut être fait simultanément à l'élaboration de la table des seuils de décision optimaux. En effet, l'algorithme simule 50 séquences de 10 000 bits pour chaque valeur de dispersion chromatique accumulée. L'équation (3.2) à la section 3.1.1 permet de calculer la puissance moyenne d'un bit. En calculant la valeur moyenne pour tous les bits simulés avec une certaine dispersion

chromatique accumulée, il est possible d'obtenir  $P_{signal}$  et de conserver cette valeur dans une table. La figure 3.11 présente la puissance moyenne du signal en fonction de la dispersion chromatique accumulée.

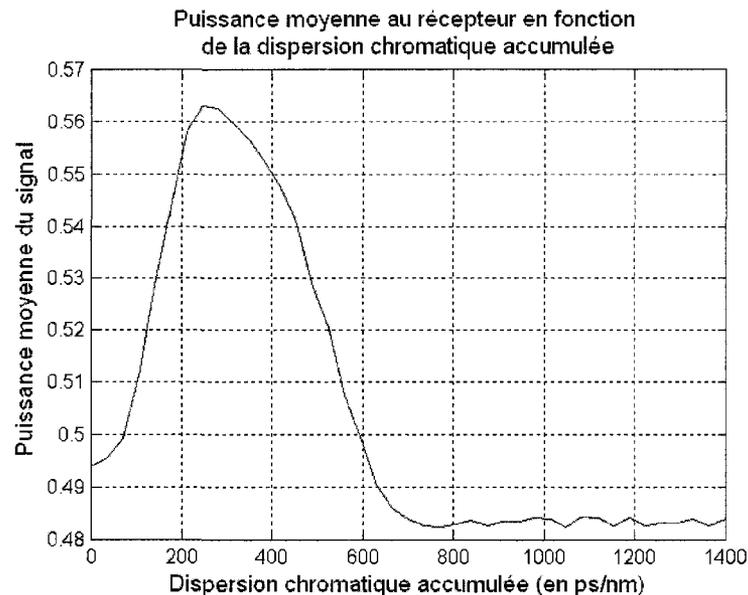


Figure 3.11 – Puissance du signal au récepteur

La formule pour le calcul du BER a été donnée précédemment :

$$BER = \frac{\text{Nombre d'erreurs}}{\text{Nombre de bits simulés}} \quad (3.5)$$

Typiquement, dans les réseaux de communication par fibre optique, les taux d'erreur visés sont très faibles, de l'ordre de  $10^{-12}$ , c'est-à-dire 1 bit en erreur par Tb. Seules des séquences de bruit ou des combinaisons de bits très rares mènent à une erreur au décodage. Une simulation Monte Carlo requiert au moins  $100/P_e$  essais pour estimer un BER de  $P_e$  ( $P_e$  : probabilité d'erreur) avec une précision relative de 10% [SMITH et coll., 1997]. Il faudrait donc simuler des centaines de Tb pour avoir un taux d'erreur à  $10^{-12}$  relativement fiable. Seulement pour la simulation du canal sans décodeur, une simulation pour 10 000 bits dure environ 1,3 s. Pour  $100 \times 10^{12}$  bits, soit 100 Tb, la simulation durerait des centaines d'années avec un seul ordinateur! Évidemment, il faut trouver une alternative.

Il existe plusieurs méthodes pour réduire le temps de calcul requis afin d'obtenir la même précision. Ces méthodes comprennent l'échantillonnage stratégique (*Importance Sampling*) et la méthode Monte Carlo multicanonique. L'idée centrale derrière ces méthodes est d'augmenter la fréquence d'occurrence des événements rares menant à des erreurs. Le problème réside dans l'application pratique de ces méthodes. L'échantillonnage stratégique est une méthode empirique qui requiert essais et erreurs afin d'identifier une distribution de probabilité satisfaisante. À défaut d'en trouver une, la méthode augmente la variance de l'estimateur du taux d'erreur au lieu de la réduire. Quant à la méthode Monte Carlo multicanonique, elle est systématique, mais sa complexité croît avec la longueur du code. Elle a déjà été utilisée avec succès pour un code LDPC(96,50) [HOLZLÖHNER et coll., 2005], mais le code utilisé dans le présent projet, soit le BCH(144,128) x BCH(256,239), est beaucoup plus long.

Appliquer la méthode Monte Carlo de base afin d'obtenir des taux d'erreur de l'ordre de  $10^{-12}$  demande un temps de calcul trop long, mais les méthodes existantes pour réduire le temps de calcul ne semblent pas s'appliquer. Comment faire alors ? Pour solutionner ce problème, il faut se rappeler l'objectif principal du projet : vérifier si l'amélioration proposée est efficace. Pour le savoir, il n'est pas nécessaire de travailler à des taux d'erreur aussi faibles. En effet, les erreurs sont beaucoup plus rares pour un taux d'erreur de  $10^{-12}$  que pour un taux de  $10^{-3}$ , par exemple. En conséquence, il y a plus de chance que le nombre d'erreurs soit en deçà de la capacité de correction du code à  $10^{-12}$  qu'à  $10^{-3}$ . Ainsi, si une amélioration est démontrée à  $10^{-3}$ , il est possible de supposer une amélioration à  $10^{-12}$ . Par conséquent, les simulations se feront plutôt en visant le taux d'erreur minimum atteignable en un temps raisonnable.

### 3.4.2 Résultats

Suite à ces considérations pratiques, il est maintenant possible d'obtenir les deux graphiques typiques pour l'étude des performances d'un canal de communication par fibre optique, soit le taux d'erreur (BER) en fonction du rapport signal sur bruit (OSNR) pour une dispersion chromatique accumulée donnée et le BER en fonction de la dispersion chromatique accumulée pour un OSNR donné. Le premier graphique est présenté à la figure 3.12 tandis que le second apparaît à la figure 3.13.

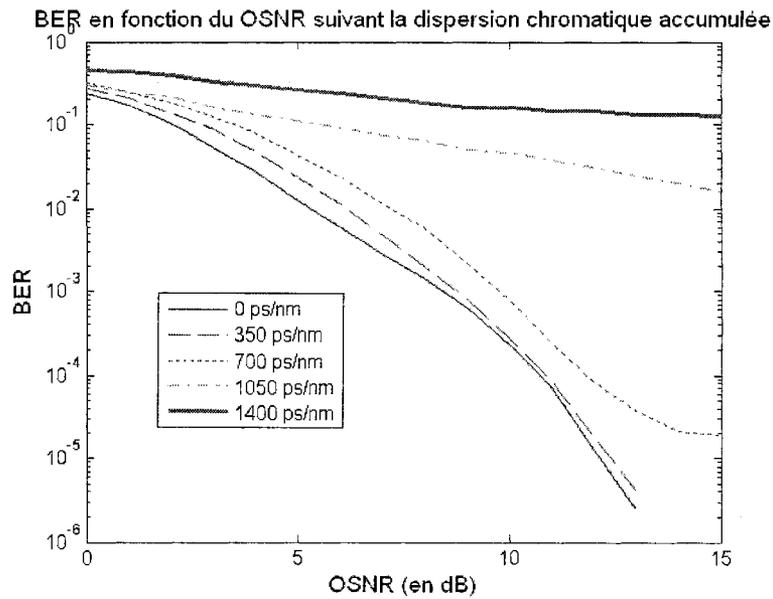


Figure 3.12 – BER en fonction du OSNR suivant la dispersion chromatique accumulée

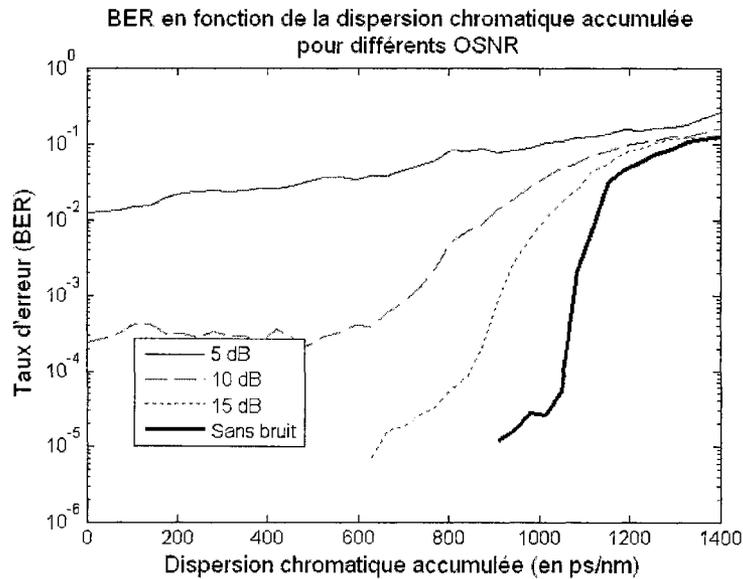


Figure 3.13 – BER en fonction de la dispersion chromatique accumulée

Ces graphiques permettent de constater que plus le rapport signal sur bruit augmente, plus faible est le taux d'erreur. Toutefois, lorsque la dispersion chromatique accumulée est grande, le fait d'augmenter la puissance du signal ne permet pas de diminuer considérablement le taux

d'erreur. En effet, entre 1050 ps/nm et 1400 ps/nm de dispersion chromatique, le taux d'erreur n'est pas nul même sans bruit. Ainsi, même en augmentant la puissance du signal par rapport au bruit, il est impossible d'obtenir un taux d'erreur inférieur au taux d'erreur sans bruit. Finalement, la figure 3.13 permet de constater l'existence d'un taux d'erreur minimum relatif au rapport signal sur bruit. Pour un OSNR donné, même en diminuant la quantité de dispersion chromatique accumulée, le taux d'erreur ne diminue pas substantiellement. Cela signifie qu'une fois le plateau atteint, la principale cause d'erreur est le bruit, et non la qualité du signal. De ce fait, augmenter davantage la qualité du signal ne permet pas de compenser les erreurs causées par le bruit.

### 3.5 Résumé

Le but du présent chapitre était de mesurer les performances du canal de communication. Pour y arriver, il a d'abord été nécessaire d'expliquer l'intégration du signal au récepteur, puis de déterminer le seuil de décision optimal. Ce dernier dépend de la distance parcourue dans la fibre, donc de la dispersion chromatique accumulée. Le calcul du seuil de décision optimal se fait en prenant la moyenne, pour 50 séquences de 10 000 bits, du seuil de décision produisant le plus faible taux d'erreur.

Les deux principaux graphiques à obtenir pour illustrer les performances du canal sont le taux d'erreur (BER) en fonction du rapport signal sur bruit (OSNR) pour une dispersion chromatique accumulée donnée et le BER en fonction de la dispersion chromatique accumulée pour un OSNR donné. Suite à des considérations pratiques, il a été déterminé que la meilleure façon d'obtenir un BER fiable en un temps raisonnable est d'utiliser la méthode de Monte Carlo, mais de limiter les résultats à des taux d'erreur beaucoup plus élevés que les  $10^{-12}$  normalement obtenus pour les réseaux de fibre optique.

Maintenant que les performances du canal sans système de correction d'erreurs sont établies, il faut s'intéresser au décodeur que le présent projet tente d'améliorer. Quelles sont les caractéristiques du code utilisé ? Quel algorithme est utilisé pour le décodage ? Dans quelle mesure son utilisation améliore-t-elle les performances du canal ? Ces questions trouveront réponses dans le prochain chapitre.

## CHAPITRE 4

### DÉCODEUR INITIAL

L'objectif du présent projet de recherche est de modifier un turbo code en bloc développé par Mitsubishi Electric [MIZUOCHI et coll., 2004], [TAGAMI et coll., 2005], [OUCHI et coll., 2006] de manière à l'optimiser pour la correction d'erreurs après transport en milieu dispersif. Mais avant de proposer une amélioration, il faut d'abord pouvoir simuler le système de référence et mesurer ses performances. Les deux chapitres précédents se sont intéressés à deux étapes préliminaires essentielles : la simulation du canal et l'obtention du taux d'erreur. Ce chapitre portera sur l'étape subséquente : l'implantation du système correcteur d'erreur de référence dans la simulation.

L'application d'un code correcteur d'erreur comprend deux étapes : l'encodage et le décodage. Quelles sont les caractéristiques du code utilisé ? Comment obtenir un mot codé à partir des bits à transmettre ? La section 4.1 répondra à ces questions. Quant au décodage, il sera abordé dans les sections subséquentes. La section 4.2 présentera un exemple simple de décodage avec un turbo code. Les algorithmes standards de décodage sont à décision binaire. Or, pour plusieurs canaux de communication, la sortie du canal n'est pas nécessairement binaire. C'est pour remédier à cette situation qu'a été créé l'algorithme de Chase [CHASE 1972], qui sera présenté à la section 4.3. Toutefois, une modification de cet algorithme est nécessaire pour décoder de façon itérative. Cette modification, le calcul de la fiabilité [PYNDIAH 1998], fera l'objet de la section 4.4. L'équipe de Mitsubishi Electric a apporté deux modifications supplémentaires à l'algorithme de Pyndiah, modifications qui améliorent les performances d'environ 0,4dB. Toutefois, les résultats de simulation, présentés à la section 4.5, seront obtenus avec l'algorithme de Pyndiah original, car la documentation disponible est plus complète.

#### 4.1 Encodage

Le décodeur étudié se base sur un système proposé par Mitsubishi et présenté dans les articles suivants : [MIZUOCHI et coll., 2004], [TAGAMI et coll., 2005] et [OUCHI et coll., 2006].

Le code est un code en bloc turbo BCH(144,128) x BCH(256,239). La figure 4.1 présente la structure d'un tel code.

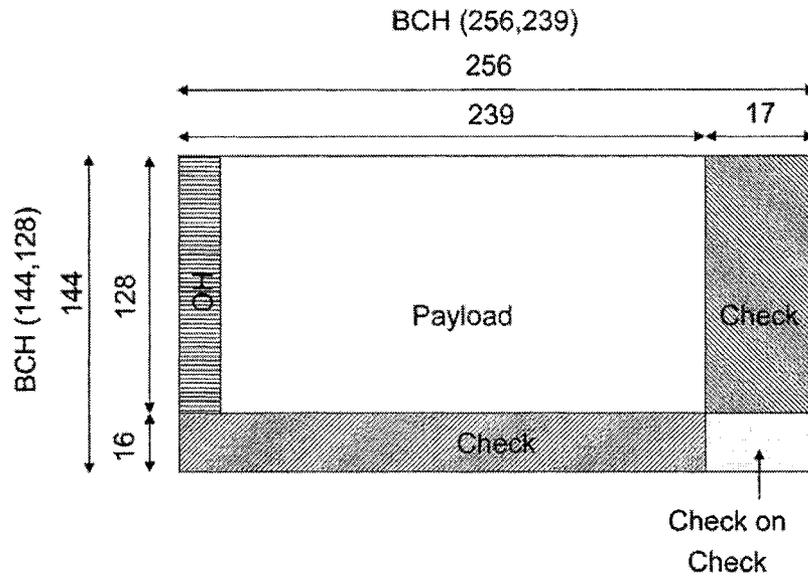


Figure 4.1 – Structure du code proposé [MIZUOCHI et coll., 2004]

#### 4.1.1 Choix du code

Le code proposé par Mitsubishi n'a pas été choisi au hasard. Il a été choisi pour se conformer à la norme G.709, un standard dans les communications par fibre optique. La trame OTN (Optical Transport Network) G.709 contient des bits supplémentaires pour l'opération, l'administration et la maintenance du réseau et propose l'utilisation d'un code RS(255,239). La figure 4.2 présente une telle trame. Les informations à transmettre sont réparties en 4 rangées de 3808 bits chacune. 16 bits d'en-tête sont ajoutés à chaque rangée, puis chaque rangée est divisée en 16 sous-rangées de 239 bits, codées chacune suivant RS(255,239). Ainsi, au total, une trame G.709 contient 64 sous-rangées de 239 bits à coder.

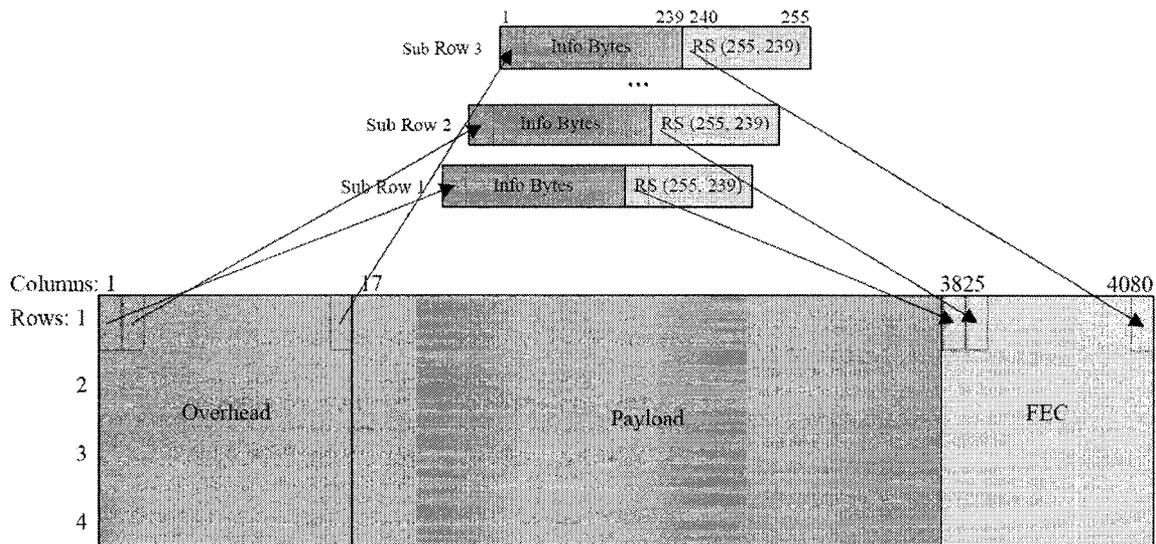


Figure 4.2 – Structure d'une trame G.709 [BARLOW 2003]

Le code de Mitsubishi, le BCH(144,128) x BCH(256,239), est donc compatible avec la norme G.709, puisqu'il contient 128 rangées de 239 bits à coder, ce qui est équivalent à deux trames G.709. Cette compatibilité est une compatibilité de format et non une compatibilité de détection/correction d'erreurs. Le BCH(144,128) x BCH(256,239) ne peut pas être décodé par le RS(255,239) du G.709. Toutefois, en prenant le code BCH(255,239) au lieu du RS(255,239) pour les colonnes de FEC et en ajoutant en surdébit les bits de parité supplémentaires, une trame OTN G.709 peut être transmise en utilisant ce code.

#### 4.1.2 Impact du surdébit

Le fait d'ajouter des bits à l'information transmise facilite l'opération du réseau et permet de corriger des erreurs de transmission, mais cela a également un impact sur le débit utile de transmission. Une trame OTN G.709 contient 16 320 bits, mais sur ces 16 320 bits, seulement 15 232 sont des bits d'information. Ainsi, pour transmettre 10 Gb/s d'information, il faut plutôt transmettre à 10,71 Gb/s. Ce 7% supplémentaire est appelé surdébit (*overhead*).

Avec le code BCH(144,128) x BCH(256,239), le surdébit est de 21% au lieu de 7%. En d'autres mots, pour transmettre 10 Gb/s d'information, il faut transmettre à 12,1 Gb/s.

Ce surdébit impose une modification de la bande passante du récepteur. Un filtre passe-bas ayant sa fréquence de coupure à 7 GHz n'est pas approprié pour un signal à 12,1 Gb/s. Une proportion trop importante du signal est coupée par le filtre. Ainsi, la bande passante du récepteur sera modifiée proportionnellement, pour être fixée à :

$$f_c = 0,7 \times 12,1 \text{GHz} = 8,47 \text{GHz} \quad (4.1)$$

C'est donc cette fréquence de coupure qui sera utilisée dans les simulations, plutôt que le 7 GHz habituel.

#### 4.1.3 Caractéristiques des codes BCH

Les codes utilisés seront de type BCH. Mais qu'en est-il exactement ? Quelles sont les caractéristiques de ces codes ? Et comment encoder les messages à transmettre ?

D'abord, pour l'histoire, les codes BCH ont été découverts indépendamment par Hocquenghem en 1959 et par Bose et Chaudhuri en 1960 [LIN et coll., 1983]. BCH est donc l'abréviation du nom de ses inventeurs. Les codes BCH sont des codes linéaires, plus particulièrement cycliques. Plusieurs algorithmes efficaces existent pour les décoder. Mentionnons l'algorithme itératif de Berlekamp [BERLEKAMP 1965] et l'algorithme de recherche de Chien [CHIEN 1964].

Pour tout entier positif  $m$  ( $m \geq 3$ ) et  $t$  ( $t < 2^{m-1}$ ), il existe un code BCH binaire ayant les paramètres suivants [LIN et coll., 1983] :

$$\text{Longueur du bloc} \quad : \quad n = 2^m - 1 \quad (4.2)$$

$$\text{Nombre de bits de parité} \quad : \quad n - k \leq mt \quad (4.3)$$

$$\text{Distance minimale} \quad : \quad d_{\min} \geq 2t + 1 \quad (4.4)$$

Un tel code est capable de corriger n'importe quelle combinaison présentant  $t$  erreurs ou moins.

Ainsi, la longueur d'un code BCH binaire doit être une puissance de 2 de laquelle on soustrait 1. Or, les codes BCH proposés dans l'article ne répondent pas à ce critère. En effet,

$256 \neq 2^m - 1$  pour tout  $m$  entier, et il en va de même pour 144. Comment obtenir ces codes particuliers à partir d'un code BCH de base ?

#### 4.1.4 BCH(256,239)

Le premier code à être utilisé est le BCH(255,239). En prenant  $m = 8$  et  $t = 2$ , on obtient un code BCH(255,239). Pour obtenir le code BCH(256,239), il suffit donc d'ajouter un bit de parité aux bits déjà obtenus par le BCH(255,239). Deux choix sont possibles : ajouter un bit pour que le total de bits à '1' donne un nombre pair (parité paire), ou l'ajouter pour que le total donne un nombre impair (parité impaire). Puisque l'article ne mentionne pas si le bit ajouté est de parité paire ou impaire, la parité paire a été choisie.

#### 4.1.5 BCH(144,128)

Une fois les lignes codées, il faut coder les colonnes suivant le code BCH(144,128). Ce code se base également sur le BCH(255,239). Pour le former, il ne faut pas oublier une caractéristique importante des codes BCH : ils sont systématiques. Cela signifie que le message à transmettre se retrouve en entier dans le mot codé. Par exemple, pour un message de 239 bits, les 255 bits seront les 239 bits du message auxquels sont ajoutés 16 bits de parité. Le code BCH(144,128) contient lui aussi 16 bits de parité. La stratégie est donc de prendre 128 bits et d'ajouter 111 bits à '0' (pour un total de 239 bits). Après avoir calculé les 16 bits de parité correspondant, les 111 bits à '0' sont retirés, d'où l'obtention d'un code BCH(144,128).

## 4.2 Exemple de décodage d'un turbo code

Pour mieux comprendre comment s'effectue le décodage d'un turbo code, mieux vaut débiter en examinant un exemple simple. D'abord, il est préférable de changer le code BCH(144,128) x BCH(256,239) à l'étude par un code similaire mais plus court. Le code BCH(7,4) x BCH(7,4) remplit ces critères. Pour le décodage, il est préférable d'utiliser d'abord un algorithme à décision binaire pour mieux comprendre le processus itératif.

La figure 4.3 présente le processus du codage de l'information. Les lignes sont d'abord codées suivant le code BCH(7,4), puis chaque colonne est obtenue en utilisant le même code.

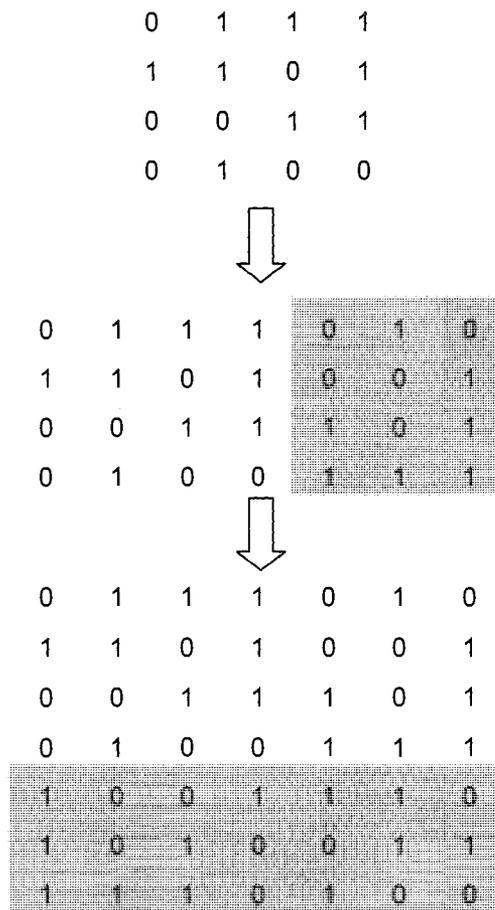


Figure 4.3 – Code BCH(7,4) x BCH(7,4) à transmettre

Avant de procéder au décodage, il faut d'abord que l'information soit transmise par le canal de communication. La figure 4.4 présente le code au récepteur après simulation pour une dispersion chromatique accumulée de 1400 ps/nm sans considérer le bruit ni augmenter la fréquence pour compenser le surdébit, en utilisant le seuil de décision optimal. Les bits en rouge sont en erreur.

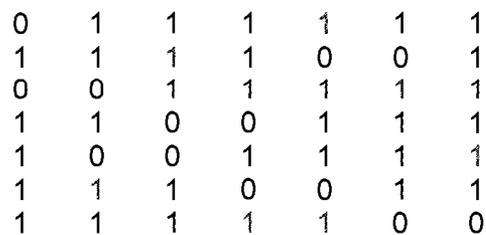


Figure 4.4 – Code BCH(7,4) x BCH(7,4) au récepteur

À la sortie du récepteur, il y a un total de 8 erreurs. Le code BCH(7,4) peut corriger jusqu'à une erreur. À partir du mot de code reçu, le décodeur trouve le code valide le plus proche, c'est-à-dire celui pour lequel il y a le moins de bits à changer. Ainsi, si le mot de code de 7 bits contient une seule erreur, le décodeur pourra la corriger, mais sinon, il existe un mot de code valide pour lequel un seul bit est à changer et c'est ce mot de code que le décodeur choisira. La figure 4.5 présente le code suite au décodage des lignes. Toutes les lignes, à l'exception de la première, ne comportaient qu'une seule erreur. Elles ont donc été décodées correctement. Toutefois, la première ligne contenait deux erreurs et le mot de code trouvé par le décodeur rajoute ainsi une erreur.

1	1	1	1	1	1	1
1	1	0	1	0	0	1
0	0	1	1	1	0	1
0	1	0	0	1	1	1
1	0	0	1	1	1	0
1	0	1	0	0	1	1
1	1	1	0	1	0	0

Figure 4.5 – Code BCH(7,4) x BCH(7,4) après une itération sur les lignes

L'avantage des codes turbo, c'est que le processus de décodage ne s'arrête pas là. Après le décodage suivant les lignes, il faut décoder suivant les colonnes. La figure 4.6 présente le résultat de cette opération. Puisque aucune colonne ne contenait plus d'une erreur, le décodeur a pu trouver le bon mot de code.

0	1	1	1	0	1	0
1	1	0	1	0	0	1
0	0	1	1	1	0	1
0	1	0	0	1	1	1
1	0	0	1	1	1	0
1	0	1	0	0	1	1
1	1	1	0	1	0	0

Figure 4.6 – Code BCH(7,4) x BCH(7,4) après une itération sur les colonnes

Cet exemple montre bien le pouvoir de correction accru du turbo code, comparativement à un code en bloc standard. Toutefois, le pouvoir de correction peut être accru encore davantage en

tenant compte de la valeur du signal reçu au récepteur avant la décision binaire. Cette amélioration, le décodage à décision pondérée, est abordée à la prochaine section.

### 4.3 Algorithme de Chase

Les informations présentées dans la présente section sont tirées de [CHASE 1972].

#### 4.3.1 Définitions préalables

Avant de détailler l'algorithme, il faut d'abord définir les variables utilisées. Soit  $\mathbf{I} = I_1, I_2, \dots, I_K$  la séquence binaire de  $K$  bits à coder. L'encodeur transforme ces  $K$  bits en  $N$  bits :  $\mathbf{X} = X_1, X_2, \dots, X_N$ . Le transmetteur transpose ce signal numérique en un signal analogique  $x(t)$ , qui est modifié par le canal de transmission. À la sortie du canal, on reçoit donc plutôt  $y(t)$ . Le récepteur déduit de ce signal la séquence des  $N$  bits reçus :  $\mathbf{Y} = Y_1, Y_2, \dots, Y_N$ . Il peut aussi ajouter de l'information supplémentaire sur la fiabilité de chaque bit :  $\alpha = \alpha_1, \alpha_2, \dots, \alpha_N$ . Finalement, le décodeur détermine  $\hat{\mathbf{X}}$ , le meilleur estimé de  $\mathbf{X}$  possible d'après les informations fournies par  $\mathbf{Y}$  et  $\alpha$ . Pour ce faire, il compare entre eux différents mots de code potentiels, nommés  $\mathbf{X}_m = X_{m1}, X_{m2}, \dots, X_{mN}$ . À chaque mot de code potentiel est associé un vecteur d'erreur  $\mathbf{Z}_m$ , qui contient des 1 aux endroits où  $\mathbf{Y}$  et  $\mathbf{X}_m$  diffèrent :  $\mathbf{Z}_m = \mathbf{Y} \oplus \mathbf{X}_m = Y_1 \oplus X_{m1}, Y_2 \oplus X_{m2}, \dots, Y_N \oplus X_{mN}$ . La figure 4.7 résume les différentes parties du système ainsi que les signaux qu'on y retrouve.

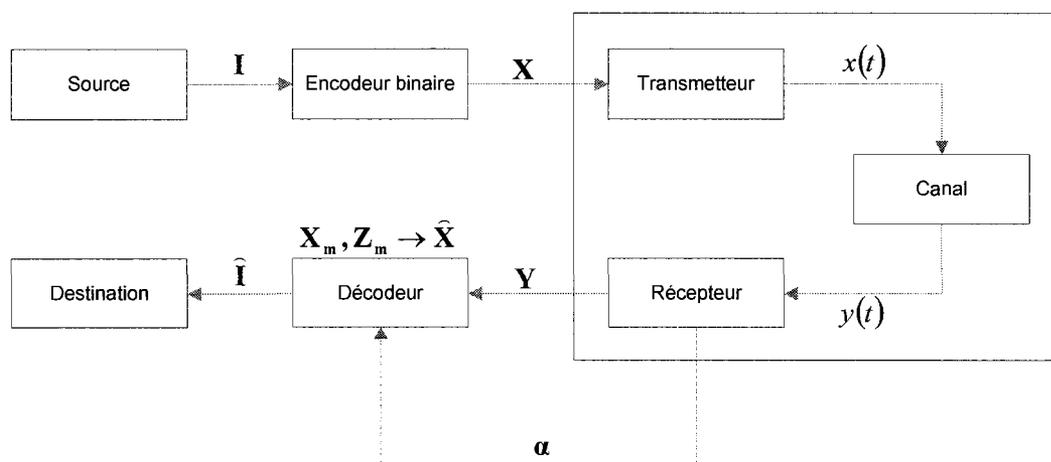


Figure 4.7 – Schéma du système de communication

### 4.3.2 Algorithme du décodeur

L'objectif du décodeur est de retrouver les bits émis  $\mathbf{X}$  à partir des bits reçus  $\mathbf{Y}$  et de l'information supplémentaire  $\alpha$ . Un décodeur binaire standard permet d'obtenir le mot de code  $\mathbf{X}_m$  qui diffère le moins de  $\mathbf{Y}$ , tant que cette distance n'est pas supérieure à  $\lfloor (d-1)/2 \rfloor$ , où  $d$  est la distance minimale entre deux mots de code. La notation  $\lfloor \ ]$  signifie que le résultat est arrondi à l'entier inférieur le plus près.

Soit  $W(\mathbf{Z}_m)$  le poids binaire de la séquence d'erreur  $\mathbf{Z}_m$ , avec  $\mathbf{Z}_m = \mathbf{Y} \oplus \mathbf{X}_m$  :

$$W(\mathbf{Z}_m) = \sum_{i=1}^N Z_{mi} \quad (4.5)$$

Alors un décodeur binaire standard produit le mot de code qui a le plus petit poids binaire.

Toutefois, le décodeur binaire standard ne tient pas compte de l'information supplémentaire  $\alpha$  sur la fiabilité de chacun des bits. Pour en tirer profit, le concept de poids analogique est introduit :

$$W_\alpha(\mathbf{Z}_m) = \sum_{i=1}^N \alpha_i Z_{mi} \quad (4.6)$$

Plus un bit  $i$  est fiable, plus la valeur de  $\alpha_i$  est élevée, et donc plus le poids analogique  $W_\alpha(\mathbf{Z}_m)$  augmente si le bit  $i$  est considéré en erreur. Ainsi, le mot de code le plus probable est celui qui possède le plus petit poids analogique, puisque les bits en erreur dans ce mot de code sont les moins fiables (valeur de  $\alpha$  faible).

Par conséquent, le décodeur de l'algorithme de Chase doit déterminer le mot de code qui satisfait  $\min_m W_\alpha(\mathbf{Y} \oplus \mathbf{X}_m)$ . Pour ce faire, Chase propose une démarche systématique présentée à la figure 4.8.

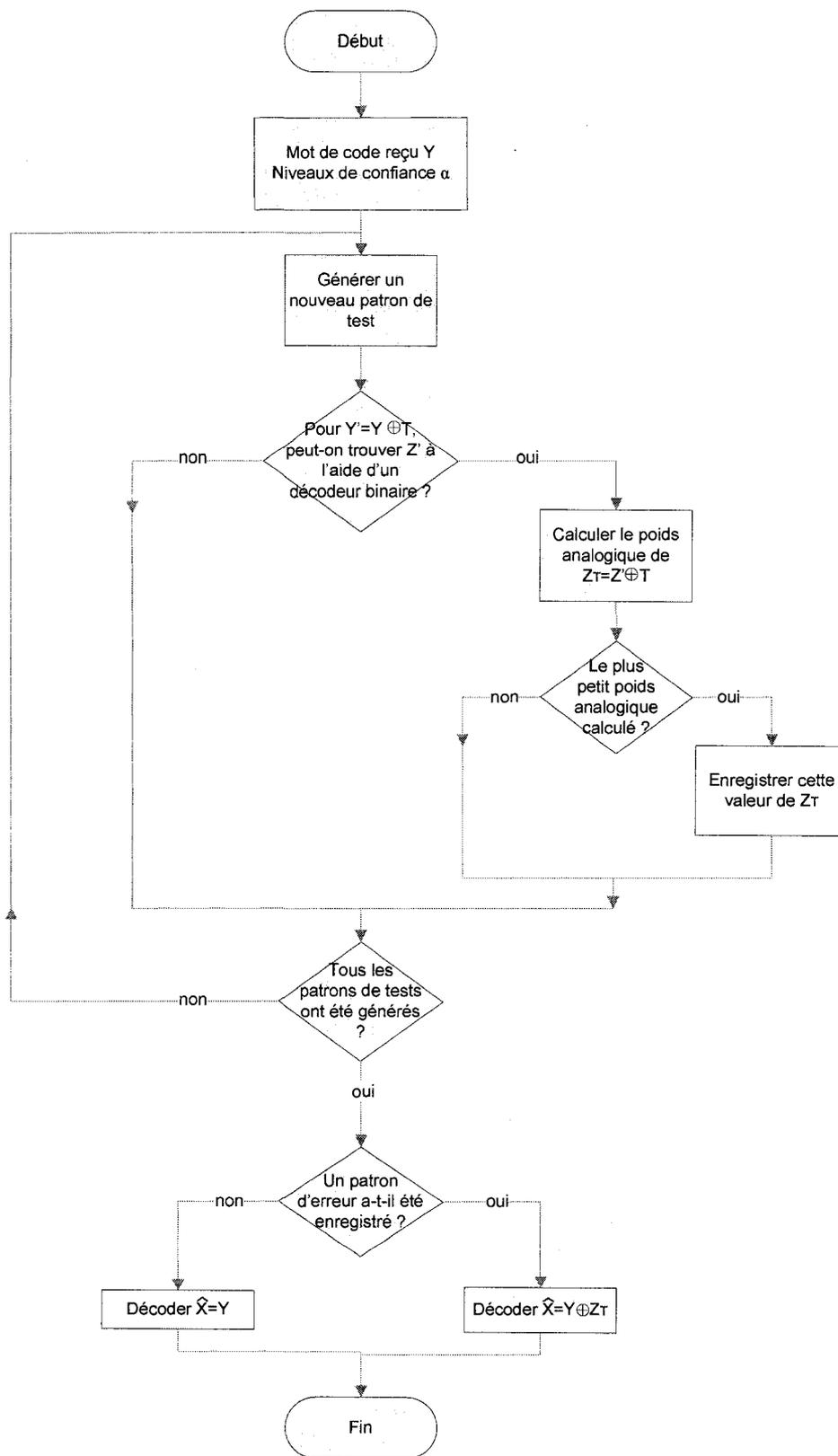


Figure 4.8 – Algorithme du décodeur

L'idée est de générer des patrons de test  $T$  afin de se promener dans l'ensemble des mots de code possibles. Un patron de test est une séquence de bits qui contient des '1' aux emplacements qu'on suppose en erreur et des '0' ailleurs. En appliquant un patron de test  $T$  à la séquence reçue  $Y$ , on obtient  $Y' = Y \oplus T$ , une séquence identique à  $Y$  sauf pour les bits supposés en erreur par le patron de test  $T$ . En décodant  $Y'$ , il est possible que le mot de code obtenu diffère de  $Y'$  en certains endroits, endroits donnés par le vecteur d'erreur  $Z'$ . Par rapport à la séquence reçue initiale  $Y$ , l'erreur est donc donnée par  $Z_T = T \oplus Z'$ . Le but est de trouver un patron de test  $T$  qui minimise le poids analogique de  $Z_T$ .

#### 4.3.3 Génération des patrons de test

L'algorithme de Chase repose sur la génération de patrons de test. Pour des codes courts, il serait envisageable de tester tous les patrons de test existants. Toutefois, puisque pour un code à  $N$  bits, il y a  $2^N$  possibilités, cela n'est pas envisageable pour les codes longs ou moyennement longs. De plus, cela ne serait pas optimal, certains patrons de test pouvant être éliminés d'office. Par exemple, le patron de test contenant uniquement des '1', donc suggérant que tous les bits sont en erreur, ne peut pas avoir le plus petit poids analogique.

Dans l'article, trois méthodes sont proposées pour limiter le nombre de patrons de test générés, en plus d'une modification applicable aux trois méthodes.

La première méthode est la plus simple, mais aussi la moins efficace. Son but est d'obtenir et de comparer entre eux tous les mots de code à une distance inférieure ou égale à  $(d - 1)$  de la séquence  $Y$ , où  $d$  est la distance minimale entre deux mots de code. Prendre comme patrons de test toutes les séquences ayant  $\lfloor d/2 \rfloor$  bits à '1' permet de s'en assurer. En effet, un décodeur binaire standard permet d'obtenir le mot de code le plus semblable jusqu'à une distance de  $\lfloor (d - 1)/2 \rfloor$ , alors en combinant l'effet du patron de test et du décodeur, tous les mots de code à une distance inférieure à  $(d - 1)$  seront trouvés.

La deuxième méthode réduit considérablement le nombre de séquences d'erreur testées. Au lieu d'accorder la même importance à tous les patrons d'erreur peu importe l'information obtenue sur la fiabilité des bits, on s'intéresse uniquement aux  $\lfloor d/2 \rfloor$  bits les moins fiables.

Ainsi, les patrons de test contiennent toute combinaison de '1' aux positions des  $\lfloor d/2 \rfloor$  bits les moins fiables. Il y a par conséquent  $2^{\lfloor d/2 \rfloor}$  patrons de test possibles.

La dernière méthode réduit le nombre de patrons de test considérés à  $\lfloor (d/2) + 1 \rfloor$ . Il est donc particulièrement utile pour les codes dont la distance minimale  $d$  est élevée, puisque le nombre de patrons de test augmente seulement linéairement avec  $d$ . Chaque patron de test possède  $i$  '1' aux positions des  $i$  bits les moins fiables. Pour  $d$  pair,  $i$  prend les valeurs  $i = 0, 1, 3, \dots, d-1$  et pour  $d$  impair,  $i = 0, 2, 4, \dots, d-1$ .  $i = 0$  correspond au patron de test nul, contenant uniquement des '0'.

Les trois méthodes présentées peuvent être modifiées par l'introduction d'un seuil. Ce seuil peut être basé sur le poids analogique d'un patron de test ayant  $k$  '1' aux positions des  $k$  bits les moins fiables,  $k$  pouvant aller de 0 jusqu'à  $N$ . Soit  $W_\alpha(T_k)$  le poids analogique d'un tel patron de test, alors  $\hat{X} = Y \oplus Z_T$  seulement si  $W_\alpha(Z_T) \leq W_\alpha(T_k)$ . Sinon,  $\hat{X} = Y$ . L'imposition d'un seuil peut être utile dans le cas d'un système ARQ (*Automatic Repeat Request*), en acheminant une demande de retransmission au-delà d'un certain seuil, ou dans le cas d'un décodage itératif.

#### 4.4 Calcul de la fiabilité

Les informations présentées ci-dessous proviennent de [PYNDIAH 1998]. Toutefois, les noms des variables ont été modifiés pour correspondre aux noms donnés dans l'algorithme de Chase et les formules finales ont été modifiées pour refléter que les bits prennent les valeurs  $\{0, 1\}$  et non  $\{-1, +1\}$  comme dans l'article.

L'article s'intéresse plus particulièrement au cas des codes turbos (appelés codes produits dans l'article) dans le cas d'un canal avec bruit blanc gaussien. Pour un tel canal, la décision optimale n'est pas le mot de code ayant le plus faible poids analogique, mais plutôt celui ayant la plus faible distance euclidienne. En d'autres termes :

$$\hat{\mathbf{X}} = \mathbf{X}_m \text{ si } |\mathbf{R} - \mathbf{X}_m|^2 \leq |\mathbf{R} - \mathbf{X}_l|^2 \quad \forall l \in [1, 2^k], l \neq m \quad (4.7)$$

où  $\mathbf{R} = r_1, r_2, \dots, r_N$  est le signal reçu à la sortie du canal.  $\mathbf{R}$  est différent du  $\mathbf{Y}$  utilisé dans l'algorithme de Chase, puisque  $\mathbf{Y}$  est la décision binaire résultant du signal  $\mathbf{R}$ .

Le carré de la distance euclidienne se calcule comme suit :

$$|\mathbf{R} - \mathbf{X}|^2 = \sum_{l=1}^N (r_l - x_l)^2 \quad (4.8)$$

Pour trouver la décision optimale  $\hat{\mathbf{X}}$ , il n'est pas souhaitable de mener une recherche exhaustive parmi tous les mots de code possibles. Il est préférable d'utiliser l'algorithme de Chase pour limiter le nombre de mots de code à comparer. Mais l'amélioration proposée par Pyndiah ne s'arrête pas là. En utilisant l'algorithme de Chase et le critère de la distance euclidienne pour trouver la décision optimale, on obtient en sortie un mot de code, une séquence binaire. Cela ne permet pas de profiter de la structure des turbo codes. En effet, après avoir décodé, par exemple, les lignes, l'information supplémentaire sur les bits reçus n'est plus accessible pour le décodage des colonnes.

Pour remédier à cette situation, l'article propose de calculer la fiabilité de la décision pour chacun des bits. Soit  $\hat{x}_j$  la décision prise au bit  $j$ . Il est probable qu'en cours de recherche, des mots de code ayant une valeur différente au bit  $j$  aient été examinés. Soit  $\mathbf{C}^{0(j)}$  le mot de code ayant la plus petite distance euclidienne parmi les mots de code où  $c_j = 0$ , et  $\mathbf{C}^{1(j)}$  celui parmi les mots de code où  $c_j = 1$ .

Alors la fiabilité de la décision au bit  $j$  est donnée par (voir annexe V) :

$$\Lambda'(\hat{x}_j) = \frac{1}{2\sigma^2} \left( |\mathbf{R} - \mathbf{C}^{0(j)}|^2 - |\mathbf{R} - \mathbf{C}^{1(j)}|^2 \right) \quad (4.9)$$

Pour l'étage suivant du décodeur, la nouvelle valeur du bit à utiliser se calcule comme suit :

$$r'_j = 0,5 \left( |\mathbf{R} - \mathbf{C}^{0(j)}|^2 - |\mathbf{R} - \mathbf{C}^{1(j)}|^2 \right) + \text{seuil} \quad (4.10)$$

L'information extrinsèque  $w_j$  est la variation de  $r_j$  causée par la connaissance de la fiabilité de la décision :  $r'_j = r_j + w_j$ .

Pour réduire l'effet de l'information extrinsèque dans les premières étapes du décodage, étapes pour lesquelles le BER est encore relativement élevé, le facteur  $\alpha$  est introduit. Il a une faible valeur lors des premières itérations et augmente au fur et à mesure que le taux d'erreur diminue. Soit  $\alpha(m)$  la valeur du facteur  $\alpha$  à l'étape  $m$  du décodeur. Dans son article, Pyndiah utilise :

$$\alpha(m) = [0; 0.2; 0.3; 0.5; 0.7; 0.9; 1; 1] \quad (4.11)$$

La figure 4.9 montre un étage du décodeur à décision pondérée tel que décrit dans [PYNDIAH 1998]. On y remarque un paramètre qui n'a pas encore été introduit, le paramètre  $\beta$ . Ce paramètre est nécessaire puisqu'il est probable que pour certains bits, aucun mot de code où  $c_j \neq \hat{x}_j$  ne sera examiné. Dans un tel cas, la sortie est approximée par :

$$r'_j = \text{seuil} + \beta(\hat{x}_j - 0,5) \quad (4.12)$$

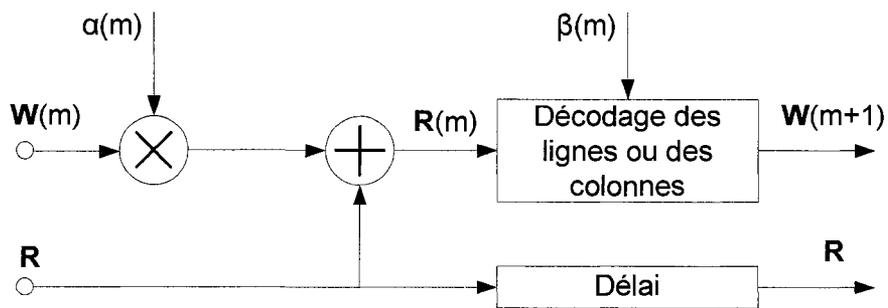


Figure 4.9 – Étage d'un décodeur à décision pondérée

L'ajout du paramètre  $\beta$  semble approximatif, mais il est justifié. Il faut comprendre que si aucun mot de code alternatif n'est trouvé, alors ce dernier est probablement très loin de  $\mathbf{R}$  en terme de distance Euclidienne. Dans ce cas, la probabilité que la décision  $\hat{x}_j$  soit correcte est relativement élevée, et donc la fiabilité de  $\hat{x}_j$  est également relativement élevée. Dans son article, Pyndiah fixe préalablement la valeur de  $\beta$  :

$$\beta(m) = [0, 2; 0, 4; 0, 6; 0, 8; 1; 1; 1; 1] \quad (4.13)$$

Les valeurs de  $\beta$  sont indépendantes des valeurs de  $\alpha$ . Des résultats expérimentaux indiquent que les performances ne sont pas dégradées significativement si les valeurs de  $\beta$  sont modifiées de  $\pm 10\%$  [PYNDIAH 1998].

Finalement, une modification est nécessaire pour éviter qu'après un certain nombre d'itérations, des bits décodés correctement deviennent en erreur. Par exemple, soit, pour un bit donné,  $r_j = 0,74$  et  $w_j(m) \times \alpha(m) = 0,53$ , alors  $r_j(m) = 1,27$ . Si suite au décodage,  $r'_j = 1$ , on aura  $w_j(m+1) = -0,27$ . Ainsi, suite au décodage, l'information extrinsèque réduirait la fiabilité du bit alors qu'au contraire, elle devrait augmenter ou rester la même. Pour éviter ce problème, les valeurs de  $R(m)$  sont limitées entre 0 et 1.

#### 4.5 Résultats de simulation

Pour déterminer les performances du décodeur, il faut obtenir les deux graphiques typiques, soit le taux d'erreur (BER) en fonction du rapport signal sur bruit (OSNR) pour une dispersion chromatique accumulée donnée et le BER en fonction de la dispersion chromatique accumulée pour un OSNR donné.

Toutefois, il faut d'abord régler quelques considérations pratiques. D'abord, Chase propose trois méthodes pour limiter la quantité de patrons de test générée. Il faut donc choisir laquelle utiliser. Ensuite, la puissance des turbo codes réside dans le procédé itératif de décodage. Après combien d'itérations les résultats cessent-ils de s'améliorer ? Finalement, compte tenu du phénomène de la dispersion chromatique, est-il préférable de débiter par le décodage des lignes ou par celui des colonnes ? Ce n'est qu'une fois ces questions résolues que pourront être obtenus les graphiques de la performance du décodeur.

##### 4.5.1 Méthode de génération des patrons de test à privilégier

La première méthode proposée par Chase est de considérer toutes les séquences ayant  $\lfloor d/2 \rfloor$  bits à '1'. Pour un code BCH(255,239),  $d = 5$  ou 6. Par conséquent, il faudrait tester toutes les séquences ayant 2 bits à '1'. Au total, il y aurait donc 32 385 séquences à considérer en considérant  $d = 5$ . Or, décoder une seule séquence prend entre 1 et 1,5 secondes en

employant le logiciel Matlab sur un ordinateur personnel typique. Même en prenant 1 seconde par séquence, décoder demanderait 9 heures. Il est donc peu pratique d'utiliser cette méthode.

La deuxième méthode réduit considérablement le nombre de séquences étudiées. Il faut considérer toute combinaison de '1' aux positions des  $\lfloor d/2 \rfloor$  bits les moins fiables, pour un total de  $2^{\lfloor d/2 \rfloor}$  séquences. Puisque  $d = 5$  ou  $6$ , il y a donc 4 ou 8 séquences à tester. Puisque la distance minimale exacte est inconnue, il est préférable de prendre  $d = 6$  et de tester 8 séquences. De plus, cela devrait augmenter le nombre de bits pour lesquels il existe un mot de code compétiteur.

Quant à la troisième méthode, le nombre de patrons de test considérés est réduit à  $\lfloor (d/2) + 1 \rfloor$ , soit 4 si l'on considère  $d = 6$ . Puisque  $d$  est pair, chaque patron de test possède  $i$  '1' aux positions des  $i$  bits les moins fiables, avec  $i = 0,1,3,5$ .

Pour comparer l'efficacité des deux méthodes, il faut les simuler en utilisant la même séquence test. Puisque la différence entre les méthodes réside dans l'information extrinsèque trouvée, l'impact devrait se faire sentir après plusieurs itérations. Le tableau 4.1 présente les résultats pour un code test après 735 ps/nm de dispersion chromatique accumulée et un OSNR de 15 dB. En utilisant le seuil de décision optimal, le taux d'erreur sans décodeur est de 0,0232.

Tableau 4.1 – Comparaison des méthodes de génération des patrons de test

Méthode	Temps de simulation	Taux d'erreur après :			
		1 itération	2 itérations	3 itérations	4 itérations
2 <sup>e</sup>	16 735 s	0,0161	0,0110	0,0089	0,0093
3 <sup>e</sup>	8 550 s	0,0201	0,019	0,0241	0,0285

Bien qu'elle soit beaucoup plus rapide, la 3<sup>e</sup> méthode ne donne pas de bons résultats. Vu le nombre restreint de patrons de test générés, il survient plus fréquemment qu'aucun code valide n'est trouvé. Dans de tels cas, aucune information extrinsèque n'est obtenue, ce qui nuit aux performances. Par conséquent, la 2<sup>e</sup> méthode est retenue.

#### 4.5.2 Nombre d'itérations nécessaire

L'objectif est d'obtenir des résultats satisfaisants en un temps raisonnable. Pour 4 itérations du décodeur avec la méthode choisie, les simulations durent 16 735 secondes, soit 4 heures 40 minutes environ. Même si un seul bloc de 36 864 bits est décodé pour chaque distance et chaque OSNR considéré, le temps de simulation total devient trop long. Ainsi, il est préférable de limiter le nombre d'itérations à 2.

#### 4.5.3 Ordre du décodage

Dernière considération : est-il préférable de débiter par le décodage des lignes ou par celui des colonnes ? Dans un canal affecté uniquement par du bruit blanc gaussien, l'ordre du décodage est sans importance, mais qu'en est-il pour un canal affecté par la dispersion chromatique ?

Pour le savoir, la séquence de test utilisée pour déterminer la meilleure méthode de génération des patrons de test a été réutilisée. Les résultats sont présentés au tableau 4.2.

Tableau 4.2 – Résultats en fonction de l'ordre du décodage

Méthode	Taux d'erreur après :	
	1 itération	2 itérations
Lignes avant les colonnes	0,0161	0,0110
Colonnes avant les lignes	0,0152	0,0078

D'après les résultats, il est préférable de décoder les colonnes avant les lignes. Une explication possible est que les bits sont transmis par le canal ligne par ligne. Ainsi, à cause de l'interférence inter-symbole, il y a une plus forte corrélation entre les bits d'une ligne que les bits d'une colonne. Or, les codes correcteurs d'erreur sont en général moins performants pour corriger des erreurs en salves que pour corriger des erreurs aléatoires. Une solution courante au problème des erreurs en salves est l'emploi de l'entrelacement. L'entrelacement consiste à modifier l'ordre de transmission des bits. Est-ce que l'emploi d'une méthode d'entrelacement pourrait permettre d'obtenir de meilleurs résultats que le décodage des colonnes avant les lignes ?

Pour répondre à cette question, deux méthodes d'entrelacement sont testées. La figure 4.10 présente la première, l'entrelacement des colonnes. Suite à la transmission des bits, l'opération inverse, le désentrelacement, permet de replacer les colonnes dans leur ordre initial.

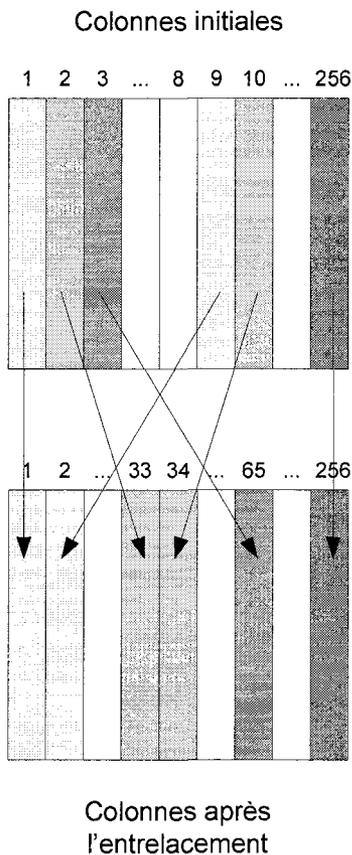


Figure 4.10 – Entrelacement des colonnes

Une autre façon de modifier l'ordre de transmission des bits est de modifier la position de chacun des bits, et non seulement la position des colonnes. Soit  $n$  la position d'un bit donné, en considérant que pour les bits de la première colonne,  $n$  va de 1 à 144, et  $n$  va de 145 à 288 pour la deuxième colonne, et ainsi de suite pour les autres colonnes. L'équation suivante permet d'obtenir  $n'$ , la position du bit  $n$  suite à l'entrelacement.

$$n' = (1 + 145(n - 1)) \bmod 36864 + 1 \quad (4.14)$$

La figure 4.11 présente de façon graphique le résultat de l'équation (4.14).

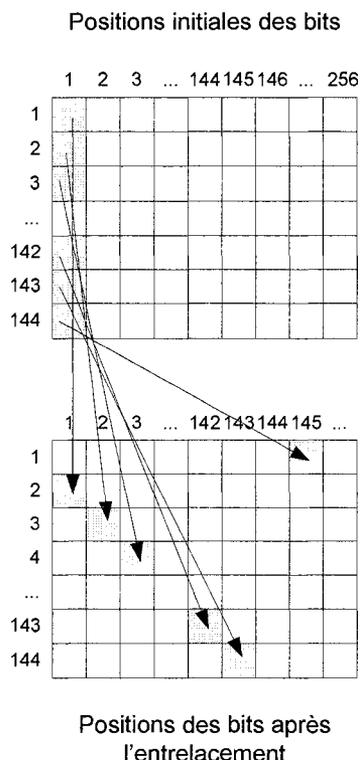


Figure 4.11 – Entrelacement complet

Laquelle des deux méthodes proposées produit les meilleurs résultats ? Pour le déterminer, le code test utilisé précédemment est repris, avec une dispersion chromatique accumulée de 735 ps/nm et un OSNR de 15 dB. Toutefois, il faut reprendre la simulation du canal pour chacune des méthodes d'entrelacement, puisque l'ordre de transmission des bits change. Par conséquent, le bruit, bien qu'il ait la même puissance moyenne, n'a pas exactement la même valeur pour un bit donné d'une simulation à l'autre. Le tableau 4.3 présente les résultats des simulations. La colonne « 0 itération » présente le taux d'erreur immédiatement à la sortie du canal, avant que le décodage soit entrepris.

Tableau 4.3 – Résultats en utilisant différents types d'entrelacement

Méthode	Taux d'erreur après :		
	0 itération	1 itération	2 itérations
Colonnes avant les lignes	0,0232	0,0152	0,0078
Entrelacement des colonnes	0,0215	0,0119	0,0025
Entrelacement complet	0,0232	0,0119	0,0045

Suite aux résultats du tableau 4.3, la méthode d'entrelacement des colonnes est choisie. Les résultats sont comparables à ceux obtenus pour l'entrelacement complet, mais la méthode est beaucoup plus simple à mettre en œuvre.

#### 4.5.4 Résultats

Il est maintenant possible d'obtenir les performances du décodeur. La figure 4.12 présente le BER en fonction de la dispersion chromatique accumulée pour différents rapports signal sur bruit. Il y a trois parties distinctes dans les performances du décodeur. Dans la première partie, lorsque la dispersion chromatique accumulée est élevée, le décodeur ne réussit pas à corriger les erreurs, peu importe le rapport signal sur bruit. Le nombre d'erreurs initial est trop élevé pour le pouvoir de correction du code. Puis, dans une seconde partie, le décodeur réussit à décoder plusieurs erreurs, mais pas toutes. À partir d'un certain seuil toutefois, qui semble être autour de 735-770 ps/nm de dispersion chromatique, toutes les erreurs sont corrigées. Cette partie n'est pas visible à la figure 4.12, car un taux d'erreur nul ne peut pas être affiché sur une échelle logarithmique.

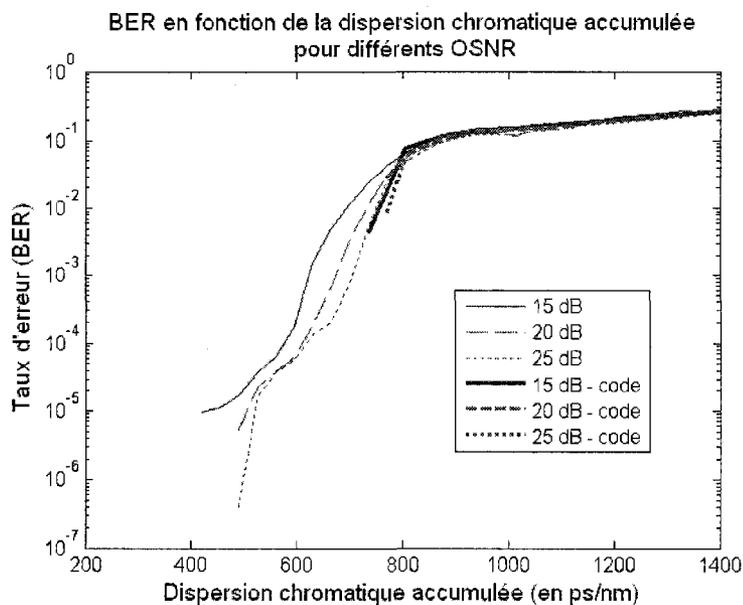


Figure 4.12 – Performances du décodeur

À la figure 4.12, les performances du décodeur sont comparées avec les performances d'un canal à 12,1 Gb/s. Il y est montré qu'à quantité égale de dispersion chromatique accumulée, en

deçà d'un certain seuil, le décodeur permet de diminuer considérablement le taux d'erreur. Or, en pratique, il faudrait plutôt comparer à distance égale plutôt qu'à quantité égale de dispersion chromatique accumulée. L'introduction du code correcteur d'erreur oblige à augmenter le débit de transmission à 12,1 Gb/s, si bien que pour une même distance, l'impact de la dispersion chromatique sur l'interférence inter-symbole est plus grande suite à l'ajout du décodeur. Puisque l'on ne souhaite pas avoir à diminuer la distance de propagation pour introduire le code correcteur d'erreur, il faut donc obtenir le « net coding gain » plutôt que le « coding gain ».

Le problème est qu'avec les résultats obtenus en pratique, il est difficile d'obtenir le « net coding gain » en comparant, à distance égale, les résultats à 10 Gb/s avec les résultats à 12,1 Gb/s avec décodeur. D'abord, pour les distances où la dispersion chromatique accumulée est supérieure au seuil du décodeur, le décodeur n'est pas efficace et donc les résultats sont forcément meilleurs à 10 Gb/s. Ensuite, il y a très peu de tests pour lesquels le décodeur a corrigé des erreurs sans aboutir à un taux d'erreur nul. En fait, les taux d'erreur ne devraient pas être nuls en réalité. Or, vu les temps de simulation importants, des taux d'erreur très faibles ne peuvent pas être obtenus en un temps raisonnable. Ainsi, il y a très peu de points à comparer entre les simulations à 10 Gb/s et celles à 12,1 Gb/s avec décodeur. Finalement, les résultats de simulation du canal ont montré que le modèle utilisé semble beaucoup plus dispersif que les fibres utilisées en pratique. Par conséquent, l'impact du surdébit est probablement surévalué. L'interférence inter-symbole supplémentaire causée par le surdébit est, de ce fait, plus importante que ce qu'elle devrait être. Ainsi, même si des résultats pouvaient être obtenus, ils ne représenteraient pas les performances réelles du décodeur.

Ces constatations ramènent à l'objectif principal, qui est de déterminer si la modification qui sera proposée au chapitre 5 améliore les performances du décodeur. Puisque la modification sera elle aussi appliquée avec un débit de 12,1 Gb/s, les comparaisons pourront se faire plus facilement.

#### **4.6 Résumé**

Le décodeur initial repose sur l'utilisation du code turbo BCH(144,128) x BCH(256,239). Ce code est conforme à la norme G.709 quant au format de trame et les chercheurs de Mitsubishi ont obtenu de très bons résultats en l'employant. Vu l'ajout de bits de redondance, il faut par contre transmettre à 12,1 Gb/s pour atteindre un débit de 10 Gb/s d'information.

L'avantage principal des turbo codes est que l'information obtenue lors du décodage des colonnes permet d'améliorer le décodage des lignes, et vice-versa. Le décodage à décision pondérée est une amélioration supplémentaire, car elle permet de prendre une décision en fonction de la fiabilité de chacun des bits. L'algorithme de Chase repose sur la génération de patrons de test et le calcul de la fiabilité, introduit par Pyndiah, ajoute l'idée d'information extrinsèque.

Les simulations montrent que, jusqu'à un certain seuil pour la dispersion chromatique accumulée, le décodeur permet de corriger toutes les erreurs, ou la plupart d'entre elles. Toutefois, lorsque la dispersion chromatique accumulée est supérieure au seuil, le décodeur donne des résultats semblables à un canal sans code correcteur d'erreurs.

Les résultats de simulation ne permettent pas de comparer les performances avec et sans l'ajout du code correcteur d'erreur. Toutefois, l'objectif n'est pas de démontrer que le code correcteur d'erreur réduit considérablement le taux d'erreur, mais plutôt que la modification proposée améliore les performances du décodeur. En ce sens, les résultats obtenus sont suffisants pour atteindre l'objectif. Le prochain chapitre porte d'ailleurs sur la dernière étape pour atteindre cet objectif : l'algorithme de contrôle d'erreurs proposé.

## CHAPITRE 5

### ALGORITHME DE CONTRÔLE D'ERREURS

Dans les chapitres précédents, les problèmes de la simulation du canal, de l'obtention du taux d'erreur et de l'algorithme de base du décodeur à décision pondérée ont été réglés. Tous les éléments sont maintenant en place pour élaborer la solution proposée, soit un algorithme de contrôle d'erreur appliqué au phénomène de dispersion chromatique.

Puisque l'algorithme final vise à tenir compte de l'effet de la dispersion chromatique sur la distribution des valeurs à la sortie du canal, la section 5.1 s'intéressera à la valeur à la sortie du canal en fonction de la séquence de bits à l'entrée. C'est à la section 5.2 que sera expliqué le cœur du système, soit l'algorithme de prétraitement du signal qui tiendra compte des résultats de la section 5.1 pour réduire davantage le taux d'erreur. Les résultats de simulation en conditions normales seront présentés à la section 5.3, tandis que la section 5.4, montrera l'impact d'un changement de condition, soit une modification du bruit ou de la dispersion chromatique accumulée, sur les performances du système. Une dernière section, la section 5.5, proposera un aperçu d'ensemble du système tel qu'il pourrait être implanté en pratique.

#### 5.1 Sortie du canal en fonction de la séquence de bits

La dispersion chromatique fait en sorte que l'énergie d'un bit s'étale sur les bits adjacents, donc à la fois sur les bits suivants et les bits précédents. Par conséquent, la valeur d'un bit à la sortie du canal dépend autant de la séquence de bits qui le suit que de celle qui le précède. Puisque le code requiert un fonctionnement à 12,1 Gb/s pour transmettre 10 Gb/s d'information, les valeurs doivent être obtenues en fonction d'un canal à 12,1 Gb/s.

D'abord, il faut se demander combien de bits il est nécessaire de considérer simultanément. La figure 5.1 présente la distribution statistique des '1' et des '0' avec une dispersion chromatique accumulée de 1400 ps/nm. Chaque courbe indique, en fonction du bit précédent et du bit suivant, la probabilité que la puissance moyenne du bit central ait une certaine valeur.

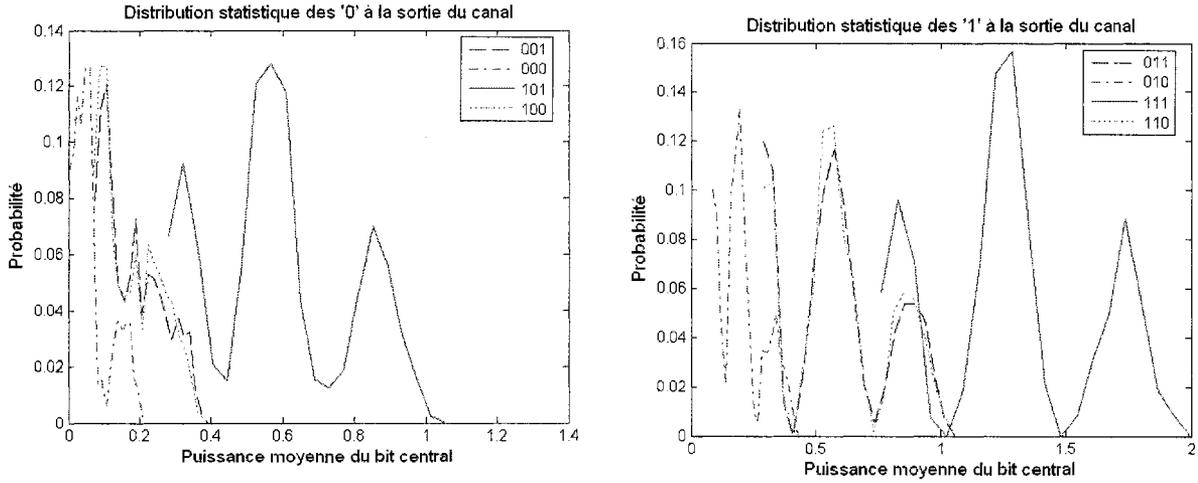


Figure 5.1 – Distribution statistique des '1' et des '0' – 1 bit avant et après

La figure 5.1 permet d'abord de constater l'importance de l'impact de la séquence de bits sur la valeur à la sortie du canal. Elle montre également l'existence d'une certaine symétrie : pour les '0', la distribution statistique de '001' est semblable à celle de '100' et pour les '1', la distribution de '011' est semblable à celle de '110'. Cette observation confirme que l'énergie des bits s'étale à la fois sur les bits suivants et sur les bits précédents. De plus, la plupart des distributions semblent morcelées. Cette observation laisse supposer que ces distributions sont constituées de sous-distributions, d'où l'intérêt d'investiguer la distribution statistique en ne considérant non pas un, mais deux bits de chaque côté du bit émis. La figure 5.2 présente les résultats obtenus.

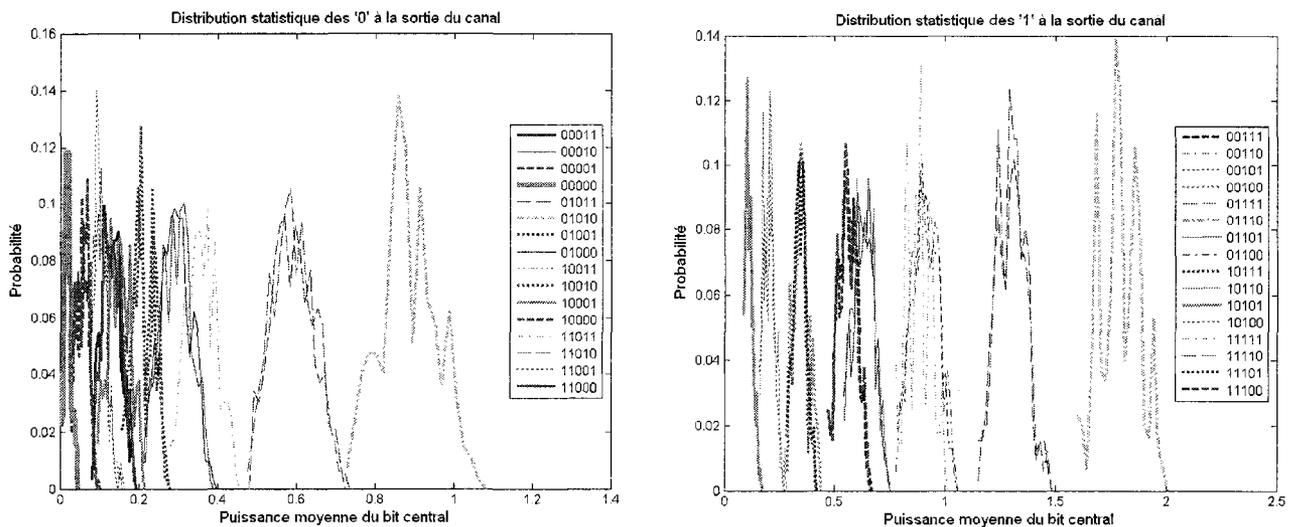


Figure 5.2 – Distribution statistique des '1' et des '0' - 2 bits avant et après

Considérer 3 bits avant et 3 bits après n'augmente pas considérablement la précision, et puisque la complexité augmente avec le nombre de bits considérés, il est préférable de limiter les résultats aux deux bits précédents et aux deux bits suivants.

## 5.2 Explication de l'algorithme de prétraitement

Le but de l'algorithme de prétraitement est de diminuer le taux d'erreur en complexifiant le processus de décision. Au lieu de comparer simplement la valeur reçue avec un seuil de décision préétabli, la décision est prise en fonction à la fois de la valeur reçue et des valeurs reçues pour les bits environnants. En ce sens, la méthode proposée se rapproche de la technique MLSE présentée au chapitre 1. Dans le but d'être compatible avec un algorithme de décodage à décision pondérée, l'algorithme doit produire en sortie une valeur dépendant de la fiabilité de la décision prise.

À la section 5.1, le choix a été fait de limiter l'analyse aux deux bits précédents et aux deux bits suivants. Chaque bit transmis appartient donc à l'une des  $2^5=32$  séquences possibles. En utilisant les résultats de la figure 5.2, il est possible de déterminer quelles séquences sont les plus probables d'avoir été envoyées. Par exemple, si la puissance moyenne d'un bit vaut 1,8, il est très probable que le bit soit un '1', précédé de '01' et suivi de '10'. Toutefois, pour certaines valeurs à la sortie, plusieurs séquences sont probables. Dans ces cas, comment déterminer la séquence la plus probable ?

Une approche possible est de calculer, pour chacun des bits, sa probabilité d'appartenance à chacune des séquences. En obtenant la distribution statistique des valeurs à la sortie du canal, il est possible d'obtenir la valeur minimale  $P_{min}$  et maximale  $P_{max}$  que peut prendre la sortie du canal en fonction de la séquence et de la dispersion chromatique accumulée. Sans aucun bruit, la probabilité d'appartenance de la valeur  $y_j$  à une séquence  $k$  serait de 1 si  $P_{min}^k \leq y_j \leq P_{max}^k$  et de 0 autrement. Or, puisque le bruit est non nul, il est possible que  $y_j$  appartienne à la séquence  $k$  sans que  $P_{min}^k \leq y_j \leq P_{max}^k$ . Soit  $\sigma$  l'écart-type du bruit, alors :

$$P[y_j \in k] = \begin{cases} e^{-\frac{(y_j - P_{max}^k)^2}{2\sigma^2}} & \text{si } y_j \geq P_{max}^k \\ 1 & \text{si } P_{min}^k \leq y_j \leq P_{max}^k \\ e^{-\frac{(y_j - P_{min}^k)^2}{2\sigma^2}} & \text{si } y_j \leq P_{min}^k \end{cases} \quad (5.1)$$

Cette équation est représentée graphiquement à la figure 5.3.

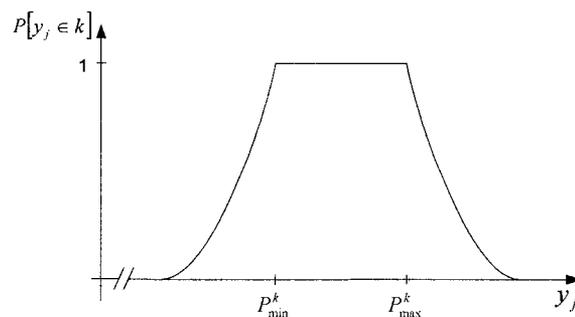


Figure 5.3 – Courbe d'appartenance à la séquence  $k$

Pour les premiers bits reçus, un calcul particulier est requis. En effet, les deux bits précédents l'envoi du premier bit sont connus. Ainsi, la probabilité d'appartenir à toute séquence  $k$  pour laquelle un de ces bits diffère est nulle.

La méthode requiert la connaissance de l'écart-type du bruit et de la dispersion chromatique accumulée. En théorie, l'opérateur d'un réseau de communication par fibre optique connaît ces valeurs. Il connaît le rapport signal sur bruit prévu à la réception et la longueur du lien. Toutefois, en pratique, les conditions exactes fluctuent dans le temps. C'est pourquoi un algorithme adaptatif permettant d'évaluer les conditions du canal est nécessaire. L'impact d'une variation de ces conditions sur les performances de l'algorithme de prétraitement sera évalué à la section 5.5.

Le calcul de probabilité permet d'écarter les séquences improbables, mais une valeur  $y_j$  à la sortie du canal peut encore appartenir à plusieurs séquences différentes. Il faut donc également tenir compte de l'information disponible pour les bits environnants.

Si, par exemple,  $y_j$  vaut 1, il peut appartenir aux séquences '11111', '00110', '01100' et '01010'. Pour déterminer si la séquence '11111' est la bonne, il faut voir si  $y_{j-1}$ , le bit juste avant  $y_j$ , appartient à '01111' ou à '11111'. Sinon, il est impossible que la séquence '11111' soit la bonne. Il faut également vérifier que  $y_{j-2}$  appartient à '00111', '01111', '10111' ou '11111', et pareillement pour  $y_{j+1}$  et  $y_{j+2}$ . Sous forme d'équation, on obtient :

$$P_j[11111] = P[y_{j-2} \in 'xx111'] \times P[y_{j-1} \in 'x1111'] \times P[y_j \in '11111'] \times P[y_{j+1} \in '1111x'] \times P[y_{j+2} \in '111xx'] \quad (5.2)$$

où :

$$P[y_{j-2} \in 'xx111'] = P[y_{j-2} \in '00111'] + P[y_{j-2} \in '01111'] + P[y_{j-2} \in '10111'] + P[y_{j-2} \in '11111']$$

$$P[y_{j-1} \in 'x1111'] = P[y_{j-1} \in '01111'] + P[y_{j-1} \in '11111']$$

(5.3)

et par symétrie :

$$\begin{aligned}
 P[y_{j+1} \in '1111x'] &= P[y_{j+1} \in '11110'] + P[y_{j+1} \in '11111'] \\
 P[y_{j+2} \in '111xx'] &= P[y_{j+2} \in '11100'] + P[y_{j+2} \in '11101'] + P[y_{j+2} \in '11110'] + P[y_{j+2} \in '11111']
 \end{aligned}
 \tag{5.4}$$

Bref, il faut déterminer si l'appartenance d'un bit à une séquence est compatible avec l'appartenance des bits environnants aux séquences liées.

Le même calcul est répété pour chacune des 32 séquences possibles, pour chacun des bits. À la sortie, il faut obtenir une valeur qui indique à la fois si le bit est un '1' et un '0' et la fiabilité  $r$  de la décision prise. Soit  $r'_j$  la valeur à la sortie pour le bit  $j$ , alors :

$$r'_j = \frac{\sum P_j['xx1xx']}{\sum P_j[k]}
 \tag{5.5}$$

En mots,  $r'_j$  est la somme des probabilités obtenues par l'équation (5.2) où le bit central est un '1', divisée par la somme de toutes les probabilités obtenues par l'équation (5.2). En utilisant cette équation,  $r'_j$  varie entre 0 et 1. Il vaut 0,5 si le '1' et le '0' sont équiprobables, et s'approche de 0 ou de 1 si la probabilité de l'un des bits est plus grande que l'autre.

### 5.3 Résultats de simulation

L'algorithme de prétraitement est d'abord utilisé seul, sans ajout du code correcteur d'erreur et du décodeur. Les figures 5.4 et 5.5 comparent les résultats obtenus à un débit de 12,1 Gb/s avec et sans l'algorithme de prétraitement.

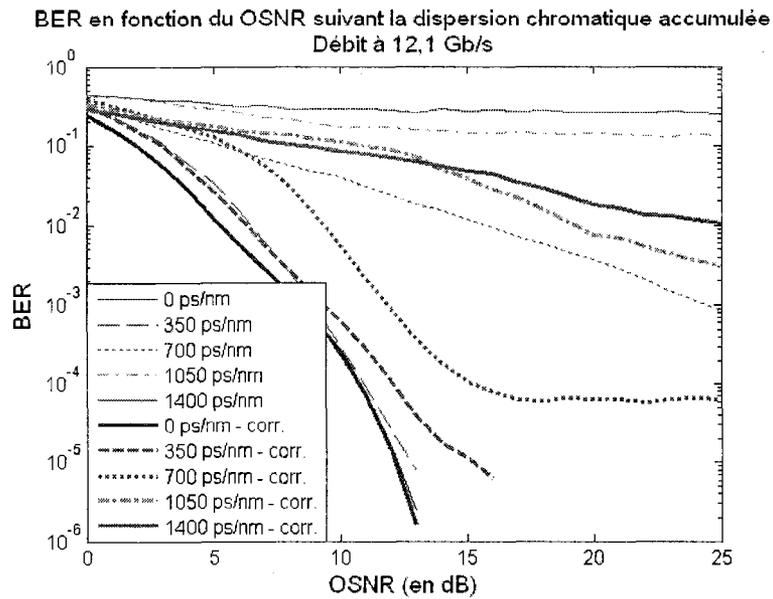


Figure 5.4 – BER en fonction du OSNR, algorithme de prétraitement employé seul

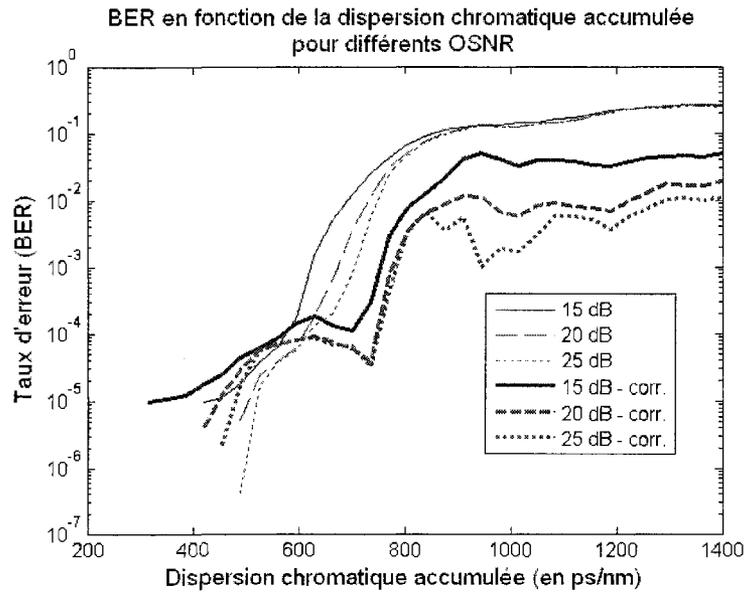


Figure 5.5 – BER en fonction de la distance, algorithme de prétraitement employé seul

Les résultats montrent que lorsque la dispersion chromatique accumulée est faible, l'algorithme de prétraitement ne permet pas de diminuer le taux d'erreur et peut même l'augmenter. Par

contre, lorsque la dispersion chromatique accumulée est plus élevée, l'amélioration est significative. De plus, plus le rapport signal sur bruit est élevé, meilleurs sont les résultats.

Le but de l'algorithme présenté est d'être utilisé comme prétraitement avant l'application du code correcteur d'erreur. Il faut donc vérifier si l'ajout de l'algorithme de prétraitement améliore les performances du décodeur, performances obtenues au chapitre 4. Il faut rappeler que le décodeur donne de bons résultats lorsque la dispersion chromatique accumulée est faible, alors qu'à l'inverse, l'algorithme de prétraitement performe mieux lorsque la dispersion chromatique est élevée. L'association de ces deux méthodes complémentaires permet-elle de réduire le taux d'erreur ?

La figure 5.6 permet de comparer entre elles les performances des différentes méthodes de correction. On y remarque que le décodeur a un comportement similaire avec ou sans prétraitement, c'est-à-dire qu'il n'améliore pas les performances tant que le taux d'erreur est au dessus d'un certain seuil. Toutefois, grâce à l'algorithme de prétraitement, le seuil est atteint pour une plus grande quantité de dispersion chromatique accumulée. Ainsi, pour un même taux d'erreur, la transmission peut se faire sur une plus longue distance.

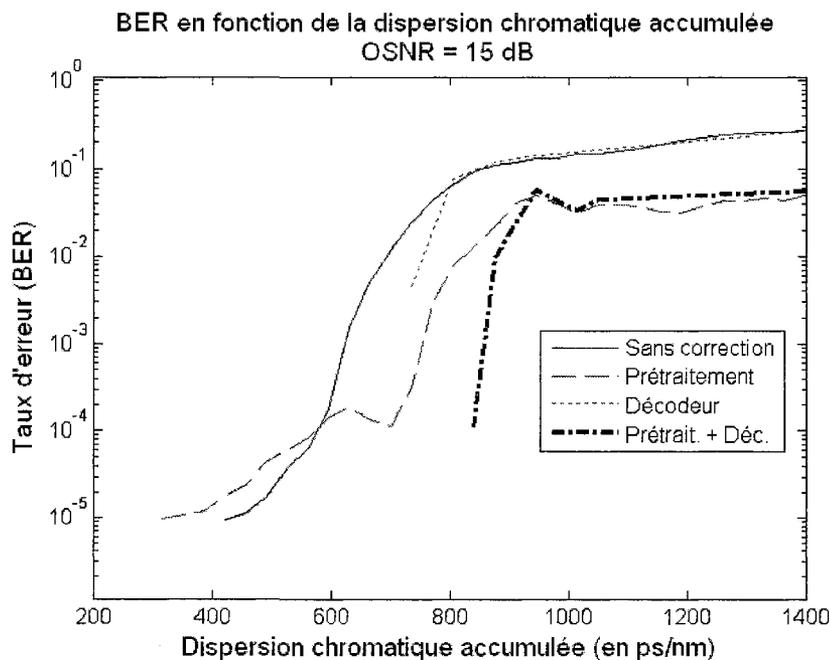


Figure 5.6 – Comparaison des performances

Les simulations effectuées jusqu'à présent supposent que les conditions exactes du canal sont connues en tout temps. En effet, le calcul de la probabilité d'appartenance de l'équation (5.1) nécessite de connaître ou d'estimer  $P_{\min}^k$  et  $P_{\max}^k$ , qui dépendent de la dispersion chromatique accumulée, ainsi que  $\sigma$ , l'écart-type du bruit. Or, en pratique, ces valeurs peuvent ne pas être connues précisément et même varier dans le temps. Il faut donc déterminer l'impact d'une variation des conditions du canal sur les performances de l'algorithme, étape qui fait l'objet de la prochaine section.

## 5.4 Performances suite à un changement de condition

### 5.4.1 Impact d'une variation du rapport signal sur bruit

L'algorithme de prétraitement suppose que le rapport signal sur bruit est connu. Qu'en est-il si le bruit estimé est différent du bruit réel? La figure 5.7 présente les performances pour un canal avec un OSNR de 15 dB selon que l'algorithme surestime ou sous-estime l'OSNR de 1, 3 ou 5 dB.

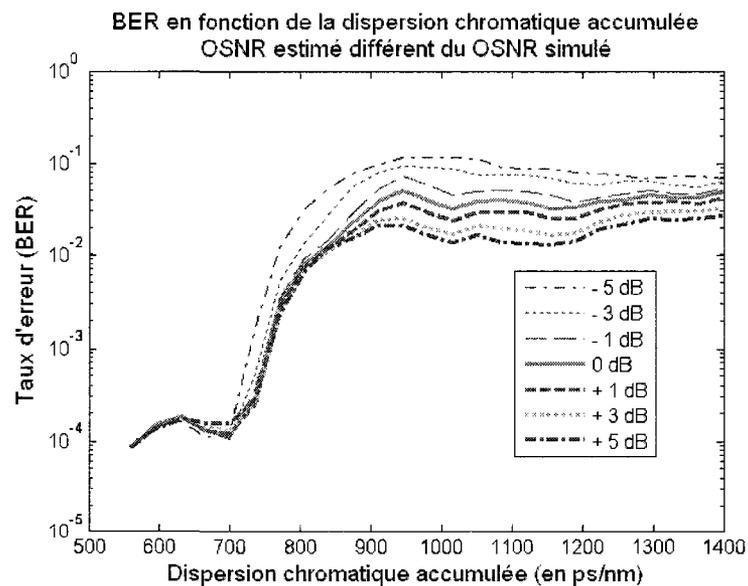


Figure 5.7 – BER lorsque l'OSNR simulé diffère de l'OSNR estimé

À la figure 5.7, les résultats sont meilleurs lorsque l'algorithme considère que le rapport signal sur bruit est de 20 dB, alors que l'OSNR réel est de 15 dB. L'amélioration se poursuit-elle si l'OSNR est davantage surestimé ? À la figure 5.8, pour un OSNR réel de 15 dB, les performances sont les meilleures lorsque l'OSNR estimé se situe entre 25 et 35 dB. En effectuant le même test pour un OSNR réel de 20 dB, l'OSNR estimé doit se situer entre 30 et 45 dB pour obtenir les meilleures performances. Pourquoi en est-il ainsi ?

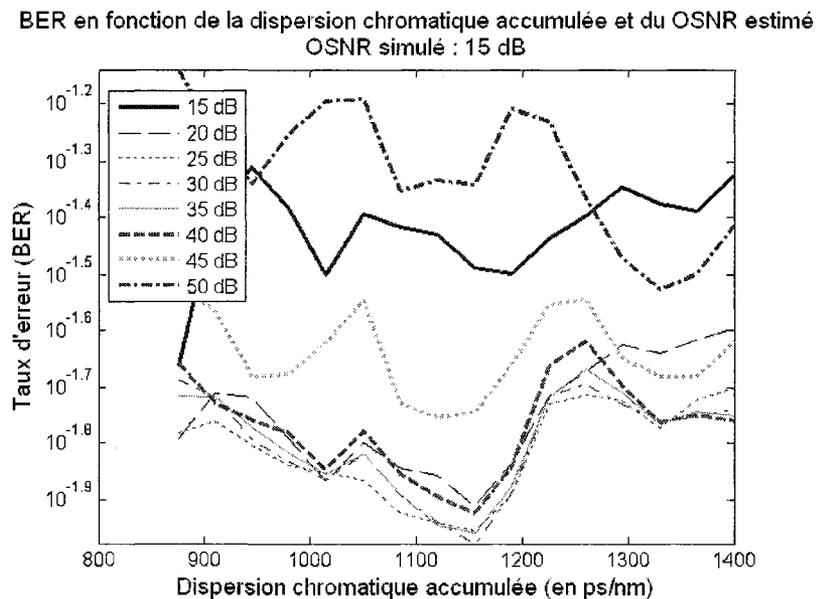


Figure 5.8 – Impact s'une surestimation de l'OSNR

Une cause possible est que le calcul de la probabilité d'appartenance à une séquence se fait à partir des extrémités des valeurs possibles. Par exemple, pour la séquence '00000' et 1400 ps/nm de dispersion chromatique accumulée, la valeur minimum sans bruit est de 0,0026, tandis que la valeur maximum est de 0,052. Si la valeur 0,054 est reçue, le calcul de la probabilité se fait en utilisant la différence entre 0,052 et 0,054, donc un bruit de 0,002. Or, la majorité des valeurs sans bruit ne se retrouvent pas aux extrémités, mais quelque part entre les deux. Par conséquent, le bruit réel est probablement supérieur à 0,002, et la probabilité d'appartenance à la séquence est surestimée.

Cette constatation ne remet toutefois pas en cause la méthode de calcul de la probabilité d'appartenance. C'est même à l'inverse un avantage : les performances sont optimales non

pas pour une seule valeur de OSNR, mais pour une plage de valeurs. En pratique, cela réduit la nécessité d'ajuster en temps réel la valeur estimée de l'OSNR. Prenons l'exemple d'un canal où l'OSNR prévu à la réception est de 15 dB. En prévoyant une marge de 3 dB, c'est-à-dire en prévoyant que la puissance du bruit pourrait doubler ou diminuer de moitié, l'estimé de l'OSNR utilisé par l'algorithme n'aurait pas à être modifié. En effet, à 15 dB, les performances sont optimales entre 25 et 35 dB. À 12 dB, il faut plutôt choisir entre 20 et 35 dB, tandis qu'à 18 dB, il faut prendre entre 30 et 40 dB. Par conséquent, en choisissant un OSNR estimé entre 30 et 35 dB, il n'est pas nécessaire d'ajuster la valeur choisie en temps réel lors de l'utilisation du système.

#### 5.4.2 Impact d'une variation de la dispersion chromatique accumulée

Il faut également vérifier comment performe l'algorithme de prétraitement si la dispersion chromatique accumulée réelle est différente de la dispersion chromatique présumée ? En d'autres mots, le taux d'erreur augmente-t-il considérablement si, par exemple, l'algorithme suppose que la dispersion chromatique est de 700 ps/nm alors qu'elle est de 735 ps/nm?

Avant de faire le test, il faut remarquer que les graphiques présentant le taux d'erreur en fonction de la dispersion chromatique accumulée peuvent être divisés en trois parties. Dans la première, entre 0 et environ 700 ps/nm, le taux d'erreur augmente en même temps que la dispersion chromatique, mais peu. Dans la deuxième partie, entre 700 et 900 ps/nm environ, le taux d'erreur augmente fortement. Finalement, dans la troisième partie, le taux d'erreur demeure à peu près constant malgré l'augmentation de la dispersion chromatique accumulée. En pratique, l'opération se ferait surtout dans la deuxième partie, car c'est dans cette partie que le décodeur devient efficace et corrige les erreurs, pour passer d'un taux d'erreur relativement élevé à un taux d'erreur nul.

La figure 5.9 présente le taux d'erreur en fonction du rapport signal sur bruit si, pour une dispersion chromatique accumulée de 805 ps/nm, l'estimé fait par l'algorithme est plutôt de 770 ps/nm ou de 840 ps/nm.

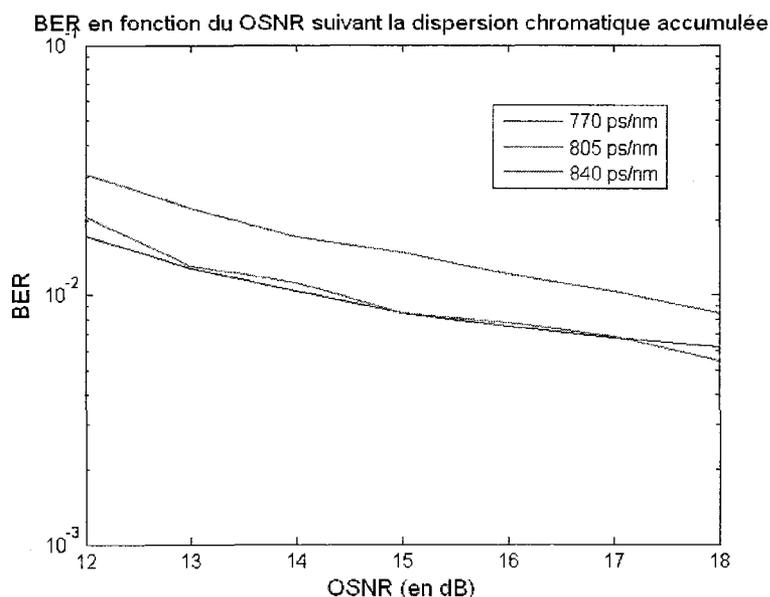


Figure 5.9 – Impact de l'estimation de la dispersion chromatique accumulée

Selon les résultats présentés à la Figure 5. 5.9, l'impact sur le taux d'erreur est faible si l'algorithme de prétraitement sous-estime la dispersion chromatique accumulée. L'impact est un peu plus grand toutefois si la dispersion chromatique est surestimée. Pour éviter ce problème, il est possible de mettre en place un système de correction de l'évaluation de la dispersion chromatique accumulée. En conditions normales d'opération, la sortie du décodeur est sans erreur, c'est-à-dire que les erreurs présentes à la sortie de l'algorithme de prétraitement sont entièrement corrigées par le décodeur. De ce fait, il est possible d'évaluer le taux d'erreur à la sortie de l'algorithme de prétraitement, et donc de déterminer si le taux d'erreur aurait été inférieur en utilisant une autre valeur pour la dispersion chromatique accumulée.

En pratique, le module de correction de l'évaluation de la dispersion chromatique accumulée serait constitué de deux modules de l'algorithme de prétraitement ainsi que d'un module de décision. Le module de décision comparerait le taux d'erreur obtenu en utilisant l'estimation actuelle de la dispersion chromatique accumulée avec ceux obtenus en utilisant une valeur légèrement supérieure ou inférieure. La valeur de dispersion chromatique retenue pour le mot de code suivant serait celle produisant le plus faible taux d'erreur.

## 5.5 Aperçu d'ensemble

Outre les composantes de base propres à tout récepteur (photodétecteur, intégrateur, système de récupération des impulsions d'horloge et de synchronisation), le système comporterait un module pour l'algorithme de prétraitement, un module pour le décodeur, une mémoire pour stocker les statistiques des valeurs à la sortie du canal et un module pour la mise à jour de l'estimation de la dispersion chromatique accumulée. La figure 5.10 présente un aperçu du système.

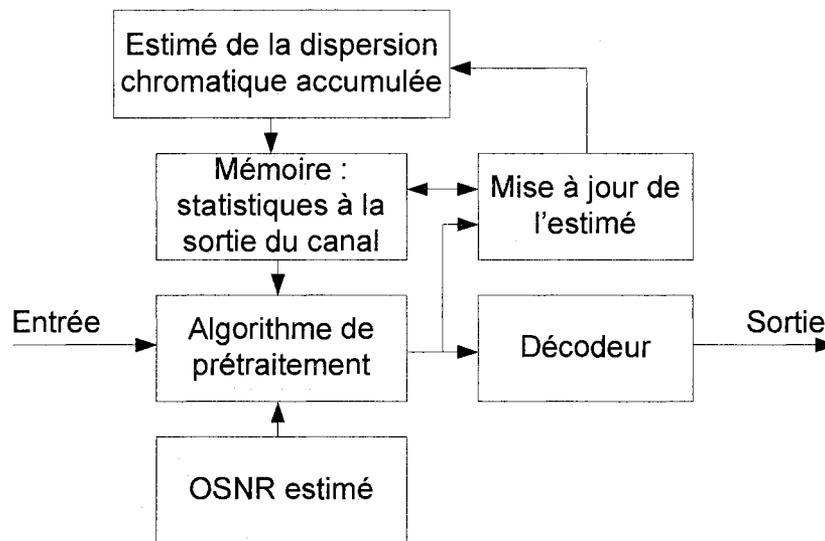


Figure 5.10 – Schéma du système

Le fait que l'algorithme de contrôle d'erreur proposé se base sur un système déjà testé règle une considération pratique importante. Il est en effet raisonnable de penser qu'il est possible de bâtir physiquement le système compte tenu des technologies disponibles actuellement : l'équipe de Mitsubishi a déjà fabriqué un décodeur BCH(144,128) x BCH(256,239) avec succès [OUCHI et coll., 2006]. L'ajout proposé, soit l'algorithme de prétraitement, n'ajoute pas beaucoup de complexité au système. Il nécessite quelques calculs, multiplications et additions, et un espace mémoire pour stocker les informations relatives aux statistiques à la sortie du canal. Comparativement, le décodage demande des opérations beaucoup plus complexes.

Par conséquent, le système de correction d'erreurs proposé semble une option réaliste dans le contexte d'un réseau métropolitain opérant à 10 Gb/s.

## 5.6 Résumé

La dispersion chromatique a un grand impact sur la distribution des valeurs à la sortie du canal. Il est possible de tirer profit de cette connaissance grâce à l'algorithme de prétraitement proposé. Cet algorithme calcule la probabilité d'appartenance de chaque bit à chacune des séquences possibles, afin de déterminer laquelle est la plus probable et par conséquent, quelle valeur a été émise. Les résultats sont très concluants lorsque la dispersion chromatique accumulée est élevée, le taux d'erreur étant fortement réduit.

L'utilisation de l'algorithme de prétraitement n'est toutefois pas suffisante pour permettre d'atteindre un taux d'erreur pratiquement nul. Pour ce faire, il faut le combiner avec le décodeur présenté au chapitre 4. Utilisés conjointement, les deux modules permettent d'augmenter considérablement la distance limite de transmission à faible taux d'erreur.

Pour une utilisation dans des conditions réelles, il est nécessaire d'implanter un système temps réel d'estimation de la dispersion chromatique accumulée. Toutefois, il n'est pas nécessaire d'évaluer le rapport signal sur bruit de façon continue, puisqu'un OSNR donné utilisé par l'algorithme de prétraitement permet d'atteindre les performances optimales pour une plage assez large d'OSNR.

Finalement, vu la simplicité de l'algorithme de prétraitement proposé, il est raisonnable de penser qu'un système contenant à la fois le décodeur et l'algorithme de prétraitement pourrait être bâti physiquement en utilisant les technologies actuelles.

## CONCLUSION

### Synthèse et prise de position

L'objectif du projet de recherche est la conception d'un compensateur de dispersion électronique performant et à faible coût pour un canal à 10 Gb/s dans le contexte d'un réseau métropolitain. La dispersion chromatique cause de l'interférence inter-symbole et constitue une limite importante à la distance maximale de transmission à partir de 10 Gb/s. Plusieurs solutions existent déjà, les principales catégories étant la pré-compensation, la compensation optique, l'égalisation et la correction d'erreurs. Dans le cadre du projet de recherche, la solution choisie est le turbo code BCH(144,128) x BCH(256,239), car les premiers résultats obtenus avec ce code correcteur d'erreurs sont impressionnants et semblent pouvoir être améliorés.

Pour tester les performances du compensateur proposé, il faut d'abord simuler le canal, simuler l'impact de la dispersion chromatique sur un signal. La méthode retenue consiste à obtenir la réponse en fréquence du canal à partir de l'équation de Sellmeier. La simulation du canal reproduit également l'impact du récepteur, soit le transfert de la puissance optique en courant électrique et la bande passante limitée de l'électronique. Le signal est modulé suivant le format NRZ-OOK et la numérisation est faite à huit échantillons par bit pour respecter le théorème d'échantillonnage.

L'hypothèse selon laquelle la limite de propagation est atteinte lorsque la moitié de l'énergie se retrouve dans les bits adjacents est infirmée lors du calcul du taux d'erreur. Pour l'obtention des performances du système, la méthode Monte Carlo est choisie à la fois pour sa simplicité et parce qu'elle s'applique avec ou sans décodeur. Les résultats permettent également de conclure que le modèle utilisé pour les simulations est plus dispersif qu'une fibre optique réelle. Par conséquent, il n'est pas possible de mesurer de manière absolue l'amélioration apportée par l'algorithme proposé par rapport à un canal sans décodeur.

Le décodeur à décision pondérée de base repose sur l'algorithme de Chase et sur le calcul de la fiabilité de Pyndiah. Les simulations montrent l'existence d'un seuil dans les performances : le décodeur permet de corriger la plupart des erreurs uniquement si le taux d'erreur initial est en deçà d'une certaine valeur.

Le décodeur de base ne tire par contre pas profit du caractère déterministe de la dispersion chromatique. Les simulations montrent qu'à cause de la dispersion chromatique, la distribution des valeurs à la sortie du canal est fortement influencée par la séquence des bits émis. L'algorithme proposé se limite à étudier pour chaque bit l'impact des deux bits précédents et des deux bits suivants. Il calcule la probabilité d'appartenance de chaque bit à chacune des séquences possibles, puis détermine quelle séquence est la plus probable. Employé seul, l'algorithme permet de réduire le taux d'erreur lorsque la dispersion chromatique accumulée est élevée.

Ici, il est bon de rappeler l'hypothèse de travail : tenir compte de l'effet de la dispersion chromatique sur la distribution statistique du bruit augmente l'efficacité du FEC en présence de dispersion chromatique. Les résultats obtenus au chapitre 5 confirment cette hypothèse. En effet, l'utilisation de l'algorithme de prétraitement améliore grandement les performances du code correcteur d'erreur étudié.

Finalement, puisque l'algorithme de prétraitement proposé est simple, un système contenant à la fois le décodeur et l'algorithme de prétraitement pourrait très probablement être réalisé physiquement en utilisant les technologies actuelles.

### **Nouvelles perspectives de recherche**

Pour améliorer ce travail de recherche, il faudrait d'abord raffiner le modèle utilisé pour les simulations afin qu'il se rapproche plus des résultats obtenus en pratique avec la fibre optique. Le piaillement pourrait également être ajouté au modèle, afin de vérifier si l'algorithme peut être utilisé avec un autre modulateur que le modulateur Mach-Zender. Autre modification possible : raffiner le modèle utilisé pour le bruit afin qu'il se rapproche le plus possible du bruit rencontré en pratique.

Outre l'amélioration du modèle de simulation, il faudrait également s'attarder à un aspect pratique important : la quantification au décodeur. En effet, toutes les valeurs simulées par le logiciel MATLAB ont une précision de 64 bits. Sur une puce, effectuer les calculs sur 64 bits

demanderait trop d'espace. Quel serait l'impact d'une réduction du nombre de bits ? Combien de bits sont nécessaires pour obtenir de bons résultats ?

Il faut également se questionner par rapport au temps de simulation. Serait-il possible d'atteindre des taux d'erreur de l'ordre de  $10^{-12}$  en un temps raisonnable ? Deux avenues de recherche sont possibles : réduire considérablement les temps de simulation ou utiliser une méthode autre que Monte Carlo. L'échantillonnage stratégique (*Importance Sampling*) et la méthode Monte Carlo multicanonique sont des candidats intéressants, s'il s'avère possible de les adapter pour de très longs mots de code.

Finalement, l'étape ultime serait de valider les résultats expérimentalement, c'est-à-dire concevoir en détails le circuit imprimé requis et le faire fabriquer, puis le tester en utilisant laser, fibre optique et amplificateurs. Un tel test requiert cependant des fonds de recherche importants.

## ANNEXE I

### CALCUL DE LA CONSTANTE DE TEMPS DU SYSTÈME D'ORDRE 1

Pour un système passe-bas d'ordre un, la réponse à l'échelon vaut :

$$y = 1 - e^{-t/\tau} \quad (\text{A1.1})$$

À trouver : la constante de temps  $\tau$  nécessaire pour que la sortie  $y$  passe de 10% à 90% de sa valeur finale en 25 ps.

Soit  $t_1$  l'instant où  $y$  atteint 10% de sa valeur finale, soit 0,1. Alors on obtient :

$$\begin{aligned} 0,1 &= 1 - e^{-t_1/\tau} \\ e^{-t_1/\tau} &= 0,9 \\ t_1 &= -\tau \ln(0,9) \end{aligned} \quad (\text{A1.2})$$

De la même façon, en posant  $t_2$  l'instant où  $y$  atteint 0,9, soit 90% de sa valeur finale, on obtient :

$$\begin{aligned} t_2 &= -\tau \ln(0,1) \\ & \quad (\text{A1.3}) \end{aligned}$$

On sait également que  $y$  doit passer de 0,1 à 0,9 en 25 ps, d'où :

$$\begin{aligned} t_2 - t_1 &= 25 \text{ ps} \\ & \quad (\text{A1.4}) \end{aligned}$$

En combinant les équations, on obtient :

$$-\tau \ln(0,1) + \tau \ln(0,9) = 25 \text{ ps} \quad (\text{A1.5})$$

$$\text{D'où : } \tau = 11,378 \text{ ps}$$

Puisque la fréquence de coupure  $f_c = \frac{1}{2\pi\tau}$ , on obtient  $f_c = 14 \text{ GHz}$ .

## ANNEXE II

### CALCUL DE LA TRANSFORMÉE DE FOURIER DU SIGNAL

Le but est de calculer la transformée de Fourier du signal suivant :

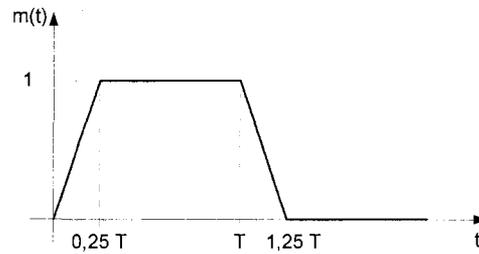


Figure All.1 – Signal pour le calcul de la transformée de Fourier

Il serait possible d'utiliser directement la formule de la transformée de Fourier :

$$M(\omega) = \int_{-\infty}^{\infty} m(t) e^{-j\omega t} dt \quad (\text{All.1})$$

Mais quelques propriétés de la transformée de Fourier et une table de résultats peuvent simplifier les calculs. En effet, la transformée de Fourier est une fonction linéaire, c'est-à-dire que :

$$ax(t) + by(t) \leftrightarrow aX(\omega) + bY(\omega) \quad (\text{All.2})$$

Et dans les tables, on retrouve la transformée du signal suivant :

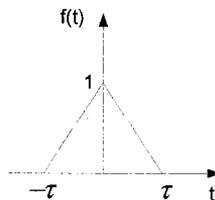


Figure All.2 – Signal triangulaire

Mathématiquement :

$$\text{si } f(t) = \begin{cases} 1 - \frac{|t|}{\tau} & |t| \leq \tau \\ 0 & |t| > \tau \end{cases}, \text{ alors } F(\omega) = \tau \left[ \text{sinc}\left(\frac{\omega\tau}{2}\right) \right]^2, \text{ avec } \text{sinc}(x) = \frac{\sin(x)}{x}$$

(AII.3)

En prenant  $\tau = 0,25T$  et en additionnant 4 exemplaires de cette fonction décalés dans le temps, il est possible de reconstituer le signal initial.

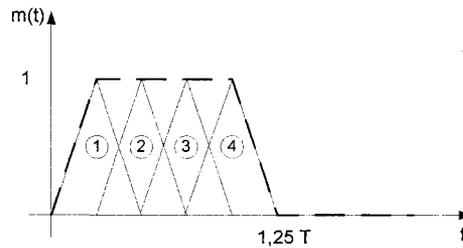


Figure AII.3 – Reconstitution du signal à partir du signal triangulaire

La propriété du décalage temporel permet de terminer le calcul :

$$x(t - t_0) \leftrightarrow e^{-j\omega t_0} X(\omega) \quad (\text{AII.4})$$

Ainsi, on a :

$$M(\omega) = 0,25T \left[ e^{-j\omega 0,25T} \left( \text{sinc}\left(\frac{\omega T}{8}\right) \right)^2 + e^{-j\omega 0,5T} \left( \text{sinc}\left(\frac{\omega T}{8}\right) \right)^2 + e^{-j\omega 0,75T} \left( \text{sinc}\left(\frac{\omega T}{8}\right) \right)^2 + e^{-j\omega T} \left( \text{sinc}\left(\frac{\omega T}{8}\right) \right)^2 \right]$$

(AII.5)

## ANNEXE III

### CALCUL DU POURCENTAGE D'ÉNERGIE DANS LES BITS ADJACENTS

L'objectif est de déterminer quel pourcentage de l'énergie émise par un bit se retrouve dans un bit adjacent. Il faut donc en premier lieu établir l'énergie totale produite par le bit.

Par définition, l'énergie totale  $E_{tot}$  vaut :

$$E_{tot} = \int_0^{\infty} P(t) dt \quad (\text{AIII.1})$$

$P(t)$  s'obtient à partir de  $P_f(n)$ , la puissance instantanée des échantillons, en faisant une interpolation linéaire. Ainsi,  $P(t)$  sera définie par intervalle de durée  $T$ . Soit  $E_n$  l'énergie contenue dans l'une de ces intervalles, alors on a :

$$E_{tot} = \sum_{n=0}^{N-1} E_n \quad (\text{AIII.2})$$

et 
$$E_n = \int_0^T P_n(t) dt \quad (\text{AIII.3})$$

où  $P_n(t)$  est la fonction résultant de l'interpolation linéaire entre les échantillons  $P_f(n)$  et  $P_f(n+1)$  :

$$P_n(t) = mt + c \quad (\text{AIII.4})$$

avec 
$$m = \frac{P_f(n+1) - P_f(n)}{T}$$

$$c = P_f(n)$$

Pour un intervalle :

$$E_n = \int_0^T P_n(t) dt = \int_0^T \left[ \frac{P_f(n+1) - P_f(n)}{T} t + P_f(n) \right] dt \quad (\text{AIII.5})$$

$$E_n = \left| \frac{P_f(n+1) - P_f(n)}{2T} t^2 \right|_0^T + \left| P_f(n) t \right|_0^T = \left( \frac{P_f(n) + P_f(n+1)}{2} \right) T$$

$$E_{tot} = \sum_{n=0}^{N-1} \left( \frac{P_f(n) + P_f(n+1)}{2} \right) T \quad (\text{AIII.6})$$

$$E_{bit} = \sum_{n=n_1}^{n_2} \left( \frac{P_f(n) + P_f(n+1)}{2} \right) T \quad (\text{AIII.7})$$

$$E_{adj} = 1 - \frac{E_{bit}}{E_{tot}} \quad (\text{AIII.8})$$

## ANNEXE IV

### OPTIMISATION DE LA DIVISION EN SÉQUENCES

Une façon de réduire le temps de simulation est de diviser la séquence à simuler, de 36864 bits, en plus petites séquences. En effet, il est plus rapide de simuler deux fois 128 bits qu'une fois 256 bits. Par contre, il faut s'assurer que les résultats de simulation demeurent valables malgré cette modification.

D'abord, quels sont les choix possibles ? Idéalement, la division choisie devrait être un facteur de 36 864, pour éviter de simuler des bits en trop. Le tableau AIV.1 en présente la liste.

Tableau AIV.1 – Liste des facteurs de 36 864

1	3	9
2	6	18
4	12	36
8	24	72
16	48	144
32	96	288
64	192	576
128	384	1152
256	768	2304
512	1536	4608
1024	3072	9216
2048	6144	18432
4096	12288	36864

Un critère important est le temps de simulation. Pour des raisons pratiques, il faudrait qu'une simulation pour 36864 bits puisse s'effectuer en une heure environ, ou plutôt 3600 secondes. En supposant que les temps de simulation demeurent constants, simuler par séquences de 256 bits ne serait alors pas envisageable, car il faudrait simuler 144 séquences de 256 bits et à 242 secondes par séquence, la simulation durerait environ 34848 secondes. Simuler par séquences de 128 bits ne serait pas acceptable non plus, car une telle simulation prendrait environ 2 heures et demi.

Toutefois, le temps de simulation n'est pas le seul critère applicable. Il faut également s'assurer que les résultats obtenus sont valables malgré la division de la séquence. Pour mieux comprendre, il est intéressant de reprendre l'exemple des simulations de 256 bits et 128 bits. Soit une même suite de 256 bits simulée deux fois. La première fois, les 256 bits sont simulés ensemble. La seconde fois, les bits sont séparés en 2 séquences de 128 bits. Comparer les résultats des deux simulations permet de voir l'impact de la division de la séquence. La figure AIV.1 présente le résultat d'une telle expérience. Deux observations peuvent être faites. Premièrement, la différence entre les résultats est généralement faible. L'erreur médiane est de  $2,84 \times 10^{-4}$ . Deuxièmement, les erreurs importantes sont concentrées près des extrémités des séquences, ici les bits 0, 128 et 256. Il faut garder à l'esprit que la même relation existe certainement entre les résultats de la simulation de 256 bits et, par exemple, une simulation de 512 bits ou tout autre multiple de 256.

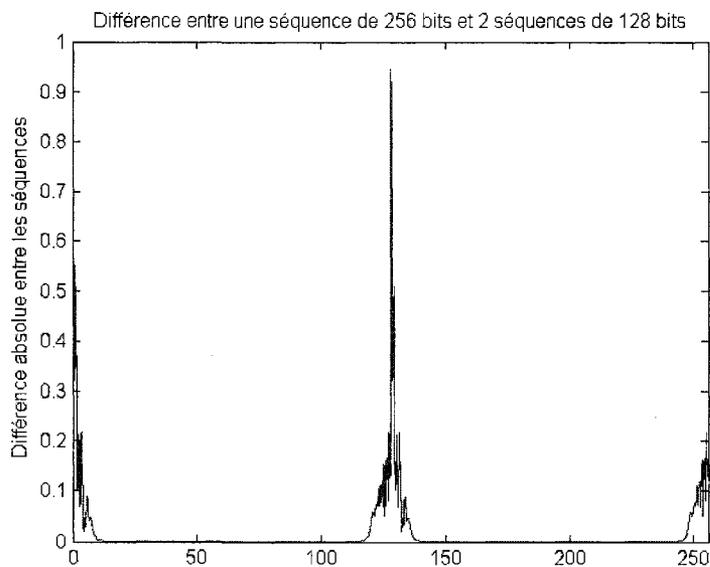


Figure AIV.1 – Différence entre la simulation de 256 bits et deux séquences de 128 bits

De ces observations découle la conclusion suivante : la division de la séquence permet d'obtenir des résultats valables, en autant qu'une mesure est prise pour limiter l'erreur aux extrémités. Une mesure possible est de simuler plus de bits qu'il n'est nécessaire. Par exemple, ajouter les derniers bits de la séquence précédente et les premiers bits de la séquence suivante à la séquence en cours de simulation, mais ne pas conserver le résultat

des bits supplémentaires ajoutés. La figure AIV.1 présente les mêmes résultats que la figure AIV.2, à la différence que les deux premiers bits de la deuxième séquence ont été ajoutés lors de la simulation de la première séquence et que les deux derniers bits de la première séquence ont été ajoutés lors de la simulation de la deuxième séquence. On remarque que cette mesure a permis de réduire considérablement l'erreur autour du 128<sup>e</sup> bit.

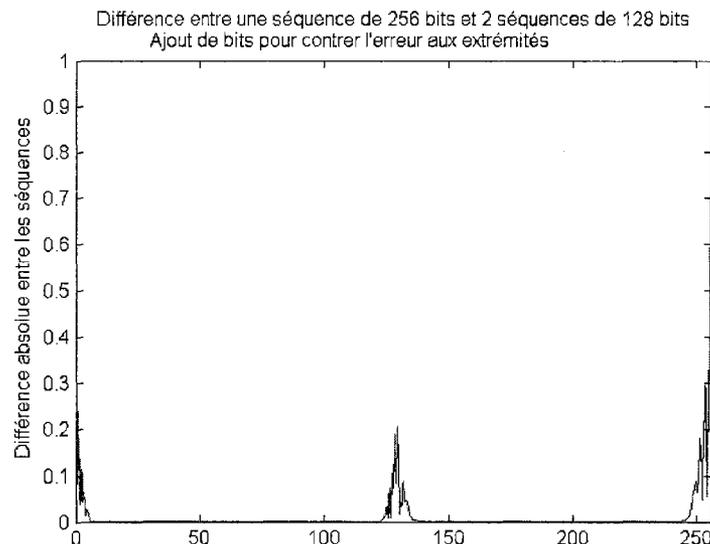


Figure AIV.2 – Effet de l'ajout de bits sur l'erreur aux extrémités

Puisque les simulations seront effectuées maintes et maintes fois, il est intéressant à ce stade-ci d'optimiser la combinaison nombre de bits par séquence / nombre de bits supplémentaires afin d'obtenir le meilleur compromis entre la précision des résultats et le temps de simulation.

Le test effectué est le suivant : chaque combinaison nombre de bits par séquence / nombre de bits supplémentaires à l'étude est simulée, pour atteindre une séquence totale de 288 bits. Par exemple, pour la combinaison 36,12, 8 séquences de 36 bits sont simulés et pour chaque séquence, les 12 bits précédents et suivants sont considérés lors des simulations. Le nombre 288 est choisi car parmi les facteurs de 36864 simulables en un temps raisonnable, c'est lui qui possède le plus grand nombre de facteurs.

Une fois les séquences simulées, les résultats sont comparés à une séquence test de 288 bits, afin de déterminer l'erreur moyenne et l'erreur maximale. Pour éviter l'effet d'erreur aux

extrémités, seulement les 200 bits centraux sont comparés. Il faut également simuler pour plusieurs séquences et pour plusieurs distances afin de s'assurer de la validité des résultats. Nous nous limiterons toutefois à trois simulations : deux séquences différentes à 80 km et une autre à 40 km.

Voici la procédure suivie afin de trouver la meilleure combinaison possible. D'abord, les résultats sont triés pour obtenir les 30 combinaisons ayant la plus faible erreur moyenne ainsi que les 30 combinaisons ayant la plus faible erreur maximale. Seules les combinaisons figurant dans les deux listes sont retenues. Puis, pour ces combinaisons, on compile l'erreur moyenne et l'erreur maximale pour les deux autres simulations. La figure AIV.3 présente les valeurs maximales des résultats obtenus pour les 3 essais, pour les combinaisons retenues. Les combinaisons 36,13 et 16,16 se démarquent des autres. Toutefois, en regardant l'échelle des axes, on se rend compte que pour toutes ces combinaisons, l'erreur moyenne et l'erreur maximale sont plutôt faibles. En effet, pour l'erreur maximale, le maximum est de 0,0058 et le minimum de 0,0031. Pour l'erreur moyenne, les valeurs varient entre  $1,8 \times 10^{-4}$  à  $9,4 \times 10^{-4}$ . On peut donc juger que toutes ces combinaisons produisent des résultats acceptables et se pencher sur le temps de simulation.

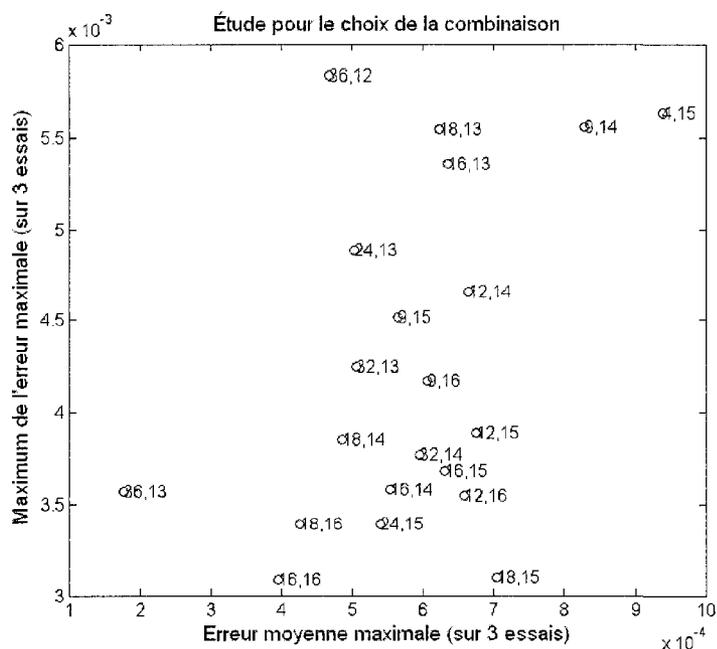


Figure AIV.3 – Maximum sur 3 essais pour l'optimisation de la combinaison

Or, il y a une assez grande différence pour les temps de simulation. La simulation la plus rapide s'effectue en 19,2 s tandis que la plus lente s'effectue en 33,4s. La différence peut sembler faible, mais pour 36684 bits, cela représente 41 minutes comparativement à 71 minutes.

Les combinaisons suivantes ont un temps de simulation proche de 20 secondes : 16,13 ; 16,14 ; 9,14 ; 12,14 ; 18,13. La figure AIV.4 reprend les mêmes données que la figure AIV.3, mais cette fois-ci les simulations les plus rapides sont mises en évidence. La combinaison 16,14 semble donc représenter le meilleur compromis entre de bons résultats et un temps de simulation court.

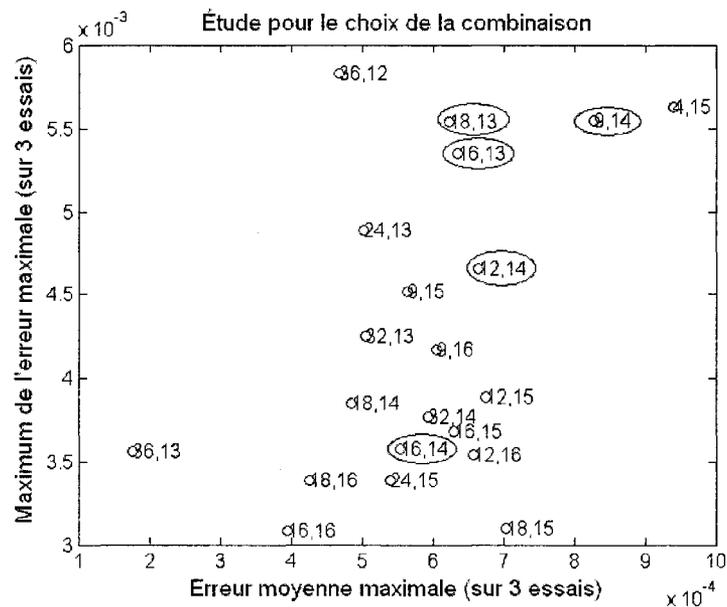


Figure AIV.4 – Mise en évidence des plus courts temps de simulation

## ANNEXE V

### CALCUL DE LA FIABILITÉ DE LA DÉCISION

Soit  $\hat{x}_j$  la décision prise au bit  $j$ . Alors la fiabilité de la décision au bit  $j$  se calcule comme suit :

$$\Lambda(\hat{x}_j) = \ln \left( \frac{P[x_j = 1 | \mathbf{R}]}{P[x_j = 0 | \mathbf{R}]} \right) \quad (\text{AV.1})$$

où  $P[x_j = 1 | \mathbf{R}]$  est la probabilité que  $x_j$ , le  $j^{\text{e}}$  bit émis, soit un '1', sachant que  $\mathbf{R}$  a été reçu.

Or,  $P[x_j = 1 | \mathbf{R}]$  n'est pas facilement calculable. En appliquant la loi de Bayes, il est toutefois possible de simplifier l'équation.

$$\text{La loi de Bayes s'énonce comme suit : } P[a = b | c] = \frac{P[c | a = b] \times P[a = b]}{P[c]} \quad (\text{AV.2})$$

En l'appliquant, on obtient :

$$P[x_j = 1 | \mathbf{R}] = \frac{P[\mathbf{R} | x_j = 1] \times P[x_j = 1]}{P[\mathbf{R}]} \quad (\text{AV.3})$$

$$P[x_j = 0 | \mathbf{R}] = \frac{P[\mathbf{R} | x_j = 0] \times P[x_j = 0]}{P[\mathbf{R}]} \quad (\text{AV.4})$$

En supposant que les différents mots de code sont distribués uniformément et donc que  $P[x_j = 1] = P[x_j = 0]$ , alors on obtient :

$$\Lambda(\hat{x}_j) = \ln \left( \frac{P[\mathbf{R} | x_j = 1]}{P[\mathbf{R} | x_j = 0]} \right) \quad (\text{AV.5})$$

$$\text{Or } P[\mathbf{R} | x_j = 1] = \sum_{\mathbf{C}^i \in \mathcal{S}_j^1} P[\mathbf{R} | \mathbf{X} = \mathbf{C}^i] \quad (\text{AV.6})$$

où  $S_j^1$  est l'ensemble des mots de code  $\{C^i\}$  tel que le  $j^{\text{e}}$  bit de  $C^i$  vaut '1'.

$$\text{De même, on peut écrire : } P[\mathbf{R}|x_j = 0] = \sum_{C^i \in S_j^0} P[\mathbf{R}|X = C^i] \quad (\text{AV.7})$$

$$\text{D'où : } \Lambda(\hat{x}_j) = \ln \left( \frac{\sum_{C^i \in S_j^1} P[\mathbf{R}|X = C^i]}{\sum_{C^i \in S_j^0} P[\mathbf{R}|X = C^i]} \right) \quad (\text{AV.8})$$

Pour un canal avec bruit blanc gaussien, on a :

$$P[\mathbf{R}|X = C^i] = \left( \frac{1}{\sqrt{2\pi}\sigma} \right)^n \exp \left( -\frac{|\mathbf{R} - C^i|^2}{2\sigma^2} \right) \quad (\text{AV.9})$$

$$\text{En substituant : } \Lambda(\hat{x}_j) = \ln \left( \frac{\sum_{C^i \in S_j^1} \exp \left( -\frac{|\mathbf{R} - C^i|^2}{2\sigma^2} \right)}{\sum_{C^i \in S_j^0} \exp \left( -\frac{|\mathbf{R} - C^i|^2}{2\sigma^2} \right)} \right) \quad (\text{AV.10})$$

Soit  $C^{1(j)}$  et  $C^{0(j)}$  le mot de code à la distance Euclidienne minimale dans  $S_j^1$  et  $S_j^0$  respectivement. Alors :

$$\Lambda(\hat{x}_j) = \ln \left( \frac{\exp \left( -\frac{|\mathbf{R} - C^{1(j)}|^2}{2\sigma^2} \right) \sum_{C^i \in S_j^1} \exp \left( \frac{|\mathbf{R} - C^{1(j)}|^2 - |\mathbf{R} - C^i|^2}{2\sigma^2} \right)}{\exp \left( -\frac{|\mathbf{R} - C^{0(j)}|^2}{2\sigma^2} \right) \sum_{C^i \in S_j^0} \exp \left( \frac{|\mathbf{R} - C^{0(j)}|^2 - |\mathbf{R} - C^i|^2}{2\sigma^2} \right)} \right) \quad (\text{AV.11})$$

Puisque  $\ln(ab) = \ln(a) + \ln(b)$  et que  $\ln\left(\frac{a}{b}\right) = \ln(a) - \ln(b)$ , on a donc :

$$\Lambda(\hat{x}_j) = \frac{1}{2\sigma^2} \left( |\mathbf{R} - \mathbf{C}^{0(j)}|^2 - |\mathbf{R} - \mathbf{C}^{1(j)}|^2 \right) + \ln \left( \frac{\sum_{\mathbf{C}^i \in \mathcal{S}_j^1} \exp \left( \frac{|\mathbf{R} - \mathbf{C}^{1(j)}|^2 - |\mathbf{R} - \mathbf{C}^i|^2}{2\sigma^2} \right)}{\sum_{\mathbf{C}^i \in \mathcal{S}_j^0} \exp \left( \frac{|\mathbf{R} - \mathbf{C}^{0(j)}|^2 - |\mathbf{R} - \mathbf{C}^i|^2}{2\sigma^2} \right)} \right) \quad (\text{AV.12})$$

L'approximation qui est faite est que pour un haut rapport signal sur bruit, le second terme tend vers 0. D'où l'approximation suivante :

$$\Lambda'(\hat{x}_j) = \frac{1}{2\sigma^2} \left( |\mathbf{R} - \mathbf{C}^{0(j)}|^2 - |\mathbf{R} - \mathbf{C}^{1(j)}|^2 \right) \quad (\text{AV.13})$$

Il faut maintenant se questionner sur le lien entre la fiabilité d'un bit et la valeur reçue à la sortie du canal. Soit  $y_j$  la décision à la sortie du canal obtenue à partir de  $r_j$  et du seuil de décision.

Alors la fiabilité de  $y_j$  se calcule comme suit :

$$\Lambda(y_j) = \ln \left( \frac{P[x_j = 1 | r_j]}{P[x_j = 0 | r_j]} \right) = \ln \left( \frac{\exp \left( -\frac{(r_j - 1)^2}{2\sigma^2} \right)}{\exp \left( -\frac{(r_j - 0)^2}{2\sigma^2} \right)} \right) = \frac{2r_j - 1}{2\sigma^2} \quad (\text{AV.14})$$

$$\text{D'où : } r_j = \sigma^2 \Lambda(y_j) + 0,5 \quad (\text{AV.15})$$

Si l'on suppose qu'il existe la même relation entre la fiabilité de la décision et la valeur du bit à utiliser pour l'étage suivante du décodeur, on obtient :

$$r'_j = \sigma^2 \Lambda'(\hat{x}_j) + 0,5 = 0,5 \left( |\mathbf{R} - \mathbf{C}^{0(j)}|^2 - |\mathbf{R} - \mathbf{C}^{1(j)}|^2 \right) + 0,5 \quad (\text{AV.16})$$

Pour optimiser les performances du décodeur, il faut toutefois apporter une modification à cette équation. Cette dernière fonctionne bien si le seuil de décision est fixé à 0,5. Or, il a été vu au chapitre 3 que le seuil de décision optimal évolue en fonction de la dispersion chromatique accumulée et ne vaut généralement pas 0,5. Il est donc préférable d'utiliser l'équation suivante :

$$r'_j = 0,5 \left( |\mathbf{R} - \mathbf{C}^{0(j)}|^2 - |\mathbf{R} - \mathbf{C}^{1(j)}|^2 \right) + \text{seuil} \quad (\text{AV.17})$$

## BIBLIOGRAPHIE

AC6539 : Produit de Archcom Technology

Fiche technique : [http://www.archcomtech.com/datasheets\\_1/AC6539,%20rev%20A.pdf](http://www.archcomtech.com/datasheets_1/AC6539,%20rev%20A.pdf)

AGAZZI, O. E., HUEDA, M. R., CARRER, H. S., CRIVELLI, D. E. (février 2005) *Maximum-Likelihood Sequence Estimation in Dispersive Optical Channels*, Journal of Lightwave Technology, vol. 23, n° 2, p. 749-763.

AGRAWAL, G. P. (1997) *Fiber-Optic Communication Systems*, New York, 2<sup>e</sup> édition, John Wiley & Sons, 555 p.

BARLOW, G. (octobre 2003) *A G.709 Optical Transport Network Tutorial*, disponible à : [http://www.innocor.com/pdf\\_files/g709\\_tutorial.pdf](http://www.innocor.com/pdf_files/g709_tutorial.pdf)

BERLEKAMP, E. (octobre 1965) *On Decoding Binary Bose-Chaudhuri-Hocquenghem Codes*, IEEE Transactions on Information Theory, vol. 11, n° 4, p.577-580.

BOUTIN, N. (2005) *Principes de base des systèmes de communications analogiques et numériques*, Université de Sherbrooke, Département de génie électrique et informatique.

BÜLOW, H. (mars 2000) *PMD mitigation techniques and their effectiveness in installed fiber*, Optical Fiber Communication Conference OFC 2000, Baltimore, ThH1-1, 3 p.

CAI, Y., MORRIS, J. M., ADALI, T., MENYUK, C. R. (mars 2003) *On Turbo Code Decoder Performance in Optical-Fiber Communication Systems with Dominating ASE Noise*, Journal of Lightwave Technology, vol. 21, n° 3, p. 727-734.

CHASE, D. (janvier 1972) *A Class of Algorithms for Decoding Block Codes With Channel Measurement Information*, IEEE Transactions on information theory, vol. IT-18, n° 1, p. 170-182.

CHIEN, R. (octobre 1964) *Cyclic Decoding Procedures for the Bose-Chaudhuri-Hocquenghem Codes*, IEEE Transactions on Information Theory, vol. 10, n° 4, p.357-363.

CORINTHIOS, M. J. (2001) *Analyse des signaux*, Montréal, 5<sup>e</sup> édition, École Polytechnique de Montréal, 545 p.

CURRI, V., GAUDINO, R., NAPOLI, A., POGGIOLINI, P. (novembre 2004) *Electronic Equalization for Advanced Modulation Formats in Dispersion-Limited Systems*, IEEE Photonics Technology Letters, vol. 16, n° 11, p. 2556-2558.

ERM568RLC : Produit de JDS Uniphase

Fiche technique: [http://products.jdsu.com/assets/public/pdf/erm568rlc\\_ds\\_cc\\_ae\\_030806.pdf](http://products.jdsu.com/assets/public/pdf/erm568rlc_ds_cc_ae_030806.pdf)

ESSIAMBRE, R.-J., WINZER, P. J. (mars 2006) *Impact of Fiber Nonlinearities on Advanced Modulation Formats Using Electronic Pre-distortion*, Optical Fiber Communication Conference OFC 2006, Anaheim, OWB1, 3 p.

GHIASI, A., MOMTAZ, A., DASTUR, A., CHANG, F., NOH, G., GOMATAM, B., IBRAGIMOV, E., SHANBAG, B., SCHREIBER, O., SU, E., CONROY, K., JAMBUNATHAN, R., WOOD, J. (mars 2006) *Experimental Results of EDC Based Receivers for 2400 ps/nm at 10.7 Gb/s for Emerging Telecom Standards*, Optical Fiber Communication Conference OFC 2006, Anaheim, OTuE3, 3 p.

GNAUCK, A. H., LIU, X., WIE, X. GILL, D. M., BURROWS, E. C. (mars 2004) *Comparison of Modulation Formats for 42.7 Gb/s Single-Channel Transmission Through 1980 km of SSMF*, IEEE Photonics Technology Letters, vol.16, n° 3, p. 909-911.

GRÜNER-NIELSEN, L., WANDEL, M., KRISTENSEN, P., JØRGENSEN, C., JØRGENSEN, L. V., EDVOLD, B. PÁLSDÓTTIR, B., JAKOBSEN, D. (novembre 2005) *Dispersion-Compensating Fibers*, Journal of Lightwave Technology, vol. 23, n° 11, p. 3566-3579.

HENMI, N., SAITO, T., ISHIDA, T. (octobre 1994) *Prechirp Technique as a Linear Dispersion Compensation for Ultrahigh-Speed Long-Span Intensity Modulation Directed Detection Optical Communication Systems*, Journal of Lightwave Technology, vol. 12, n° 10, p. 1706-1719.

HOLZLÖHNER, R., MAHADEVAN, A., MENYUK, C. R., MORRIS, J. M., ZWECK, J. (février 2005) *Evaluation of the Very Low BER of FEC Codes Using Dual Adaptive Importance Sampling*, IEEE Communications Letters, vol. 9, n° 2, p.163-165.

HOULE, A. (2006) *GEI705 : Réseaux de communication par fibre optique*, notes de cours personnelles, Sherbrooke.

ISSA, V. (2007) *L'effet de la dispersion chromatique sur le taux d'erreur de transmission par fibre optique pour un débit de 10 Gb/s*, mémoire de maîtrise, Université de Sherbrooke, 90 p.

KAMINOW, I. P., KOCH, T. L. (1997) *Optical Fiber Telecommunications IIIA*, Academic Press, 608 p.

LIN, S., COSTELLO, D. J. (1983) *Error Control Coding : Fundamentals and Applications*, Englewood Cliffs, Prentice-Hall, 603 p.

MATSUMOTO, Y., KURIYAMA, T., INAMI, D., OHTA, M., SHIRAIWA, M. (mars 2001) *An adaptive decision threshold control of the optical receiver for multi-gigabit terrestrial DWDM transmission systems*, Optical Fiber Communication Conference OFC 2001, Anaheim, TuR2, 3 p.

MIZUOCHI, T. et coll. (mars/avril 2004) *Forward Error Correction Based on Block Turbo Code With 3-Bit Soft Decision for 10-Gb/s Optical Communication Systems*, IEEE Journal of selected topics in quantum electronics, vol. 10, n° 2, p. 376-386.

NOÉ, R., SANDEL, D., MIRVODA, V. (mars/avril 2004) *PMD in High-Bit-Rate Transmission and Means for Its Mitigation*, IEEE Journal of Selected Topics in Quantum Electronics, vol. 10, n° 2, p. 341-355.

OUCHI, K., KUBO, K., MIZUOCHI, T., MIYATA, Y., YOSHIDA, H., TAGAMI, H., SHIMIZU, K., KOBAYASHI, T., SHIMOMURA, K., ONOHARA, K., MOTOSHIMA, K. (mars 2006) *A Fully Integrated Block Turbo Code FEC for 10 Gb/s Optical Communication Systems*, Optical Fiber Communication Conference OFC 2006, Anaheim, OTuK4, 3 p.

PUREUR, D., DELEVAQUE, E., MUGNIER, A. (mai 2004) *La compensation de dispersion chromatique pour les systèmes de transmission haut débit*, 72<sup>e</sup> congrès de l'ACFAS, Montréal, 6p.

PYNDIAH, R.M. (août 1998) *Near-Optimum Decoding of Product Codes : Block Turbo Codes*, IEEE Transactions on communications, vol. 46, n° 8, p. 1003-1010.

RAMASWAMI, R., SIVARAJAN, K. (2001) *Optical Networks : A Practical Perspective*, San Francisco, 2<sup>e</sup> édition, Morgan Kaufmann, 864 p.

SMITH, P. J., SHAFI, M., GAO, H. (mai 1997) *Quick Simulation : A Review of Importance Sampling Techniques in Communications Systems*, IEEE Journal on Selected Areas in Communications, vol. 15, n° 4, p.597-613.

SPINNLER, B., KRUMMRICH, P. M., SCHMIDT, E.-D. (mars 2006) *Chromatic Dispersion Tolerance of Coherent Optical Communications Systems With Electrical Equalization*, Optical Fiber Communication Conference OFC 2006, Anaheim, OWB2, 3 p.

TAGAMI, H., KOBAYASHI, T., MIYATA, Y., OUCHI, K., SAWADA, K., KUBO, K., KUNO, K., YOSHIDA, H., SHIMIZU, K., MIZUOCHI, T., MOTOSHIMA, K. (août 2005) *A 3-bit Soft-Decision IC for Powerful Forward Error Correction in 10-Gb/s Optical Communication Systems*, IEEE Journal of Solid-State Circuits, vol. 40, n° 8, p.1695-1705.

WANG, J., KAHN, J. M. (mai 2004) *Performance of Electrical Equalizers in Optically Amplified OOK and DPSK Systems*, IEEE Photonics Technology Letters, vol. 16, n° 5, p. 1397-1399.

WINZER, P. J., FIDLER, F., MATTHEWS, M. J., NELSON, L. E., THIELE, H. J., SINSKY, J. H., CHANDRASEKHAR, S., WINTER, M., CASTAGNOZZI, D., STULZ, L. W., BUHL, L. L. (janvier 2005) *10-Gb/s Upgrade of Bidirectional CWDM Systems Using Electronic Equalization and FEC*, Journal of Lightwave Technology, vol. 23, n° 1, p. 203-210.