TABLE DES MATIÈRES

Page

SOMMAIRE i

ABSTRACT ii

REMERCIEMENTS iii

TABLE DES MATIÈRES v

LISTE DES TABLEAUX viii

LISTE DES FIGURES ix

LISTE DES ABRÉVIATIONS ET SIGLES xii

INTRODUCTION 1

CHAPITRE 1 APERÇU D'UN SYSTÈME DE COMMUNICATION NUMÉRIQUE M-QAM 4

1.1 Introduction 4
1.2 Codage et décodage de canal 5
1.3 Modulation en bande de base 5
1.3.1 Modulation M-QAM 6
1.4 Mise en forme, interpolation et décimation 9
1.4.1 Filtre de mise en forme 9
1.5 Translation en fréquence 12
1.6 Canal de transmission 13
1.6.1 Bruit blanc gaussien additif 13
1.6.2 Propagation multi chemin 14
1.6.2.1 Effet Doppler 15
1.6.2.2 Étalement des délais 16
1.6.2.3 Sélectivité du canal 17
1.6.2.4 Atténuation et évanouissement du canal 17
1.6.3 Modèle SUI 17
1.7 Contrôleur automatique de gain numérique 19
1.8 Égalisation de canal 19
1.8.1 Topologies d'égaliseurs basés sur le critère EQMM 20
1.8.2 Égaliseur à échantillonnage fractionné 22
1.8.3 Égaliseur à longueur finie 23
1.9 Conclusion 25

CHAPITRE 2 ÉGALISATION ADAPTATIVE 27

2.1 Introduction 27
2.2 Filtre de Wiener 27
2.3 L'algorithme LMS 30
2.3.1 Convergence de l'algorithme LMS 31
2.3.2 Complexité de l'algorithme LMS 33
2.4 Les algorithmes CMA et MMA 34
2.4.1 Convergence des algorithmes CMA et MMA 37
2.4.2 Complexité de calcul des algorithmes CMA et MMA 37
2.5 L'algorithme BLMS 38
2.5.1 Convergence de l'algorithme BLMS 38
2.5.2 Complexité de calcul de l'algorithme BLMS 39
2.6 L'algorithme FBLMS 39
2.6.1 Convolution dans le domaine fréquentiel 39
2.6.1.1 Méthode overlap-save 40
2.6.2 Adaptation des coefficients FBLMS 41
2.6.3 Convergence de l'algorithme FBLMS 43
2.6.3.1 Convergence de l'algorithme FBLMS non contraint 43
2.6.4 Complexité de calcul de l'algorithme FBLMS 45
2.6.4.1 Complexité de calcul d'une FFT/IFFT 45
2.7 Complexité de l'algorithme FBLMS versus l'algorithme LMS en bloc 49
2.8 Conclusion 51

CHAPITRE 3 CONCEPTION ET SIMULATION EN VIRGULE FLOTTANTE D'UN ÉGALISEUR FRÉQUENTIEL 52

3.1 Introduction 52
3.2 Environnement de simulation 52
3.3 Conception de l'architecture en point flottant de l'égaliseur fréquentiel 60
3.3.1 Réduction de la complexité 61
3.3.1.1 Effet de la longueur de l'égaliseur sur la convergence de l'EQM 67
3.3.2 Égalisation fréquentielle autodidacte 70
3.3.2.1 L'algorithme FBMMA3M 71
3.3.3 Pas d'adaptation variable 76
3.4 Conclusion 79

CHAPITRE 4 QUANTIFICATION ET LATENCE DE L'ÉGALISEUR FBMMA3M 80

4.1 Introduction 80
4.2 Stratégie de quantification de l'égaliseur FBMMA3M 80
4.2.1 Quantification de l'erreur MMA 85

4.2.2 Quantification de l'erreur LMS 86
4.2.3 Quantification du seuil d pour la sélection MMA-LMS 86
4.2.4 Impact de la précision sur la convergence de l'EQM 87
4.3 Latence de l'algorithme FBMMA3M 91
4.3.1 Complexité de l'algorithme FBMMA3M sans latence 91
4.3.2 Effet de la latence sur l'égaliseur FBMMA3M 92
4.4 Conclusion 95

CHAPITRE 5 RÉALISATION DE L'ÉGALISEUR FBMMA3M 97

5.1 Introduction 97
5.2 Stratégie d'implémentation et de vérification 98
5.2.1 VHDL 98
5.2.2 Implémentation 99
5.2.3 Vérification pré synthèse 100
5.3 Description des modules implémentés 105
5.3.1 Module de chevauchement et sauvegarde 107
5.3.2 Choix du module de FFT 109
5.3.3 Modules r22pcx 110
5.3.3.1 Ordonnancement 112
5.3.3.2 Largeur des bus internes et mise à l'échelle 113
5.3.3.3 Latence 115
5.3.3.4 Initialisation du module r22pc 115
5.3.3.5 Complexité multiplicative du module r22pc 116
5.3.4 Module de temporisation des données 117
5.3.4.1 Synchronisation à l'initialisation 118
5.3.4.2 Configuration du module sfifo 119
5.3.5 Module de multiplication complexe 119
5.4 Module de sélection des données 122
5.4.1 Modules de gestion de l'erreur 123
5.4.1.1 Module de décision 124
5.4.1.2 Module de l'erreur MMA 127
5.4.1.3 Module de sélection de l'erreur (mux_mma_lms) 128
5.4.2 Module d'insertion de zéros 129
5.4.3 Module de contrainte du gradient 130
5.4.4 Module d'adaptation des coefficients 131
5.4.5 Module de temporisation des coefficients 133
5.5 Configuration et programmation de l'égaliseur FBMMA3M 135
5.5.1 Complexité de l'égaliseur 141
5.5.1.1 FPGA XCV2P40-7 141
5.5.1.2 Estimation théorique des ressources requises 142
5.5.1.3 Ressources requises après placement routage 143
5.6 Quelques résultats de performances de l'égaliseur FBMMA3M 145
5.7 Conclusion 149

CONCLUSION..... 151
BIBLIOGRAPHIE..... 155

LISTE DES TABLEAUX

Page

Tableau I Classification des canaux SUI 19

Tableau II Sommaire des étapes de calcul de l'algorithme LMS 30

Tableau III Sommaire des multiplications requises dans l'algorithme LMS 34

Tableau IV Sommaire des étapes de calcul de l'algorithme FBLMS 42

Tableau V Sommaire des multiplications réelles requises dans l'algorithme FBLMS avec la méthode de chevauchement et sauvegarde à 50% 49

Tableau VI Configuration du canal SUI-3 55

Tableau VII Configuration du canal SUI-5 55

Tableau VIII Rapport signal sur bruit blanc inséré pour déterminer la quantification de l'égaliseur 81

Tableau IX Seuil de décision (d) entre l'algorithme MMA et LMS 87

Tableau X Détail des longueurs requises pour la partie entière des registres pour chaque point de quantification 90

Tableau XI Succession des états de la MEF du bloc de chevauchement et sauvegarde 108

Tableau XII Illustration du concept d'ordonnancement pour une transformée de Fourier de 8 points 112

Tableau XIII Configuration de l'ordonnancement des cinq transformées de Fourier dans l'égaliseur FBMMA3M 113

Tableau XIV Paramètres programmables et configurables de l'égaliseur FBMMA3M 137

Tableau XV Configuration de l'égaliseur FBMMA3M pour les figures 65, 66 et 67 146

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

LISTE DES FIGURES

Page

Figure 1 Aperçu générique d'un système de communication numérique sans-fil.....5
Figure 2 Constellation 4, 16, 32, 64 QAM..... 8
Figure 3 Taux d'erreur par symboles pour 4, 16, 64, 256 QAM..... 10
Figure 4 Réponse en fréquence du filtre à cosinus surélevé..... 13
Figure 5 Modèle d'un canal de télécommunication..... 15
Figure 6 Schéma bloc simplifié d'un modem dans le domaine des fréquences..... 20
Figure 7 Schéma bloc d'un égaliseur linéaire (LTE).....22
Figure 8 Schéma bloc d'un égaliseur à retour de décision (DFE)..... 23
Figure 9 Spectre à l'entrée de l'égaliseur (C(f) = 1)..... 24
Figure 10 Filtre de Wiener.....28
Figure 11 Rayon de convergence de l'algorithme CMA.....35
Figure 12 Problèmes de convergence avec l'algorithme CMA.....36
Figure 13 Filtrage linéaire en fréquence par la méthode overlap-save avec N' = 2N... 41
Figure 14 Schéma bloc de l'égaliseur FBLMS avec N' = 2N.....42
Figure 15 Opération papillon radix-2.....47
Figure 16 Opération papillon radix-4 pour une décimation en fréquence..... 48
Figure 17 Complexité multiplicative des algorithmes FBLMS (chevauchement 50%) et BLMS.....50
Figure 18 Étapes permettant de générer les coefficients du canal SUI..... 54
Figure 19 Amplitude et phase de coefficients SUI-3 sur une période de une seconde. 56
Figure 20 Amplitude et phase de coefficients SUI-5 sur une période de une seconde. 57
Figure 21 Amplitude réelle et imaginaire des coefficients des canaux SUI-3 et SUI-5 utilisés pour les analyses et vérification de l'algorithme FBMMA3M.....57
Figure 22 Schéma bloc du CAGN intégré dans le simulateur..... 58

Figure 23 Modèle de simulation des différentes architectures d'égaliseurs fréquentiels 59
Figure 24 Schéma bloc de l'égaliseur FBLMS avec N' = 2N 61
Figure 25 Schéma bloc de l'égaliseur FBLMS avec calcul de l'erreur directement en fréquence et et N' = 2N 62
Figure 26 Constellation avant et après l'égaliseur FBLMS avec calcul de l'erreur directement en fréquence et N' = N 63
Figure 27 Courbe de convergence des algorithmes FBLMS contraint (CFBLMS) et non contraint (UFBLMS). a) Canal SUI-5, 64 coefficients d'égaliseur b) Canal SUI-5, 256 coefficients d'égaliseur 65
Figure 28 Coefficients des égaliseurs CFBLMS vs UFBLMS 67
Figure 29 EQM de l'égaliseur FBLMS lorsque soumis au canal SUI-3 69
Figure 30 Algorithme FBMMA3M 72
Figure 31 Sélection de l'erreur avec l'algorithme basé sur la sortie de l'égaliseur 74
Figure 32 EQM estimée en mode MMA-LMS 75
Figure 33 EQM estimée (MMA-LMS) vs MMA 75
Figure 34 EQM estimé avec pas d'adaptation variable 78
Figure 35 Pas d'adaptation variable quantifié et non quantifié 78
Figure 36 Points de quantification de l'algorithme FBMMA3M 84
Figure 37 Erreur MMA 86
Figure 38 EQM en fonction de la précision fractionnaire pour une modulation 4-QAM 88
Figure 39 EQM en fonction de la précision fractionnaire pour une modulation 64-QAM 89
Figure 40 Comportement de l'EQM lorsque le gradient est retardé (D = 5) 94
Figure 41 Comportement de l'EQM lorsque le gradient est retardé (D = 3) 95
Figure 42 Diagramme bloc du processus d'implémentation et outils de CAO 100
Figure 43 Diagramme bloc du processus d'implémentation et outils de CAO 103

Rapport-Gratuit.com

Figure 44 Exemple de courbe de vérification point flottant vs VHDL pour le module OSB 104
Figure 45 Exemple de courbe de vérification point flottant vs VHDL pour le module update_coeffs 104
Figure 46 Schéma d'implémentation de l'égaliseur FBMMA3M 106
Figure 47 Module de chevauchement et sauvegarde 107
Figure 48 Module de transformée de Fourier directe et inverse 110
Figure 49 Chronogramme du module r22pc pour N = 8 116
Figure 50 Module de temporisation des données 117
Figure 51 Module de multiplication complexe 119
Figure 52 Schéma d'implémentation du module de multiplication complexe 121
Figure 53 Module de sélection des données 122
Figure 54 Modules de décision et de l'erreur MMA 124
Figure 55 Configuration de la table de de décision pour une modulation 64-QAM 127
Figure 56 Configuration de la table de référence de l'erreur MMA pour une modulation 64-QAM 128
Figure 57 Module d'insertion de zéros 129
Figure 58 Module module de contrainte du gradient 130
Figure 59 Module d'adaptation des coefficients 131
Figure 60 Module de temporisation des coefficients 133
Figure 61 Module FBMMA3M 135
Figure 62 Exemple d'initialisation du module FBMMA3M 140
Figure 63 Configuration d'une slice d'un Virtex II Pro. (Tiré de (Xilinx, 2005)) 141
Figure 64 Complexité matérielle de l'égaliseur FBMMA3M avec gradient contraint et non contraint 145
Figure 65 Résultats en sortie du module FBMMA3M avec une constellation de 4-QAM et N = 4 147
Figure 66 Résultats en sortie du module FBMMA3M avec une constellation de 16-QAM et N = 64 148

Figure 67 Résultats en sortie du module FBMMA3M avec une constellation de 64-QAM et N = 256 149
Figure 68 Coexistence OFDM-SC-FDE, tiré de (Falconeer, 2002) 154

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

LISTE DES ABRÉVIATIONS ET SIGLES

Abréviation	Définition
α	Facteur d'excès de bande passante
a_k	Coefficients du canal de transmission
ASIC	Application specific integrated circuit
B_c	Bande de cohérence d'un canal de transmission
BLMS	Block least mean square
BMULT	Block multiplier
bps	Bit par seconde
BRAM	Block RAM
$C(f)$	Fonction de transfert d'un canal de transmission
CAGN	Contrôleur automatique de gain numérique
CAO	Conception assisté par ordinateur
CIC	Cascaded integrator-comb
CDMA	Code division multiple access
CFBLMS	Constrained fast block least mean square
CMA	Constant modulus algorithm
CPI	Cyclic prefix index
$d(n)$	Signal désiré
d^e_{min}	Distance euclidienne minimum
$\nabla J(\boldsymbol{w})$	Gradient de la fonction de coût
DFE	Decision feedback equalizer
$e(n)$	Erreur en sortie de l'égaliseur
EQM	Erreur quadratique moyenne
EQMM	Erreur quadratique moyenne minimum
$\boldsymbol{e}_F(k)$	Vecteur d'erreur dans le domaine des fréquences
E_s	Énergie moyenne d'un symbole
F	Transformée de Fourier directe
F^{-1}	Transformée de Fourier inverse
FBLMS	Fast block least mean sqare
FBMMA3M	Fast block multi modulus algorithm three modes
f_d	Fréquence relative de Doppler
FFT	Fast Fourier transform
FIR	Filtre à réponse impulsionnelle finie
f_m	Étalement en fréquence de Doppler
F_{max}	Fréquence de Nyquist
FPGA	Field programmeble gate array
F_s	Fréquence d'échantillonnage
$g(n)$	Gain du contrôleur automatique de gain numérique

$G_T(f)$	Fonction de transfert du filtre d'émission en cosinus surélevé
$G_R(f)$	Fonction de transfert du filtre de réception en cosinus surélevé
I	Composante en phase
I_s	Taux d'interpolation du filtre RRC
IES	Interférence entre symboles
IEEE	Institute of electrical engineering
IF	Fréquence intermédiaire
IFFT	Inverse fast Fourier transform
IIR	Infinite impulse response
ISI	Inter symbol interference
$J(\boldsymbol{w})$	Fonction de coût
λ_{max}	Valeur propre maximale de la matrice d'autocorrélation $\boldsymbol{R}$
λ_{min}	Valeur propre minimale de la matrice d'autocorrélation $\boldsymbol{R}$
L	Longueur du vecteur de chevauchement
LMS	Least mean square
LOS	Line of sight
LTE	Linear transversal equalizer
sym/s	Symboles par secondes
μ	Pas d'adaptation de l'égaliseur
M	Niveau d'une constellation ou Misadjustment
MAC	Multiplicateur accumulateur
MEF	Machine à état fini
MMA	Multi modulus algorithm
η	Bruit du canal
N	Nombre de coefficients temporels d'égaliseur
N'	Nombre de coefficients fréquentiels d'égaliseur
nbCoeffs	Nombre de coefficients du filtre RRC
nbSym	Nombre de symboles couverts par le filtre RRC
NLOS	None line of sight
N_0	Densité spectrale de puissance du bruit du canal
OFDM	Orthogonal frequency division multiplexing
$\boldsymbol{p}$	Matrice d'intercorrélation l'entrée et l'erreur
$P_{\sqrt{M}}$	Probabilité d'erreur d'un symbole PAM
P_{M-QAM}	Probabilité d'erreur d'un symbole M-QAM
PAM	Pulse amplitude modulation
PSK	Phase shift keying
Q	Composante en quadrature
QAM	Quadrature amplitude modulation
$r(n)$	Signal reçu à l'entrée de l'égaliseur
$\boldsymbol{R}$	Matrice d'autocorrélation du signal d'entrée
RAM	Random access memory
$\boldsymbol{r}_F(k)$	Signal reçu dans le domaine des fréquences

$R2^2PC$	Radix- 2^2 parameterizable core
$R_n(\tau)$	Fonction d'autocorrélation
RC	Raised cosine
R_{CMA}	Rayon CMA
R_{MMA_1}	Rayon MMA
RF	Fréquence radio
RMS	Root mean square
RRC	Root raised cosine
$\tilde{s}(k)$	Résultat de la IFFT sur $y_F(k)$
$\hat{s}(n)$	Symbole détecté à la sortie de l'égaliseur
SC−FDE	Single carrier frequency domain equalizer
SQNR	Rapport signal sur bruit de quantification
σ_τ	Étalement des délais d'un canal de transmission
SNR	Rapport signal sur bruit
SUI	Stanford university interim
τ_k	Délais du canal de transmission
τ_{rel}	Étalement relatifs des échos en fonction du taux de transmission
T_{sym}	Période d'un symbole
TEB	Taux d'erreur binaire
TES	Taux d'erreur par symboles
${}_u w_F(k)$	Coefficients fréquentiels non contraint de l'égaliseur
UFBLMS	Unconstrained fast block least mean square
VHDL	Very high speed integrated hardware description language
w_{opt}	Coefficients optimums de l'égaliseur
$w(n)$	Coefficients de l'égaliseur dans le domaine du temps
$w_F(k)$	Coefficients fréquentiels de l'égaliseur
W_N	Facteurs de phases d'une transformée de Fourier
$x_{rc}(t)$	Filtre à cosinus surélevé
$y_F(k)$	Résultat de la convolution fréquentielle
ZFE	Zero forcing equalizer

INTRODUCTION

La dispersion des symboles engendrée par un canal de transmission par voie Hertzienne est proportionnelle au taux de transmission des symboles, (Dmochowski & McLane, 2001). Par exemple, posons un lien de communication caractérisé par un environnement à multiples chemins dont le délai effectif entre le trajet principal et ses échos est d'une micro seconde. La dispersion des symboles vue par un récepteur cadencé à un méga symboles par seconde dont l'égaliseur effectue son traitement deux fois plus rapidement se situera autour de deux symboles. Si toutefois on décuple le taux de transmission, le même égaliseur percevra une dispersion dix fois plus grande.

Seul un égaliseur de longueur infinie peut annuler complètement les distorsions occasionnées par l'interférence entre symboles, (Proakis, 2001). Ainsi, plus le filtre d'un égaliseur est long, meilleur est la probabilité de détection du symbole transmis.

Les deux affirmations exposées aux moyens des deux paragraphes ci haut nous incitent à accomplir la conception, la simulation et la réalisation d'un égaliseur basé sur l'algorithme FBLMS. En effet, l'algorithme FBLMS est une traduction fréquentielle de la technique temporelle LMS et la complexité de calcul requise par la méthode FBLMS est $O\{Nlog_2 N\}$ versus $O\{N^2\}$ pour son homonyme temporel. Cette caractéristique permet de déployer une topologie d'égaliseur couvrant une grande dispersion en nombre de symboles de façon moins coûteuse en terme de calcul.

L'objectif de ce projet de recherche est d'implémenter un noyau matériel d'égaliseur fréquentiel configurable de 4 à 256 coefficients avec des modulations programmables de 4 à 64 QAM. Afin de concevoir l'égaliseur, nous avons réalisé un modèle de simulation dans lequel nous avons intégré un simulateur de canal SUI pour analyser les différents paramètres et également pour déterminer l'architecture de l'égaliseur.

Ce travail fait évoluer la recherche dans le domaine des télécommunications puisque nous avons conçu un noyau d'égaliseur basé sur l'algorithme FBLMS en langage de description matérielle (VHDL) que l'on nomme FBMMA3M. En plus d'avoir accompli une implémentation matérielle d'un égaliseur dans le domaine fréquentiel, nous avons modifié la structure de l'algorithme FBLMS afin de permettre un traitement des données de manière totalement sérielle et par conséquent, capable d'accommoder des débits très élevés (37.5 Msym/s). Par surcroît, la topologie de l'égaliseur développé est montée de manière à ce que les coefficients puissent être adaptés en mode aveugle, en mode dirigé et en mode entraîné. À l'appui de ce module d'égaliseur fréquentiel, nous avons construit un modèle de simulation en bande de base pour une modulation M-QAM, dans lequel on a implémenté un simulateur de canal SUI.

Le premier chapitre de ce mémoire présente les principes de base concernant les communications numériques. Une attention particulière est portée à l'explication des concepts liés à l'égalisation tels que la propagation à multiples chemins et l'interférence entre symboles. Nous nous attardons également à montrer, par le biais d'un exemple, les avantages d'utiliser un long filtre adaptatif pour accomplir l'égalisation.

Le deuxième chapitre est consacré à l'explication théorique de l'égalisation basée sur les algorithmes LMS, CMA et MMA. Les détails permettant de traduire l'algorithme LMS dans le domaine fréquentiel y sont insérés. Une analyse théorique de la complexité de calcul des algorithmes LMS et FBLMS complète ce chapitre.

Le modèle de simulation développé pour caractériser l'égaliseur FBMMA3M est présenté au troisième chapitre. Les différentes topologies étudiées ont été analysées et simulées avec ce modèle. Nous présentons les grandes lignes de nos réflexions à ce sujet. Les résultats de paramétrisations de l'algorithme FBMMA3M sont également à l'honneur dans cette section.

Le chapitre quatre porte sur les stratégies théoriques et pratiques employées pour quantifier l'égaliseur FBMMA3M. Nous exposons également l'effet de la latence sur la convergence des coefficients.

La stratégie d'implémentation matérielle est exposée au chapitre cinq. Les différents logiciels utilisés pour cette tâche y sont cités de même que la méthodologie employée pour en tirer profit. Chaque sous module du noyau FBMMA3M est expliqué. Une analyse pratique de complexité matérielle est présentée. Puis, pour terminer ce chapitre, nous illustrons quelques résultats de convergences de l'égaliseur FBMMA3M implémenté dans une puce programmable de Xilinx, numéro XCV2P40-7.

En conclusion, nous effectuons la synthèse globale du travail accompli. Nous abordons également quelques contextes dans lesquelles il serait intéressant de poursuivre les recherches élaborées dans le cadre de ce projet de maîtrise.

CHAPITRE 1

APERÇU D'UN SYSTÈME DE COMMUNICATION NUMÉRIQUE M-QAM

1.1 Introduction

Il sera démontré dans ce chapitre que l'égalisation est utilisée principalement dans le but de réduire les interférences entre symboles (IES). On montrera également que le nombre de symboles dispersés par un canal de transmission donné augmente avec le taux de symboles émis par secondes.

Les techniques d'égalisations sont très variées, (Proakis, 2001; Qureshi, 1985). On a recourt à ces techniques dans les systèmes de communication numériques tels que le PSK, le QAM, le CDMA, l'OFDM, etc... Ce chapitre expose les principales composantes d'un système de communication M-QAM, et ce dans le but de permettre au lecteur de mieux cerner les objectifs de l'égalisation dans un système de communication sans fil. L'égalisation basée sur le critère de l'erreur quadratique moyenne sera élaborée plus en détail.

La figure 1 illustre le traitement numérique et analogique que subit l'information à transmettre dans une chaîne de communication numérique typique. Chaque bloc effectue une tâche essentielle. Les sous sections suivantes sont consacrées d'une part à la définition générique de ceux-ci et d'autre part, on éttofera les modules intimement liés à l'égalisation.

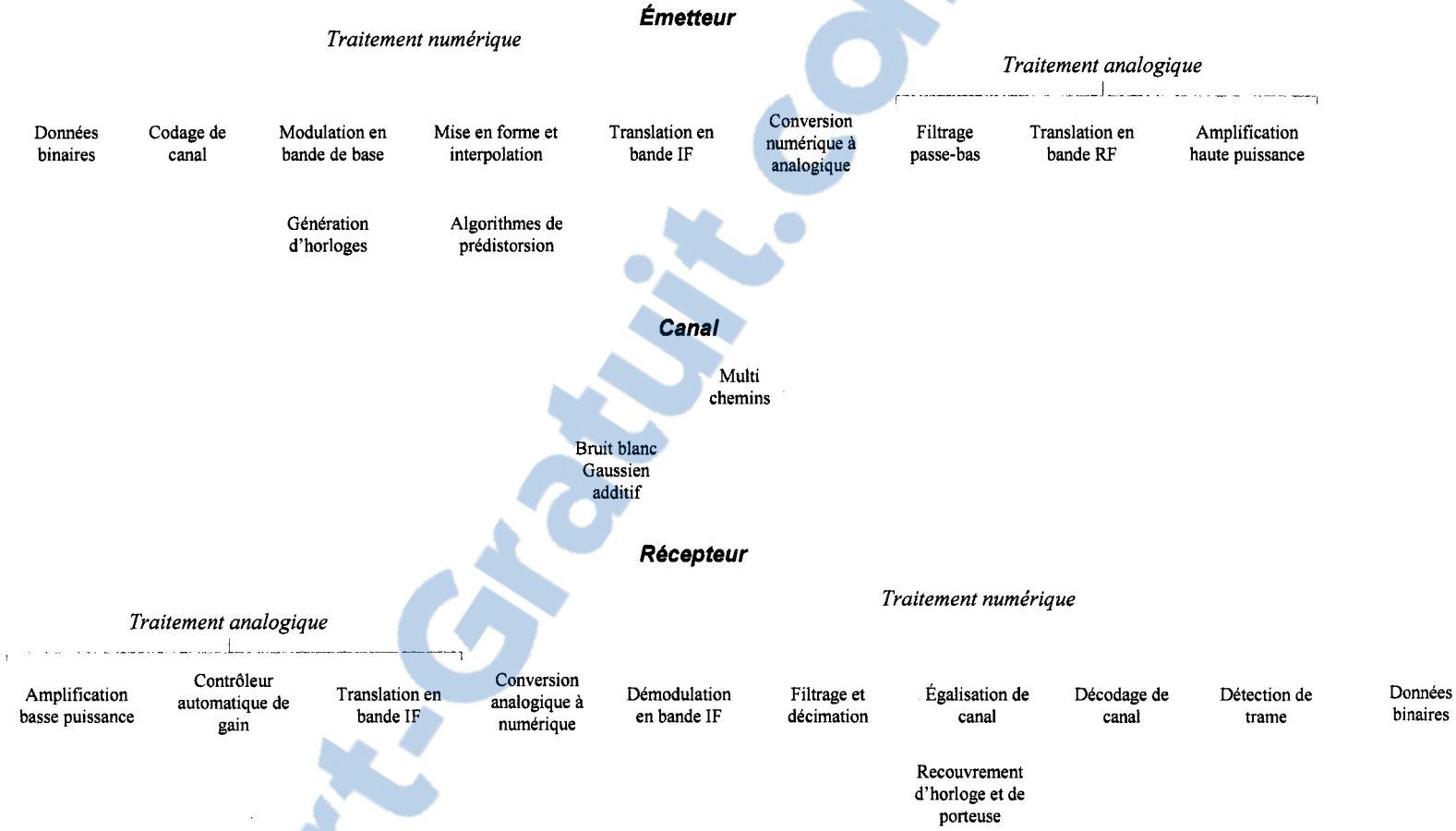


Figure 1 Aperçu générique d'un système de communication numérique sans-fil

Rapport-Gratuit.com

1.2 Codage et décodage de canal

Le codage de canal est utilisé pour améliorer les performances du lien de communication, (Proakis, 2001). Le gain visé est en fait de réduire les erreurs sur l'information binaire décodée au récepteur causées par les effets brouilleurs du canal. Il existe plusieurs algorithmes de codage de canal. Parmi les plus courants, on cite le codage convolutionnel et le codage en bloc.

1.3 Modulation en bande de base

La modulation en bande de base permet d'améliorer l'efficacité spectrale du signal à transmettre. L'objectif est de regrouper plusieurs bits ensemble pour créer ce qu'on appelle un symbole. Dans la littérature, on distingue plusieurs techniques de modulation. Chaque méthode possède des avantages et des inconvénients. On choisit le type de modulation en fonction des besoins du lien de communication à implémenter. Parmi les critères de sélections, on note:

- le débit de transmission requis,
- le taux d'erreur binaire requis,
- la largeur de bande disponible.

Dans un contexte de radio configurable, la modulation M-QAM représente un choix efficace. Effectivement, en fonction de l'environnement de communication, on peut aisément configurer un niveau de constellation M simplement qu'en modifiant une table de référence qui peut être logée dans un bloc de mémoire vive. Ceci en conservant sensiblement la même architecture matérielle, (Joshi, R. B., 1992).

1.3.1 Modulation M-QAM

La modulation M-QAM rectangulaire est une combinaison de deux signaux $\sqrt{M}$-PAM. On agence les symboles PAM en coordonnées complexes dont les axes, réels et imaginaires, sont gradués comme suit:

$$\{I_i, Q_i\} \in (2\ i - 1 - \sqrt{M}), \quad i = 1, 2, \ldots, \sqrt{M}. \tag{1.1}$$

Les symboles M-QAM sont ainsi représentés en deux dimensions dont les indices s'expriment comme suit:

$$(I_i, Q_i) = \begin{bmatrix} (-\sqrt{M}+1, \sqrt{M}-1) & (-\sqrt{M}+3, \sqrt{M}-1) & \cdots & (\sqrt{M}-1, \sqrt{M}-1) \\ (-\sqrt{M}+1, \sqrt{M}-3) & (-\sqrt{M}+3, \sqrt{M}-3) & \cdots & (\sqrt{M}-1, \sqrt{M}-3) \\ \vdots & \vdots & \vdots & \vdots \\ (-\sqrt{M}+1, -\sqrt{M}+1) & (-\sqrt{M}+3, -\sqrt{M}+1) & \cdots & (\sqrt{M}-1, -\sqrt{M}+1) \end{bmatrix}, \tag{1.2}$$

où I et Q indiquent respectivement les coordonnées réelles et imaginaires. La figure 2 montre les constellations 4, 16, 32 et 64-QAM. La distance entre chaque point PAM de l'axe I et Q est normalisée selon le niveau $\sqrt{M}$.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

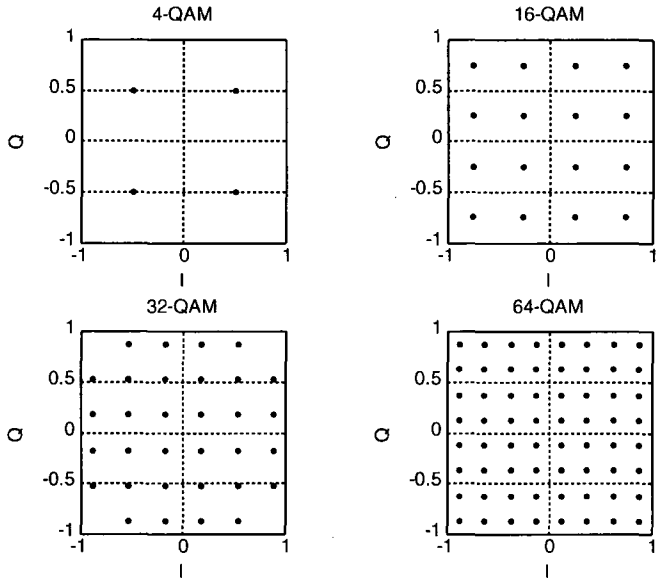


Figure 2 Constellation 4, 16, 32, 64 QAM

On remarque, sur la figure 2, que si M n'est pas un nombre carré, la constellation M-QAM prend une forme octogonale. Cette propriété peut s'avérer intéressante pour limiter les distorsions engendrées par les non-linéarités de l'amplificateur de puissance à l'émetteur, (Ghorbani & Sheikhan, 1991).

De l'équation 1.2, on observe que chaque symbole combine $\log_2 M$ bits. Ainsi, plus M est grand, plus l'efficacité spectrale est grande. Cependant, en examinant la figure 1, on note que la distance minimale entre chaque symbole diminue en fonction de M. Or, en présence de bruit blanc gaussien additif, la probabilité d'erreur lors de la détection des symboles augmente avec M. On montre que la probabilité d'erreur pour des constellations M-QAM peut être approximée par la borne supérieure suivante, (Proakis, 2001):

$$P_{M-QAM} < (M-1)\,Q\left(\sqrt{\frac{\left(d_{min}^{e}\right)^2}{2\,N_0}}\right) \tag{1.3}$$

où N_0 représente la densité spectrale de puissance du bruit du canal, d^e_{min} est la distance euclidienne minimale entre les points de la constellation et *Q(·)* correspond à la fonction de densité de probabilité d'une variable aléatoire de distribution gaussienne.

$$Q(x)=\frac{1}{\sqrt{2\pi}}\int_x^{\infty} e^{\frac{-x^2}{2}}\,dx \tag{1.4}$$

Notons que pour des constellations M-QAM rectangulaires, on a:

$$P_{M-QAM}=1-\left(1-P_{\sqrt{M}}\right)^2 \tag{1.5}$$

$P_{\sqrt{M}}$ dénote la probabilité d'erreur d'un symbole PAM en fonction de l'énergie moyenne, E_s, d'un symbole.

$$P_{\sqrt{M}}=\frac{2\left(\sqrt{M}-1\right)}{\sqrt{M}}Q\left(\frac{3E_s}{(M-1)N_0}\right) \tag{1.6}$$

La figure 3 illustre les courbes de probabilité d'erreur par symbole pour des constellations 4, 16, 64 et 256-QAM. On constate que pour obtenir un même taux d'erreur par symbole, le rapport signal sur bruit, E_s/N_0 requis augmente en fonction du niveau M.

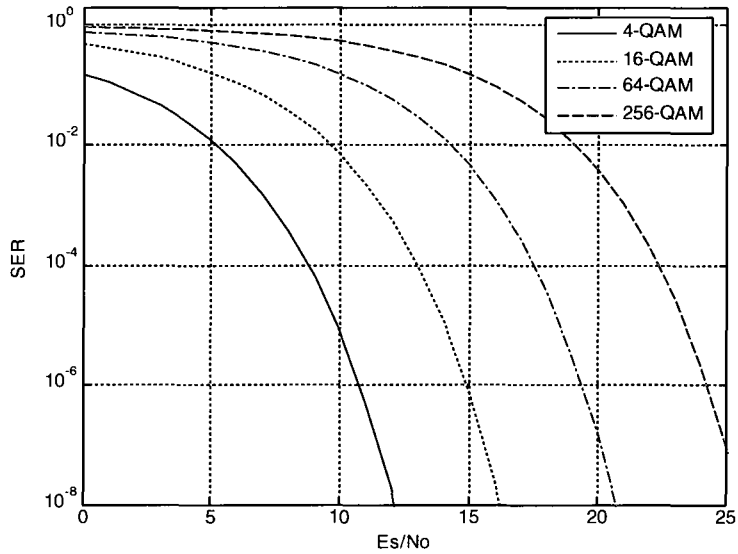


Figure 3 Taux d'erreur par symboles pour 4, 16, 64, 256 QAM

1.4 Mise en forme, interpolation et décimation

1.4.1 Filtre de mise en forme

Si le temps de transition entre les différents niveaux de symboles est nul, la réponse fréquentielle des symboles M-QAM est infinie et son amplitude prend la forme d'un *sinc(x)*. En pratique, cette propriété a pour effet de créer des oscillations entre chaque transition de symbole. Ces oscillations s'additionnent ou se soustraient directement aux symboles présents et futurs. C'est ce qu'on appelle l'interférence entre symboles (IES). Si on observe la partie échantillonnée réelle du signal M-QAM contenant de l'IES à la sortie du canal de transmission, on a:

$$y(n)=I(n)x(0)+\sum_{k=0,k\neq n}^{\infty} I(k)x(n-k)+\eta(n)\,. \tag{1.7}$$

Ainsi, le $n^{ième}$ symbole d'information est corrompu par tous les symboles passés et le bruit du canal η.

L'objectif du filtrage de mise en forme est donc de limiter la bande passante du signal transmis et de réduire l'IES causée par les transitions brusques entre chaque niveau de symbole. La condition nécessaire et suffisante pour l'IES nulle a été énoncée en 1924 par M. Harry Nyquist , (Nyquist H., 2002) et s'écrit comme suit:

$$x(n)=\begin{cases}1 & n=0\\ 0 & n\neq 0\end{cases}. \tag{1.8}$$

Pour une largeur de bande W donnée, Nyquist a également démontré que la plus petite période de symbole T pour laquelle on obtient une transmission avec IES nulle est bornée par:

$$T<\frac{1}{2\mathrm{W}}. \tag{1.9}$$

Le filtre à cosinus surélevé (RC) permet d'obtenir les caractéristiques spectrales basées sur les critères régis par les équations 1.8 et 1.9. Sa réponse à l'impulsion dans le temps continu est:

$$x_{rc}(t)=sinc\left(\frac{\pi t}{T}\right)\frac{\cos(\pi\alpha t/T)}{1-4\alpha^2 t^2/T^2} \tag{1.10}$$

α est le facteur d'excès de bande passante. Il est borné entre 0 et 1. Plus il est faible, plus on se rapproche de la borne fréquentielle 1.9 et vice versa.

Souvent, on sépare le filtre de mise en forme en deux entités que l'on dispose au transmetteur et au récepteur. Dans le cas spécial où le canal de transmission est idéal, on a:

$$X_{rc}(f)=G_T(f)G_R(f)=\sqrt{(X_{rc}(f))}\sqrt{(X_{rc}(f))} \tag{1.11}$$

$G_T(f)$ et $G_R(f)$ sont respectivement le filtre d'émission et de réception. En pratique, on implémente ce type de filtre à l'aide d'un FIR. Il existe plusieurs méthodes d'implémentation de cette architecture de filtre, (Proakis & Manolakis, 1996). Règle

générale, sans considérer la réutilisation de ressources, un multiplicateur-accumulateur (MAC) est requis pour chacun des coefficients. Le nombre de coefficients *nbCoeffs* nécessaire dépend de la quantité de symboles *nbSym* à couvrir et du taux d'interpolation du symbole I_s.

$$nbCoeffs = nbSym\, I_s \tag{1.12}$$

La réponse fréquentielle du filtre RC s'écrit:

$$X_{rc}(f) = \begin{cases} T & \left(0 \le |f| \le \frac{1-\alpha}{2T}\right) \\ \frac{T}{2}\left\{1 + \cos\left[\frac{\pi T}{\alpha}\left(|f| - \frac{1-\alpha}{2T}\right)\right]\right\} & \left(\frac{1-\alpha}{2T} \le |f| \le \frac{1+\alpha}{2T}\right) \\ 0 & \left(|f| > \frac{1+\alpha}{2T}\right) \end{cases}. \tag{1.13}$$

En observant l'équation 1.13, on remarque que pour des valeurs de α supérieures à zéro, la largeur de bande du signal excède la fréquence maximale du théorème d'échantillonnage de Nyquist. Rappelons que la fréquence maximale F_{max} doit être inférieure au double de la fréquence d'échantillonnage F_s. Autrement, il y a risque de chevauchement avec les répliques spectrales causées par l'échantillonnage. Cependant, la zone de coupure du filtre de Nyquist est symétrique. Ainsi, les composantes fréquentielles supérieures $F_s/2$ viennent s'additionner symétriquement aux répliques spectrales. La véritable réponse fréquentielle est alors une ligne droite sur toute la plage d'intérêt. La ligne en trait pâle de la figure 4 illustre ce phénomène.

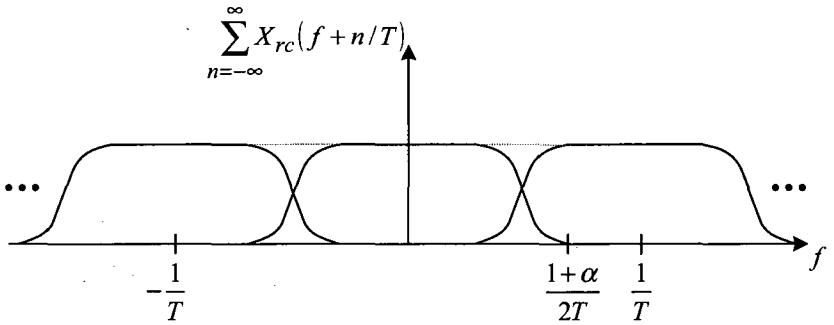


Figure 4 Réponse en fréquence du filtre à cosinus surélevé

La tendance actuelle vise à repousser les barrières du traitement numérique le plus proche possible de l'étage RF. En général, on discrétise le signal jusqu'à l'étage IF. Pour y arriver à l'émetteur, on doit augmenter la fréquence de traitement du signal au niveau de l'oscillateur IF. Une technique pour y arriver consiste à interpoler le signal au travers un filtre CIC. Ces types de filtres sont particulièrement intéressants d'un point de vue implémentation puisqu'ils ne nécessitent aucun multiplicateur et aucune gestion de coefficients, (Hogenauer, 1981)[1]. Cette particularité permet ainsi d'augmenter la fréquence d'échantillonnage du signal à un taux assez élevé pour effectuer le décalage en fréquence de manière entièrement numérique.

Au récepteur, on effectue l'opération inverse avec le même type de filtre. Ainsi, on décime le signal reçu jusqu'à une fréquence d'opération choisie pour le traitement en bande de base.

1.5 Translation en fréquence

Au transmetteur, la première étape de translation en bande passante se fait généralement entièrement numériquement. Les symboles I et Q sont multipliés par un oscillateur numérique générant deux signaux orthogonaux, sinus et cosinus, afin de conserver la

1 À noter que le filtre CIC offre une réponse en fréquence non linéaire de type *sinc(x)*. Si besoin il y a, on corrige cet effet à l'aide d'un filtre de compensation dont la réponse en fréquence est l'inverse de celle du filtre CIC.

distinction entre la partie réelle et imaginaire du signal M-QAM. La somme est convertie en analogique à l'aide d'un convertisseur numérique à analogique puis est acheminée à travers toute la chaîne de translation RF, (Stremler, 1992). L'étape inverse est effectuée au récepteur. La démodulation est accomplie à l'étage IF et le traitement se fait généralement de manière numérique.

1.6 Canal de transmission

Les canaux radio introduisent des limitations de performances aux systèmes de communications sans fil. La littérature consacre une attention particulière envers la caractérisation et la modélisation des canaux radios. Les chercheurs se basent sur des caractéristiques physiques aléatoires et variables dans le temps. Ainsi, la modélisation d'un canal de télécommunication est entièrement fondée sur des données statistiques.

1.6.1 Bruit blanc gaussien additif

Le bruit blanc est caractérisé comme un processus stochastique possédant une densité spectrale de puissance constante $N_0/2$ sur toute la bande de fréquence, d'où le qualitatif *blanc*. Sa densité de probabilité suit une loi normale de moyenne nulle et de variance $\sigma^2 = N_0/2$. Ses sources sont principalement les composantes électroniques qui constituent le système de communication. La fonction d'autocorrélation du bruit blanc gaussien est:

$$R_n(\tau) = \frac{N_0}{2} F^{-1}\{n(f)=1\} = \frac{N_0}{2}\delta(\tau) . \tag{1.14}$$

Ainsi, $R_n(\tau)$ est nul pour toutes valeurs de τ sauf zéro. Ceci implique que deux échantillons de bruits différents ne sont en aucun cas corrélés. Cette propriété est importante puisqu'elle permet de séparer la composante de bruit blanc des analyses mathématiques en termes indépendants et par surcroît, allège la conception du récepteur de façon notable.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

1.6.2 Propagation multi chemin

Les signaux radios sont transmis dans un environnement composé entre autres de montagnes, de gratte ciels, d'arbres et d'objets en mouvements. Ces obstacles entraînent des réflexions, des diffractions et des diffusions du signal transmis. Conséquemment, le signal emprunte différents trajets et arrive à l'antenne réceptrice à des instants différents. Ainsi, chaque symbole transmis est corrompu par les échos des symboles passés et futurs. Nous sommes à nouveau en présence d'interférences entre symboles. Ce phénomène introduit des changements aléatoires sur le signal reçu au niveau de l'amplitude, de la phase et du temps. En pratique, on modélise un canal de transmission par une ligne à délai à laquelle on ajoute un signal de bruit blanc gaussien η tel qu'illustré à la figure 5, (Proakis, 2001).

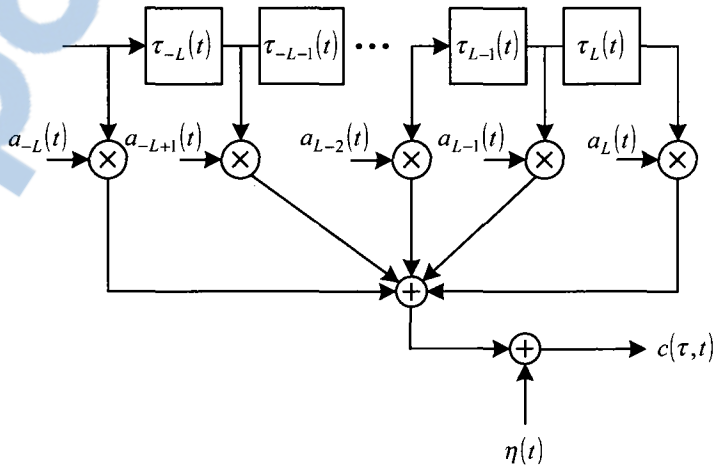


Figure 5 Modèle d'un canal de télécommunication

Le modèle mathématique en bande de base du canal est:

$$c(\tau,t)=\sum_{k=-L}^{L} a_k(t)\delta(\tau-\tau_k(t))+\eta(t)\,. \tag{1.15}$$

Les facteurs physiques qui influencent les coefficients a_k et les délais τ_k d'un canal sont basés sur des mesures à grande échelle pour caractériser l'affaiblissement du signal entre l'émetteur et le récepteur et des mesures à petite échelle, pour quantifier les effets reliés à la propagation multi chemin, (Rappaport, S. Theodore, 2002).

1.6.2.1 Effet Doppler

L'effet Doppler survient principalement lorsque le récepteur est en mouvement. L'exemple le plus cité pour expliquer ce phénomène est celui-ci: si un véhicule est en déplacement par rapport à une antenne, son angle de réception change en fonction de sa position. Ainsi, la fréquence relative de la porteuse varie proportionnellement à son angle d'incidence θ et au rapport de la vitesse v du véhicule sur la longueur d'onde λ du signal.

$$f_d=\frac{v}{\lambda}\cos(\theta) \tag{1.16}$$

La fréquence relative f_d varie en fonction des composantes multi trajets de la porteuse. Cette fluctuation est caractérisée par l'étalement Doppler, f_m.

L'effet Doppler peut aussi être perçu dans un environnement de transmission fixe. Dans ce contexte, ce sont les effets de dispersion du signal qui engendre le décalage de Doppler. Les obstacles entre les deux points de communication sont en mouvement aléatoire et ils entraînent intrinsèquement des variations sur la trajectoire du signal.

Le spectre de puissance de Doppler est différent pour une liaison mobile et une liaison fixe. Avec la liaison mobile, son allure est en forme de U borné par $\pm f_m$ et centré sur

la porteuse. Son modèle est nommé spectre de Jakes, (Rappaport, S. Theodore, 2002). La liaison fixe offre un spectre de Doppler en forme de cloche également centrée sur la porteuse et bornée bilatéralement par la fréquence d'étalement de Doppler.

1.6.2.2 Étalement des délais

La longueur des échos dépend des trajets empruntés par l'information transmise. Pour caractériser un canal, on se base principalement sur la mesure de l'étalement des délais RMS. Ce paramètre est défini par:

$$\sigma_\tau = \sqrt{\frac{\left(\left(\sum_{k=-L}^{L} a_k^2 \tau_k^2\right) - \left(\sum_{k=-L}^{L} a_k^2 \tau_k\right)^2\right)}{\sum_{k=-L}^{L} a_k^2}}. \tag{1.17}$$

Dans un environnement urbain, σ_τ peut se chiffrer jusqu'à 25 micro secondes, (Rappaport, S. Theodore, 2002).

Vu par le récepteur, le délai entre les échos est fonction du taux de symboles. Par exemple, si on suppose un taux de symbole *Ts* de 100 Ksym/s dans un canal à deux trajets espacés de 5 micro secondes, la dispersion τ_{rel} vue par le récepteur serait de 2 symboles. Maintenant, si on décuple le taux de transmission, la dispersion vue par le récepteur augmente à 20 symboles. L'équation 1.18 résume ce phénomène.

$$\tau_{rel} = \frac{\tau_k}{T_s} \tag{1.18}$$

Ainsi, pour un canal donné, la dispersion des symboles dépend du taux de transmission. Le concepteur d'un modem doit donc tenir compte de ce phénomène lorsqu'il établit les différents paramètres systèmes de la radio. L'égaliseur est notamment un module directement affecté par cette propriété.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

1.6.2.3 Sélectivité du canal

Un canal est sélectif en fréquence si la largeur de bande du signal transmis est supérieure à celle de l'étalement des délais $1/\tau_m$. Concrètement, on peut imaginer que deux sinusoïdes séparées en fréquence par un delta plus grand que $1/\tau_m$ seront affectées différemment par le canal. Ainsi, la bande de cohérence B_c d'un canal est inversement proportionnelle à τ_m.

1.6.2.4 Atténuation et évanouissement du canal

Tel que mentionné plus haut, la puissance du signal reçu dépend du contexte de propagation du canal. Les fluctuations sont principalement causées par les multiples trajets et l'effet Doppler. Lorsque la puissance du signal passe sous un seuil de tel sorte à ce qu'il soit quasi indétectable au récepteur, on dit que le signal s'est évanoui. Typiquement, on peut s'attendre à observer autour de f_m évanouissements ou affaissements de signal par secondes. Ainsi, la fréquence des évanouissements est intimement liée à l'étalement de Doppler.

Il existe plusieurs modèles qui caractérisent les fluctuations en puissance d'un canal. Typiquement, on utilise une distribution de Rayleigh pour les liaisons en vue non directe (NLOS) tandis que pour les liaisons en vue directe (LOS), on utilise une distribution de Rice, (Proakis, 2001). En pratique, les conditions de canal ne sont pas déterministes et ainsi, le comportement du canal peut passer d'une forme Rayleigh à Rice et vis et versa dans une même séquence de transmission.

1.6.3 Modèle SUI

Les canaux sans fil sont caractérisés principalement par l'atténuation du signal, l'étalement des délais, les fluctuations de puissances et l'étalement de Doppler. Sachant

que tous ces paramètres sont variables, on peut aisément imaginer une infinité de situations de propagation du signal. Le modèle SUI en est un qui regroupe plusieurs études concernant les caractéristiques mentionnées ci haut pour un contexte de communication fixe en vue non directe (NLOS). Les coefficients du canal sont déterminés selon une distribution de Rice à trois trajets. Six contextes différents ont été élaborés, (IEEE 802.16, 2003). Ces scénarios sont divisés en trois types de terrains tel que reporté au tableau I:

Tableau I

Classification des canaux SUI

Terrain	Type	Étalement des Délais	Étalement Doppler
A (Plat, faible densité d'obstacles) Facteur K élevé	SUI-1	*faible*	*faible*
	SUI-2	*faible*	*faible*
B (Plat, densité d'obstacles modérée) Facteur K modéré/faible	SUI-3	*faible*	*faible*
	SUI-4	*modéré*	*élevé*
C (Montagneux, densité d'obstacles élevée) Facteur K faible	SUI-5	*élevé*	*élevé*
	SUI-6	élevé	*faible*

Au tableau I, on remarque que plus l'indice du canal SUI est grand, plus le contexte est considéré hostile. Ainsi, un canal SUI-6 engendrera plus de distorsions sur le signal qu'un canal SUI-1. Le facteur K correspond au ratio de la puissance du trajet principal sur la puissance des trajets secondaires. Il dépend principalement du type d'antenne utilisé, de la distance entre les deux points de communication, du vend, du trafic et de la saison. Lorsque K tend vers zéro, la distribution de Rice s'approche de celle de

Rayleigh. Puisque le contexte de télécommunication SUI considère une liaison fixe, l'étalement de Doppler est faible par rapport aux valeurs typiques d'une liaison mobile. Dans le tableau I, les qualitatifs faibles et élevés sont relatifs aux scénarios d'une liaison fixe. Ainsi, pour un canal SUI, lorsqu'on spécifie un étalement de Doppler élevé, on fait référence à un f_m de l'ordre de deux à quatre Hertz.

Le modèle SUI peut être représenté comme une ligne à délai de la même forme que la figure 5 dont les coefficients sont obtenus aléatoirement selon les paramètres mentionnées ci-haut.

1.7 Contrôleur automatique de gain numérique

Le contrôleur automatique de gain numérique est utilisé pour ajuster la puissance de la constellation autour d'un niveau donné dans le but de maximiser la plage numérique définie.

1.8 Égalisation de canal

Les distorsions engendrées par le canal augmentent les probabilités d'erreur de détection d'une manière significative. Pour obtenir une transmission idéale, c'est à dire sans interférences entre symboles, la fonction de transfert totale d'un modem devrait être $X_{rc}(f)$. Cependant, ainsi que nous l'avons observé plus tôt, la propagation multi chemins de même que la mobilité de l'environnement apportent de l'IES. Si on globalise un système de communication numérique dans le domaine fréquentiel, on obtient le schéma bloc de la figure 6.

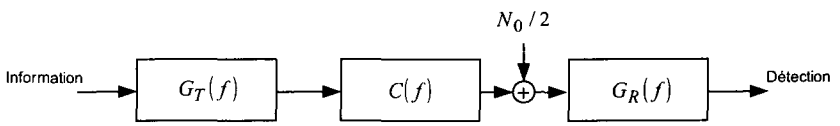


Figure 6 Schéma bloc simplifié d'un modem dans le domaine des fréquences

Afin de retrouver la fonction de transfert de Nyquist, de façon intuitive, on pourrait proposer d'annuler l'action de $C(f)$ en introduisant un bloc qui effectue les distorsions inverses de ce dernier. Ce raisonnement constitue la base de la théorie d'égalisation de canal. Ainsi, la fonction de transfert d'un égaliseur de longueur infinie aurait la forme

$$E(f)=\frac{1}{C(f)}. \tag{1.19}$$

L'équation 1.19 correspond à l'algorithme de forçage à zéro (ZFE). On note qu'en appliquant ce critère, l'inversion ne tient pas compte du SNR du signal. Ainsi, l'égalisation ZFE utilisée avec un canal sélectif en fréquence peut apporter des distorsions notables sur le signal, (Stremler, 1992). Une solution pour réduire le problème causé par les fluctuations du SNR consiste à minimiser l'erreur entre le symbole estimé et le symbole transmis. Cette approche prend en compte le bruit du canal et ainsi contribue davantage à diminuer l'interférence entre symboles. En négligeant les filtres d'émission et de réception, la fonction de transfert de l'égaliseur de longueur infinie basée sur le critère de l'erreur quadratique moyenne minimale est

$$E(f)=\frac{1}{C(f)+N_0}. \tag{1.20}$$

1.8.1 Topologies d'égaliseurs basés sur le critère EQMM

On distingue deux catégories d'égalisation de canal basé sur le critère de l'erreur quadratique moyenne minimale. Soit l'égalisation linéaire et non linéaire. La topologie avec filtre linéaire de type FIR (LTE) est présentée à la figure 7 et celle de type non linéaire à la figure 8. Pour les deux topologies, les coefficients des filtres sont calculés de manière à minimiser l'erreur de détection des symboles. La littérature démontre que le DFE surpasse le LTE à bien des égards. En effet, un égaliseur est caractérisé par son taux de convergence, ses aptitudes de poursuites, sa capacité à réduire l'erreur quadratique moyenne et sa stabilité. Ainsi, pour un même canal et un même algorithme

d'adaptation des coefficients, le DFE offre de meilleures caractéristiques, puisqu'il est muni d'un deuxième filtre (filtre arrière) qui permet de réduire l'IES résiduelle en sortie du filtre avant (Sayed, 2004). Cependant, les coefficients du DFE doivent être surveillés de manière étroite. En effet, l'ajout du filtre arrière dans la topologie DFE implique une réponse à l'impulsion de type IIR. Ainsi, on doit contrôler les coefficients de sorte que les zéros du filtre arrière ne puissent pas annuler le dénominateur de la fonction de transfert de l'égaliseur DFE, (Proakis, 2001).

Comme il a été montré plus tôt, les distorsions du canal sont variables dans le temps. Ceci implique qu'il est impossible d'obtenir des coefficients d'égaliseurs fixes dans le temps. Au chapitre 2, nous explorerons quelques algorithmes adaptatifs permettant de calculer les coefficients qui minimisent l'EQMM.

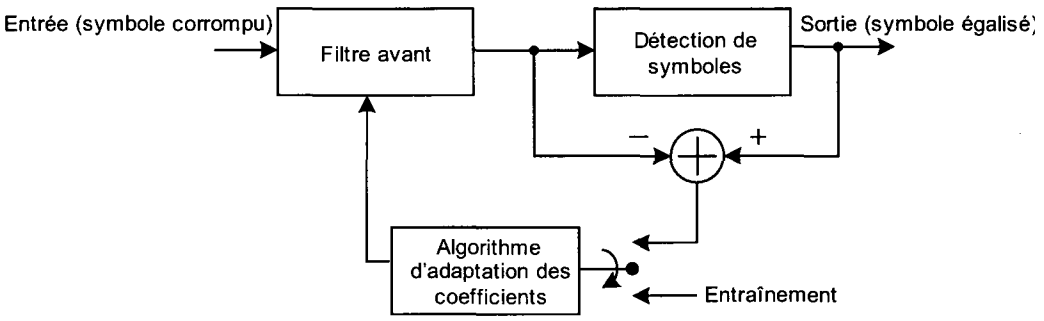


Figure 7 Schéma bloc d'un égaliseur linéaire (LTE)

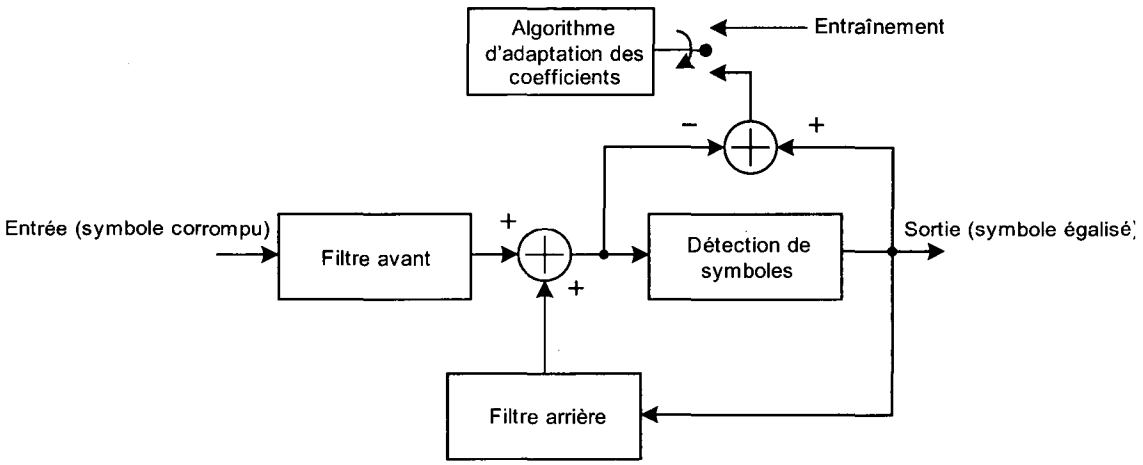


Figure 8 Schéma bloc d'un égaliseur à retour de décision (DFE)

1.8.2 Égaliseur à échantillonnage fractionné

Jusqu'à maintenant, nous avons supposé une synchronisation parfaite entre l'émetteur et le récepteur. En réalité, cette situation n'est pas facile à obtenir. Les circuits de génération d'horloge à l'émetteur, ainsi que ceux de recouvrement d'horloge et de porteuse au récepteur accomplissent la fonction de synchronisation. Les algorithmes qui les constituent ne sont pas parfaits. Les erreurs de phases doivent donc être constamment compensées. En plus de détériorer la probabilité d'erreur du lien de communication, (Noguchi, 1986), une erreur de phase τ_0 entraînent une dégradation du spectre en bande de base. Dans (Qureshi, 1985), on montre que les erreurs de phase introduisent une dissymétrie spectrale dans la zone de coupure du spectre. Avec un égaliseur échantillonné au taux de symbole, l'addition des empiétements provoque des distorsions constructives et destructives qui dégradent les performances de l'égaliseur.

Contrairement à l'égalisation échantillonnée au taux de symbole, avec l'égalisation à espacement fractionnaire, on cherche à échantillonner au taux de Nyquist. Par exemple, la fréquence maximale F_{max} d'un signal mis en forme par un filtre RC est

$$F_{max} = \frac{1+\alpha}{2T} . \tag{1.21}$$

En appliquant le théorème de Nyquist, le signal peut être échantillonné à $2F_{max}$. Ainsi, si α égale 1, l'égaliseur pourrait échantillonner le signal à $T_s/2$. De cette manière, les images spectrales ne se chevaucheront pas au spectre principal. La figure 9 a) illustre de manière heuristique l'effet d'une erreur de phase sur le spectre à l'entrée de l'égaliseur avec un canal parfait $(C(f)=1)$. La figure 9 b) montre le spectre à l'entrée d'un égaliseur fractionné à $T_s/2$. On constate que les spectres ne se chevauchent plus.

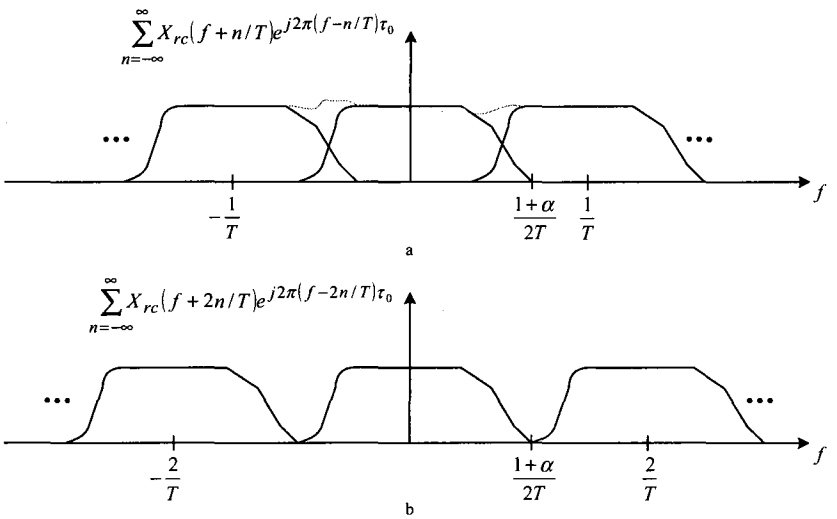


Figure 9 Spectre à l'entrée de l'égaliseur ($C(f) = 1$)
a- Échantillonnage à chaque symbole
b- Échantillonnage 2 fois par symbole

1.8.3 Égaliseur à longueur finie

Pour annuler complètement l'IES, il faudrait que l'égaliseur soit muni d'un nombre infini de coefficients. La convolution des coefficients $c(k)$ du canal avec ceux de l'égaliseur $w(k)$ devrait être

$$q(n)=\sum_{k=-\infty}^{\infty} w(k)c(n-k)=\begin{cases}1 & (n=0)\\ 0 & (n\neq 0)\end{cases}. \tag{1.22}$$

Toutefois, pour un égaliseur de longueur N, $q(n>1)$ n'est pas tout à fait nul. C'est ce qu'on appelle l'interférence résiduelle. Si on décrit la fonction de transfert d'un canal dans le domaine Z, on obtient la forme suivante:

$$C(z)=\sum_{k=-L}^{L} a_k z^{-k}+N_0 . \tag{1.23}$$

La fonction de transfert d'un égaliseur parfait serait 1.20. Cependant, le résultat de l'inversion d'un tel polynôme produit une suite infinie de coefficients. Les limites matérielles qu'imposent les unités de traitement numériques nous restreignent à une longueur d'égaliseur finie. On peut alors se demander combien de coefficients d'égaliseur permettraient de réduire l'IES à une valeur n'altérant pas trop la performance du modem. En communication sans fil, on ne connaît pas exactement le comportement du canal. Seules les données statistiques peuvent nous aider à dimensionner un égaliseur. Plusieurs critères doivent être considérés dans la sélection du nombre de coefficients d'un égaliseur. Ces critères dépendent du type d'égaliseur (DFE ou LTE) et de l'algorithme d'adaptation des coefficients. Il n'y a donc pas de solution déterministe associée à la sélection optimale de N.

Une solution plutôt empirique serait de choisir N en fonction de l'étalement des délais. Par exemple, posons un canal à un seul trajet secondaire de puissance a et espacé de Es symboles. En supposant un SNR élevé, on peut prendre le critère ZFE pour trouver la réponse impulsionnelle infinie de l'égaliseur.

$$E(z)=\frac{1}{1+az^{-Es}}=1+\sum_{k=0}^{\infty}\left[(-1)a\right]^k z^{-kEs} \tag{1.24}$$

Ainsi, un égaliseur à espacement simple devrait pouvoir couvrir au moins la longueur de l'écho pour compenser le premier terme de la division polynomiale. De plus, afin de réduire davantage l'erreur quadratique moyenne, l'équation 1.24 montre qu'il faut doubler la longueur de l'égaliseur pour couvrir le second terme de la longue division. Sachant qu'à certains instants, le trajet secondaire peut devenir le trajet principal, il faut prévoir

Rapport-Gratuit.com

une longueur d'égaliseur capable de s'étaler sur au moins $2Es$ symboles. Par l'équation 1.18, nous avons montré que la dispersion des symboles engendrée par le canal est fonction du taux de symboles. Si l'égaliseur fonctionne à espacement fractionnaire, la dispersion de symbole relative sera alors fonction de l'espacement fractionnaire. Ainsi, un égaliseur fractionné à $T_s/2$ requiert deux fois plus de coefficients qu'un égaliseur à espacement simple et donc devrait couvrir $4Es$ symboles. Cependant, par principe d'interpolation, sa résolution sera deux fois plus grande, (Proakis & Manolakis, 1996).

1.9 Conclusion

Dans ce chapitre, les grandes lignes décrivant un système de communication numérique M-QAM ont été présentées. Nous avons porté une attention particulière au traitement en bande de base puisque le sujet de ce travail se fonde sur les équations inhérentes au traitement numérique dans cette bande. Les principes intrinsèques à l'égalisation ont été élaborés plus en détail dans le but de permettre au lecteur de se familiariser avec les concepts de bases de ce module.

Par le biais d'une analyse empirique, nous avons dégagé que le nombre de coefficients d'égaliseur requis est entre autre fonction de l'étalement des délais. De plus, nous avons montré que l'étalement des délais est fonction du taux de symboles. Donc, la quantité de symboles dispersés par le canal vue de l'égaliseur est fonction du taux d'espacement fractionnaire de ce dernier ainsi que de la fréquence des symboles transmis. Par surcroît, pour un étalement de délai donné, la longueur de l'égaliseur doit être ajustée de manière intimement liée au taux de transmission des symboles.

Les équations 1.22 et 1.24 nous ont également montrées que plus l'égaliseur est long, plus l'IES et par le fait même, l'EQM en sont réduites. Sachant que la probabilité d'erreur d'un système M-QAM augmente en fonction de M, nous pouvons également avancer que plus M est grand, plus on a avantage à réduire l'IES et donc à augmenter le nombre de coefficients de l'égaliseur.

CHAPITRE 2

ÉGALISATION ADAPTATIVE

2.1 Introduction

Ce chapitre présente les principes de base de l'égalisation adaptative dans le domaine temporel ainsi que dans le domaine fréquentiel. Nous exposons les équations inhérentes à l'algorithme LMS ainsi que celles des algorithmes autodidactes CMA et MMA. Puis, nous portons une attention particulière sur les étapes qui servent à convertir l' algorithme LMS dans le domaine fréquentiel.

Puisque nous nous intéressons principalement à l'implémentation matérielle d'un égaliseur dans le domaine fréquentiel, nous développons les équations de complexité de calcul des algorithmes à l'étude. Pour aider à la conception architecturale de l'égaliseur fréquentiel, on étudie également les propriétés de convergence de ces derniers.

L'égalisation adaptative est une application quasi directe au domaine du filtrage adaptatif. En effet, tel qu'indiqué sur la figure 10, quatre signaux sont essentiels pour accomplir un filtre adaptatif. Soit le signal d'entrée, le signal de référence, la sortie et l'erreur. De la même manière, un égaliseur prend les symboles complexes reçus comme entrée. Puis, la décision sur les symboles ainsi que la séquence d'entraînement sont les homologues du signal désiré. Pour compléter la similitude, le symbole estimé est la sortie du filtre adaptatif. Suivant cette réflexion, ce chapitre débute avec les fondements du filtrage adaptatif.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

2.2 Filtre de Wiener

Le filtre de Wiener constitue la base de la théorie du filtrage adaptatif, (Bose, 2004). C'est le filtre optimal que la plupart des algorithmes de filtrage adaptatifs tentent

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

d'atteindre. Le développement mathématique est équivalent pour les signaux réels et complexes. Pour alléger la présentation, nous allons donc exposer le développement pour un signal réel.

Le signal à l'entrée, soit l'échantillon reçu $r(n)$, est filtré de manière à ce que la sortie, soit l'estimé du symbole transmis $\hat{s}(n)$, corresponde au signal désiré $d(n)$. Le but est de trouver les coefficients dont la réponse à l'impulsion fait en sorte que l'erreur quadratique moyenne (EQM) soit minimale. La sortie du filtre de Wiener s'écrit:

$$\hat{s}(n)=\boldsymbol{w}^T\boldsymbol{r}(n) \quad [2] \tag{2.1}$$

où $\boldsymbol{w}^T=[w_0\, w_1 \ldots w_{N-1}]$ et $\boldsymbol{r(n)}=[r(n)r(n-1)\ldots r(n-N-1)]$. La figure 10 illustre le filtre de Wiener sous sa forme linéaire transversale de type FIR.

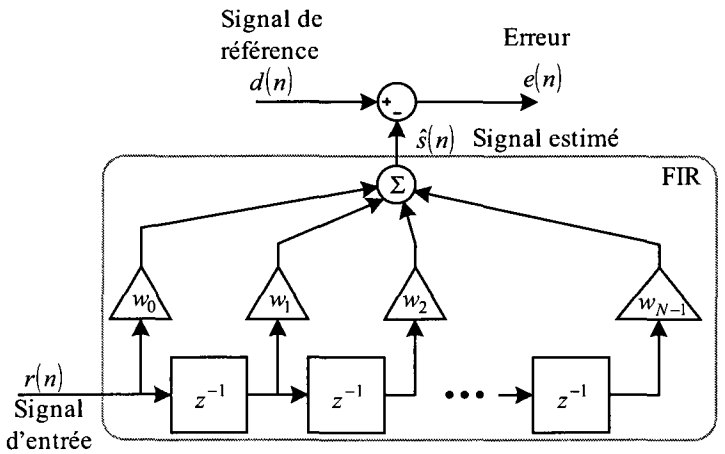


Figure 10 Filtre de Wiener

Les z^{-1} indiquent un délai d'échantillonnage. Dans une réalisation matérielle, on le conçoit avec une bascule dont le signal d'activation (*ce*) est stimulé au taux de symbole ou à une fraction du taux de symbole que l'on appelle espacement fractionnaire (FSE).

2 Dans ce mémoire, les symboles mathématiques en caractères gras représentent des vecteurs.

La fonction de coût, ou le critère de l'EQM, consiste à minimiser l'erreur quadratique entre le signal désiré $d(n)$ et le symbole estimé $\hat{s}(n)$. Cette fonction s'écrit:

$$EQM = J(w) = E\left\{[d(n) - \hat{s}(n)]^2\right\}. \tag{2.2}$$

Pour minimiser $J(\boldsymbol{w})$, on doit chercher le vecteur $\boldsymbol{w}$ qui annule le gradient de la fonction de coût:

$$\nabla J(\boldsymbol{w}) = \frac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}} = 0. \tag{2.3}$$

La solution de 2.3 est appelée le filtre de Wiener et s'écrit:

$$\boldsymbol{w}_{opt} = \boldsymbol{R}^{-1}\, \boldsymbol{p} \tag{2.4}$$

où nous avons utilisé l'indice *opt* pour indiquer les coefficients optimaux du filtre de Wiener. $\boldsymbol{R}$ représente la matrice d'auto corrélation du signal d'entrée $r(n)$.

$$\boldsymbol{R} = \boldsymbol{E}\left\{\boldsymbol{r}(n)\boldsymbol{r}^T(n)\right\} \tag{2.5}$$

$\boldsymbol{p}$ est défini comme la matrice d'inter corrélation entre le signal reçu $r(n)$ et le signal désiré $d(n)$.

$$\boldsymbol{p} = \boldsymbol{E}\left\{d(n)\boldsymbol{r}(n)\right\} \tag{2.6}$$

La résolution de l'équation 2.4 implique une inversion de matrice. Cette opération est très coûteuse en terme de complexité de calcul. Pour cette raison, la plupart du temps, on utilise un algorithme itératif pour solutionner le filtre de Wiener. L'algorithme LMS est parmi les algorithmes les plus utilisés pour obtenir la solution des coefficients du filtre de Wiener.

2.3 L'algorithme LMS

L'algorithme LMS est une version approximée de la méthode *steepest descent*, (Bose, 2004). Selon l'algorithme *steepest descent*, les coefficients se calculent comme suit:

$$\boldsymbol{w}(n+1)=\boldsymbol{w}(n)+\mu[\boldsymbol{p}-\boldsymbol{R}\boldsymbol{w}(n)] \tag{2.7}$$

où μ est une constante appelée *pas d'adaptation* et est introduit pour contrôler le taux de convergence des coefficients de l'égaliseur. En égalisation, la matrice d'auto corrélation du signal d'entrée ainsi que la matrice d'inter corrélation $\boldsymbol{p}$ ne peuvent qu'être estimées. Ainsi, en simplifiant les équations 2.5 et 2.6 de cette manière:

$$\boldsymbol{R}=\boldsymbol{E}\left\{\boldsymbol{r}(n)\boldsymbol{r}^T(n)\right\}\simeq\boldsymbol{r}(n)\boldsymbol{r}^T(n)\,, \tag{2.8}$$

$$\boldsymbol{p}=\boldsymbol{E}\left\{\boldsymbol{d}(n)\boldsymbol{r}(n)\right\}\simeq d(n)\boldsymbol{r}(n)\,. \tag{2.9}$$

Puis, en substituant les équations 2.8 et 2.9 dans l'algorithme de *steepest descent*, on obtient l'équation récursive suivante:

$$\boldsymbol{w}(n+1)=\boldsymbol{w}(n)+\mu[\boldsymbol{r}(n)e(n)]\,. \tag{2.10}$$

L'équation 2.10 constitue le fondement de base de l'algorithme LMS. Les étapes de calcul de l'algorithme LMS sont rapportés au tableau II.

Tableau II

Sommaire des étapes de calcul de l'algorithme LMS

1.	Filtrage	$\hat{s}(n)=\boldsymbol{w}^T\boldsymbol{r}(n)$
2.	Estimation de l'erreur	$e(n)=d(n)-\hat{s}(n)$
3.	Adaptation des coefficients	$\boldsymbol{w}(n+1)=\boldsymbol{w}(n)+\mu[\boldsymbol{r}(n)e(n)]$

2.3.1 Convergence de l'algorithme LMS

Il a été démontré, (Widrow & Sterns, 1985), que la disposition des valeurs propres de la matrice d'auto corrélation $\boldsymbol{R}$ déterminent le taux de convergence des algorithmes dont le gradient est basé sur la méthode *steepest descent.* Ainsi, l'algorithme LMS convergera plus ou moins rapidement selon l'étalement des valeurs propres λ_i de la matrice $\boldsymbol{R}$. L'équation 2.10 peut être récrite sous la forme suivante, (Farhang-Boroujeny, 1998):

$$\boldsymbol{w}(n+1)=\boldsymbol{w}(n)+\sum_{i=0}^{N-1}\alpha_i\left(1-2\mu\lambda_i\right)^2 q_i \tag{2.11}$$

où q_i représente les vecteurs propres et α_i est une constante. Cette expression indique clairement que la convergence des coefficients de l'égaliseur est gouvernée par la somme de N termes exponentiels chacun étant fonction de sa valeur propre λ_i. Chaque terme contrôle indépendamment la direction des vecteurs propres q_i. Ainsi, la convergence est rapide quand le terme $\left|\left(1-2\mu\lambda_i\right)\right|$ est faible.

Pour renchérir ce que nous venons de mentionner dans les deux derniers paragraphes, on note que le taux de convergence de l'algorithme LMS peut être évalué à partir de l'étalement des valeurs propres $\lambda_{max}/\lambda_{min}$. On montre dans (Farhang-Boroujeny, 1998) que les valeurs λ_{max} et λ_{min} peuvent être approximées asymptotiquement par les puissances maximale et minimale du signal à l'entrée de l'égaliseur.

Ainsi, pour assurer la stabilité de l'algorithme LMS, en manipulant l'équation 2.11, il est possible de prouver que le *pas d'adaptation* μ doit être borné par

$$0<\mu<1/\lambda_{max}\leq tr\{\boldsymbol{R}\} \tag{2.12}$$

où $tr\{\cdot\}$ dénote la somme des éléments de la diagonale de $\boldsymbol{R}$. Par surcroît, les éléments de la diagonale de la matrice d'auto corrélation représentent la puissance des

symboles aux instants n jusqu'à $n-N+1$. Par exemple, pour un signal d'entrée de puissance moyenne unitaire, μ devrait être sélectionné entre les bornes

$$0<\mu<1/N. \tag{2.13}$$

Plus μ est choisi près de la borne supérieure de 2.12, plus l'algorithme convergera rapidement et vice versa. Cependant, il faut être prudent car lorsqu'on pose un *pas d'adaptation* trop élevé, il y a risque d'instabilité. De plus, l'écart entre les coefficients optimums et ceux trouvés par l'algorithme LMS est dépendant du choix de μ selon la relation

$$M \simeq \mu\, tr\{\boldsymbol{R}\} \tag{2.14}$$

M est appelé en anglais le *misadjustment* et fait référence au rapport

$$M=\frac{EQM-EQMM}{EQMM}. \tag{2.15}$$

Or, si le *misadjustment* est élevé, l'EQM l'est aussi et entraîne une réduction du SNR. Suivant ce raisonnement, on peut alors dire que le *pas d'adaptation* doit être choisi plus faible pour des constellations plus denses. En effet, tel que nous l'avons montré au chapitre 1, le rapport signal sur bruit doit être ajusté suivant la courbe de SER pour des symboles M-QAM.

Notons qu'en communication sans fil, le comportement du canal de transmission n'est pas connu à priori. Ainsi, les bornes montrées ci haut peuvent servir à titre indicatif mais en général, on a recourt à des modèles de simulations qui s'appuient sur des données statistiques de l'environnement dans lequel le modem sera intégré pour déterminer μ.

2.3.2 Complexité de l'algorithme LMS

On s'intéresse à l'algorithme LMS en raison de sa simplicité mais aussi à cause de sa faible complexité de calcul. Pour décrire la complexité de calcul d'un algorithme, on examine le nombre de multiplications réelles requises mais aussi d'autres facteurs tels que le nombre d'additions et l'espace mémoire.

Il existe une multitude de stratégies d'implémentations. Certaines implémentations sont optimisées pour la vitesse, d'autres pour l'espace matérielle en utilisant des techniques de réutilisations de ressources. Il y a aussi des réalisations qui combinent l'optimisation des deux derniers critères. Ainsi, il est difficile voire même impossible de comparer justement deux implémentations. Cependant, le nombre d'opérations requises pour accomplir l'adaptation LMS des coefficients est le même pour chaque égaliseur basé sur cet algorithme. De plus, dans toutes les réalisations matérielles, le nombre de multiplicateurs disponibles est un critère primordial. Ainsi, dans ce chapitre, nous allons effectuer l'analyse de complexité de calcul des algorithmes à l'étude en énonçant seulement que le nombre de multiplications réelles requises par itérations.

Ainsi, en reprenant les équations du tableau II, on observe qu'à chaque itération, l'étape de filtrage requiert N multiplications complexes. En effet, on considère le fait que les symboles M-QAM sont complexes et que les coefficients de l'égaliseur le sont eux aussi. Rappelons que la multiplication de deux nombres complexes requiert quatre multiplications réelles. L' équation 2.15 démontre cette observation.

$$(a+jb)\cdot(d+jc)=(ad-bc)+j(ac+bd) \tag{2.16}$$

Donc, l'équation 1 du tableau II demande $4N$ multiplications réelles. Le calcul de l'erreur est obtenu par une addition complexe mais ne nécessite pas de multiplication. Suivant ce raisonnement, pour terminer l'analyse de complexité de l'égaliseur LMS, on doit ajouter les $4N$+2 multiplications pour accomplir l'équation 3 du tableau II (le 2 provient de la multiplication du *pas d'adaptation*, un scalaire, au résultat de l'inter corrélation

$[\boldsymbol{r}(\boldsymbol{n})e(n)]$). Le tableau III rapporte les résultats de cette analyse en combinant la complexité de l'équation de filtrage et avec celle de l'adaptation des coefficients.

Tableau III

Sommaire des multiplications requises dans l'algorithme LMS

Filtrage	$4N$
Adaptation des coefficients	$4N+2$
Total	$8N+2$

2.4 Les algorithmes CMA et MMA

La plupart du temps, pour que l'algorithme LMS converge, on doit inclure une séquence connue dans le protocole de communication. Cette séquence d'entraînement s'accompagne de deux inconvénients notables. Dans un premier temps, l'ajout d'une séquence d'entraînement amoindrit l'efficacité spectrale du lien de communication. De plus, afin de retracer la séquence d'entraînement, on doit mettre en oeuvre un circuit non trivial de recouvrement de la trame qui consiste essentiellement en une corrélation glissante semblable à ce qu'on fait avec le CDMA, (Proakis, 2001).

Pour éviter les embarras engendrés par l'ajout de la séquence d'entraînement, on a recourt à des algorithmes dits autodidactes. Ces types d'algorithmes sont basés sur des propriétés statistiques d'ordre supérieur dont sont agrémentées les constellations QAM et PSK. Dans ce travail, nous aborderons l'algorithme CMA (constant modulus algorithm) de (Godard, 1980) ainsi que le CMA modifié connue sous l'acronyme MMA (multilmodulus algorithm), (Yang, Werner & Dumont, 2002).

En somme, les étapes d'adaptations des coefficients de l'algorithme LMS demeurent les mêmes. Cependant, on utilise une fonction de coût différente. Pour sa part, l'algorithme CMA utilise la fonction de coût 2.14. Celle-ci minimise la dispersion des symboles autour d'un cercle tel que montré sur la figure 11.

$$J(w)_{CMA} = E\left\{\left[\left|\hat{s}(n)^2\right| - R_{CMA}^2\right]^2\right\} \tag{2.17}$$

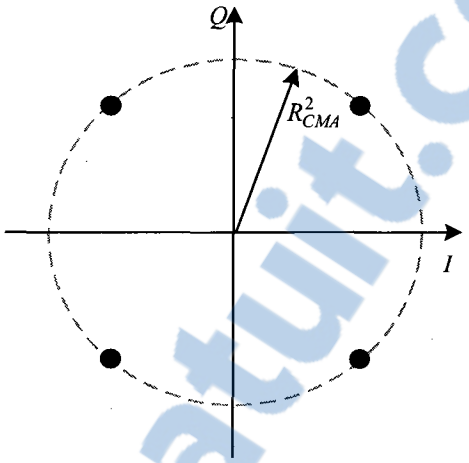


Figure 11 Rayon de convergence de l'algorithme CMA

Où le rayon R_{CMA}^2 est constant et correspond à

$$R_{CMA}^2 = \frac{E\left\{|s(n)|^4\right\}}{E\left\{|s(n)|^2\right\}}. \tag{2.18}$$

Par exemple, pour une constellation 4-QAM normalisée dont les symboles seraient distribués de manière équiprobable, on fixerait R_{CMA}^2 à l'unité.

Un des problèmes que l'on rencontre avec l'algorithme CMA est qu'il ne permet que de corriger le module des symboles complexes. Ainsi, il arrive souvent que les coefficients de l'égaliseur convergent avec un déphasage. Ce déphasage peut être nocif d'un point de vue performance. Pour illustrer l'impact de cette problématique, la figure 12 illustre une situation où le bloc de décision est induit en erreur par le déphasage des coefficients pour une constellation 16-QAM. On remarque que pour certaines régions de décisions (en trait pâle), la décision est erronée ou ambiguë.

Rapport-Gratuit.com

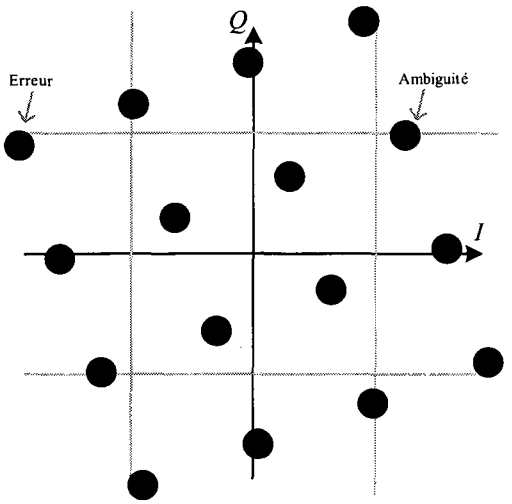


Figure 12 Problèmes de convergence avec l'algorithme CMA

L'algorithme MMA permet d'améliorer cette ambiguïté de phase. Pour y arriver, on sépare la fonction de coût en deux parties, réelle et complexe. La fonction de coût s'écrit:

$$J(w)_{MMA}=E\left\{\left[(\Re\{\hat{s}(n)\})^2-R^2_{MMA_1}\right]^2\right\}+jE\left\{\left[(\Im\{\hat{s}(n)\})^2-R^2_{MMA_2}\right]^2\right\} \tag{2.19}$$

où les rayons $R^2_{MMA_1}$ et $R^2_{MMA_2}$ sont constants puis correspondent à

$$R^2_{MMA_1}=\frac{E\left\{|\Re\{s(n)\}|^4\right\}}{E\left\{|\Re\{s(n)\}|^2\right\}}\ ,\ R^2_{MMA_2}=\frac{E\left\{|\Im\{s(n)\}|^4\right\}}{E\left\{|\Im\{s(n)\}|^2\right\}} \tag{2.20}$$

Cette modification fait en sorte que la phase et l'amplitude peuvent être ajustées de manière autodidacte. Cependant, on peut observer que si la distorsion de phase engendrée par le canal de transmission est supérieure à 45 degrés, il peut à nouveau y avoir ambiguïté de phase après convergence.

2.4.1 Convergence des algorithmes CMA et MMA

Le calcul de l'adaptation des coefficients des algorithmes CMA et MMA a la même apparence que celle d'un égaliseur LMS. La différence se situe au niveau du terme de l'erreur $e(n)$ du gradient stochastique de l'équation 2.10. Ainsi, pour l'algorithme CMA, l'erreur s'écrit:

$$e_{CMA}(n)=\hat{s}(n)\left(|\hat{s}(n)|^2-R_{CMA}^2\right) \tag{2.21}$$

et pour l'algorithme MMA, on a:

$$e_{MMA}(n)=\left[\Re\{s(n)\}\left((\Re\{\hat{s}(n)\})^2-R_{MMA_1}^2\right)\right]+j\left[\Im\{s(n)\}\left((\Im\{\hat{s}(n)\})^2-R_{MMA_2}^2\right)\right]. \tag{2.22}$$

Donc, puisque les étapes de filtrage sont semblables à la méthode LMS, on peut stipuler que ces algorithmes autodidactes sont dépendants des valeurs propres de la matrice d'auto corrélation $\boldsymbol{R}$. On peut alors choisir un *pas d'adaptation* en suivant sensiblement la même procédure qu'avec un égaliseur LMS.

On montre dans (Godard, 1980) que pour augmenter les probabilités de convergence vers la bonne solution, de même que la vitesse de d'adaptation, on a avantage à initialiser le gain du coefficient de référence à une valeur de l'ordre de la puissance moyenne des symboles de la constellation. Par exemple, pour une constellation 4-QAM normalisée, on initialiserait les coefficients comme suit:

$$\boldsymbol{w}(0)=[\underbrace{0\cdots 0}_{N/2-1}1\ \underbrace{0\cdots 0}_{N/2}]. \tag{2.23}$$

2.4.2 Complexité de calcul des algorithmes CMA et MMA

La complexité de calcul des algorithmes CMA et MMA est à toute fin pratique identique à celle de l'algorithme LMS. La différence réside dans le calcul de l'erreur. Dans les deux cas, on doit ajouter 4 multiplications. On constate toutefois que $e_{CMA}(n)$ et

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

$e_{MMA}(n)$ ne sont variables qu'en fonction de $\hat{s}(n)$ puisque R_{CMA} et R_{MMA} sont des constantes dépendantes de la constellation donnée. C'est pourquoi on choisit souvent d'utiliser une table de référence pour générer ces erreurs dans le processus d'égalisation.

2.5 L'algorithme BLMS

Les algorithmes présentés jusqu'ici utilisent la valeur instantanée du gradient pour l'adaptation des coefficients. L'algorithme BLMS quant à lui, utilise une moyenne temporelle du gradient sur L échantillons. Ainsi, les signaux $d(n)$, $e(n)$ et $\hat{s}(n)$ sont transformés en vecteurs (ou blocs) de longueur L. D'où le qualitatif BLMS qui signifie *block least mean sqare*. La sortie d'un égaliseur de type BLMS est donnée par:

$$\hat{\boldsymbol{s}}(k+i)=\boldsymbol{w}^T\boldsymbol{r}(k+i), \qquad i=0\ldots L-1\,. \tag{2.24}$$

L'étape d'adaptation des coefficients s'écrit:

$$\boldsymbol{w}(k+1)=\boldsymbol{w}(k)+\frac{\mu_{LMS}}{L}\sum_{i=0}^{L-1}[\boldsymbol{r}(kL+i)\boldsymbol{e}^*(kL+i)] \tag{2.25}$$

où les indices n et k sont reliés par

$$n=kL+i\,. \tag{2.26}$$

À partir des équations 2.7, 2.1 et 2.2, on remarque que si on pose le bloc L = 1, on obtient exactement l'algorithme LMS.

2.5.1 Convergence de l'algorithme BLMS

Dans (Clark, Mika & Parker, 1981), on montre que les performances de l'algorithme BLMS sont équivalentes à celles de l'algorithme LMS. Le taux de convergence ainsi que le *misadjustment* sont également gouvernés par les mêmes équations tirées des valeurs propres de la matrice d'auto corrélation $\boldsymbol{R}$. Cependant, on note un avantage basé sur le

fait que les coefficients du filtre ne sont ajustés qu'une seule fois à tous les blocs de L échantillons. On obtient alors une courbe de convergence plus lisse que celle obtenue avec l'algorithme LMS. Toutefois, on observe que plus L est choisit grand, plus la cohérence du gradient avec le canal est amoindrie. La longueur du bloc optimale est $L=N$, où nous rappelons que N est le nombre de coefficients de l'égaliseur.

2.5.2 Complexité de calcul de l'algorithme BLMS

La complexité de l'algorithme BLMS est la même que celle de l'algorithme LMS pour un bloc de N échantillons. Donc, le nombre total de multiplications requises est de $N(8N+2)$ par blocs de N échantillons.

2.6 L'algorithme FBLMS

L'algorithme FBLMS a été présenté pour la première fois par (Ferrara, 1980). Fondamentalement, la structure de cet algorithme n'est rien d'autre qu'une adaptation dans le domaine fréquentiel de la méthode BLMS. Le principal objectif de cette traduction est de réduire la complexité de calcul. La technique consiste à tirer profit des algorithmes performants de transformée de Fourier (FFT) ainsi que du fait qu'une multiplication point à point en fréquence est équivalente à une convolution dans le domaine du temps.

2.6.1 Convolution dans le domaine fréquentiel

La multiplication de deux séquence de longueur N dans le domaine fréquentiel résulte en une convolution circulaire, (Oppenheim & Schafer, 1975). Or, l'étape de filtrage de l'algorithme LMS correspond à une convolution linéaire. Pour pallier à la circularité d'une convolution fréquentielle, on utilise soit la méthode *overlap-save* ou la méthode *overlap-add*, (Proakis & Manolakis). Pour l'implémentation de l'algorithme FBLMS, la méthode par chevauchement et sauvegarde (*overlap-save*) est la plus répandue car sa

réalisation s'avère moins complexe en terme de synchronisation et de ressources de mémoire que la méthode de chevauchement et addition (*overlap-add*), (Shynk, 1992).

2.6.1.1 Méthode *overlap-save*

La longueur du bloc du signal d'entrée est $N'=N+L-1$ et donc les transformées de Fourier rapides (FFT/IFFT) sont de longueur N'. Chaque bloc d'entrée est construit avec les L-1 échantillons de l'ancien bloc concaténés des N nouveaux symboles. Une FFT doit être calculée à chaque bloc de symboles. On définit le vecteur fréquentiel des symboles d'entrés comme suit:

$$\boldsymbol{r}_F(\boldsymbol{k})=FFT\left\{\left[\underbrace{r(kN-L+1),\ldots,r(kN-1)}_{L-1\text{ anciens symboles}},\underbrace{r(kN),\ldots r(kL-1)}_{N\text{ nouveaux symboles}}\right]\right\}. \tag{2.27}$$

Pour être cohérent, le vecteur des coefficients doit être convertit en fréquence de la manière suivante:

$$\boldsymbol{w}_F(\boldsymbol{k})=FFT\left\{\left[\underbrace{w(kN-L+1),\ldots,w(kN-1)}_{L-1},\underbrace{0,\ldots,0}_{N}\right]\right\}. \tag{2.28}$$

Le résultat de la convolution fréquentielle des deux vecteurs $\boldsymbol{r}_F(k)$ et $\boldsymbol{w}_F(k)$ est alors reconverti dans le domaine du temps à l'aide d'une IFFT. La réponse de la IFFT sur N' points donne:

$$\tilde{\boldsymbol{s}}(k)=[\tilde{s}(0),\tilde{s}(1),\ldots,\tilde{s}(L-1),\hat{s}(L)\ldots,\hat{s}(N')]. \tag{2.29}$$

On note que les L-1 premiers échantillons du vecteur $\tilde{s}(n)$ sont corrompus par les alias spectrales. Ils doivent donc être éliminés. Les derniers N points de $\tilde{s}(n)$ correspondent de manière identique au résultat d'une convolution linaire dans le domaine du temps. La figure 13 reproduit les étapes de filtrages dans le domaine fréquentiel pour le cas où N'=2N.

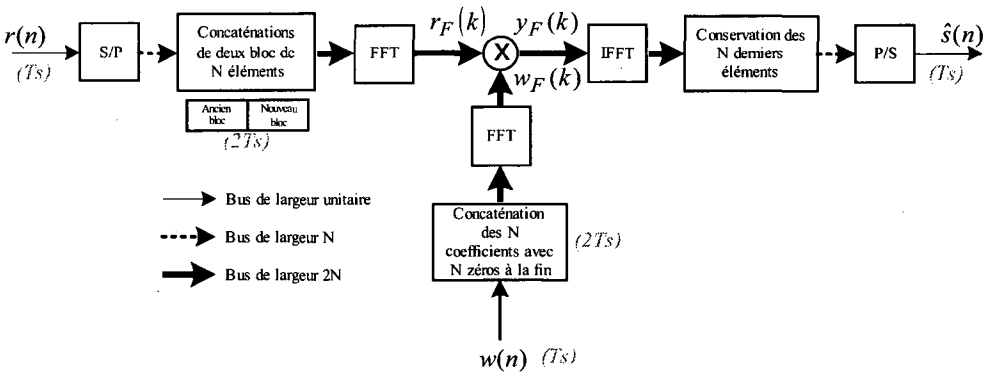


Figure 13 Filtrage linéaire en fréquence par la méthode *overlap-save* avec $N' = 2N$

Sur la figure 13, on peut observer que la fréquence de fonctionnement d'un filtre fréquentiel dont le vecteur d'entrée est de longueur $2N$ doit être fixée deux fois plus élevée que le taux d'arrivée des échantillons.

2.6.2 Adaptation des coefficients FBLMS

Pour effectuer l'adaptation fréquentielle des coefficients, on doit convertir le vecteur d'erreur dans une forme fréquentielle comme suit:

$$\boldsymbol{e}_F(\boldsymbol{k}) = FFT\left\{\left[\underbrace{0,\ldots,0}_{L-1}, \underbrace{w(kN-L+1),\ldots,w(kN-1)}_{N}\right]\right\}. \tag{2.30}$$

L'équation récursive des coefficients dans le domaine fréquentiel s'écrit:

$$\boldsymbol{w}_F(k+1) = \boldsymbol{w}_F(k) + \mu_{LMS}\left[\boldsymbol{r}_F^*(k) \otimes \boldsymbol{e}(k)\right]_C \tag{2.31}$$

où $\otimes$ dénote une multiplication de vecteur élément par élément et $[\cdot]_C$ désigne l'étape de contrainte du gradient. En effet, pour être cohérent avec le processus de filtrage linéaire dans le domaine fréquentiel montré plus haut, on doit transformer le vecteur $\boldsymbol{r}_F^*(k) \otimes \boldsymbol{e}(k)$ dans le domaine du temps puis remplacer les $L-1$ derniers éléments par autant de zéros. Puis, on doit ensuite reconvertir ce vecteur en fréquence. L'équation 2.16 résume ce calcul:

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

$$\left[\boldsymbol{r}_F^*(k)\otimes\boldsymbol{e}(k)\right]_C = FFT\left\{\begin{matrix} N \text{ premiers éléments de } IFFT\left\{r_F^*(k)\otimes\boldsymbol{e}(k)\right\} \\ \mathbf{0} \end{matrix}\right\} \tag{2.32}$$

où le $\mathbf{0}$ désigne un vecteur d'éléments nuls de longueur $L-1$. La figure 14 illustre le schéma bloc d'un égaliseur de type FBLMS avec $N'=2N$.

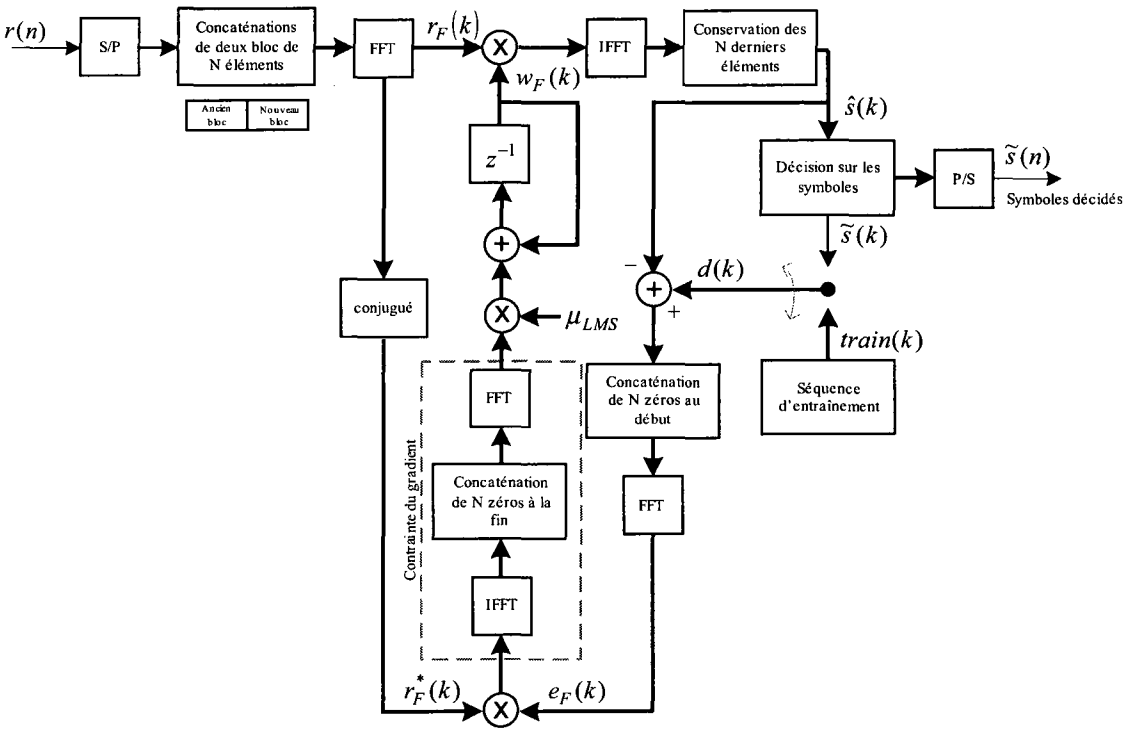


Figure 14 Schéma bloc de l'égaliseur FBLMS avec $N' = 2N$

Le tableau IV résume les étapes de calcul de l'algorithme FBLMS:

Tableau IV

Sommaire des étapes de calcul de l'algorithme FBLMS

1.	**Transformée en fréquence des données**	$\boldsymbol{r}_F(k)=FFT\left\{\left[\underbrace{r(kN-L+1),\ldots,r(kN-1)}_{L-1 \text{ anciens symboles}},\underbrace{r(kN),\ldots r(kL-1)}_{N \text{ nouveaux symboles}}\right]\right\}$
2.	**Filtrage**	$\boldsymbol{y}_F(k)=\boldsymbol{w}_F(k)\otimes\boldsymbol{r}_F(k)$

3.	**Calcul de la sortie temporelle**	$\hat{s}(k)=N$ derniers éléments de $IFFT\left\{\boldsymbol{s}_F(k)\right\}$
4.	**Calcul de l'erreur en fréquence**	$\boldsymbol{e}_F(k)=FFT\left\{\begin{matrix}\mathbf{0}\\ \boldsymbol{d}(k)-\hat{\boldsymbol{s}}(k)\end{matrix}\right\}$
5.	**Inter corrélation**	$\left\{r_F^*(k)\otimes\boldsymbol{e}(k)\right\}$
6.	**Contrainte du gradient**	$\left[\boldsymbol{r}_F{}^*(k)\otimes\boldsymbol{e}(k)\right]_C=FFT\left\{\begin{matrix}N \text{ premiers éléments de } IFFT\left\{r_F^*(k)\otimes\boldsymbol{e}(k)\right\}\\ \mathbf{0}\end{matrix}\right\}$
7.	**Adaptation des coefficients**	$\boldsymbol{w}_F(k+1)=\boldsymbol{w}_F(k)+\mu_{LMS}\left[\boldsymbol{r}_F{}^*(k)\otimes\boldsymbol{e}(k)\right]_C$

2.6.3 Convergence de l'algorithme FBLMS

L'algorithme FBLMS n'est rien d'autre qu'une implémentation fréquentielle de l'algorithme BLMS et il a été démontré, (Boroujeny, 1998), que cet algorithme se comporte de manière très similaire à l'algorithme LMS. Ainsi, sa convergence est dépendante de la disparité des valeurs propres de la matrice d'auto corrélation $\boldsymbol{R}$. Qui elle, est intimement liée à la variation de puissance des symboles reçus.

2.6.3.1 Convergence de l'algorithme FBLMS non contraint

En observant les équations menant à l'adaptation des coefficients, on note qu'il est requis de contraindre le gradient de manière à respecter les étapes de filtrage linéaire dans le domaine fréquentiel. En effet, le vecteur $\boldsymbol{r}_F(k)$ est composé d'un bloc de N anciens échantillons et N nouveaux. Ceci fait en sorte que l'inter corrélation fréquentielle qu'accomplit le gradient est circulaire. Toutefois, plusieurs études ont montrées que dans certaines conditions, l'omission de cette contrainte peut donner des propriétés de convergences similaires voir même meilleures dans certains cas, (Mansour, & Gray,. (1982); Lee, & Un,. (1989); Farhang-Boroujeny, (2000)). Pour bien distinguer les deux

algorithmes, nous allons spécifier UFBLMS lorsque le gradient n'est pas contraint (*unconstrained* FBLMS) et CFBLMS lorsqu'il est contraint (*constrained* FBLMS).

Les chercheurs mentionnés ci haut ont montré que lorsque l'on pose un nombre suffisant de coefficients N pour couvrir les distorsions du canal, les deux algorithmes, UFBLMS et CFBLMS se comportent à toute fin pratique de la même manière. Ainsi, dans cette condition, la circularité ne se fait pas ressentir. Cependant, lorsque la longueur de l'égaliseur n'est pas assez élevée par rapport à la réponse à l'impulsion du canal, l'algorithme UFBLMS se comporte différemment. On montre qu'il converge tout de même vers la solution optimale de Weiner mais, dépendamment de la sévérité du canal, il peut générer une EQM plus instable et peut prendre plus de temps à converger. Dans son article, Mansour a soulevé un point intéressant. Il a montré que dans certaines conditions de canal, l'EQM de l'égaliseur UFBLMS peut parfois être inférieure à celle de l'égaliseur CFBLMS. Surtout dans le cas où la longueur de l'égaliseur ne couvre pas tout le canal. En effet, ne pas contraindre le gradient fait en sorte que l'égaliseur UFBLMS est muni de deux fois plus de coefficients. Soit un vecteur de coefficient semblable à ce qu'on obtient avec l'algorithme CFBLMS et un autre vecteur de coefficients issus de la corrélation circulaire. L'équation d'adaptation des coefficients s'écrit:

$$ {}_{u}\boldsymbol{w}_{F}(k+1) = {}_{u}\boldsymbol{w}_{F}(k) + \mu_{LMS}\boldsymbol{r}_{F}^{*}(k) \otimes \boldsymbol{e}(k) \tag{2.33} $$

où ${}_{u}\boldsymbol{w}_{F}(k)$ représente le vecteur de coefficients non contraints. Dans le domaine du temps, ce vecteur est composé comme suit:

$$ IFFT\left\{{}_{u}\boldsymbol{w}_{F}(k)\right\} = \begin{bmatrix} \boldsymbol{w}(k) \\ {}_{u}\boldsymbol{w}(k) \end{bmatrix} \begin{matrix} \text{longueur } N \\ \text{longueur } N \end{matrix}. \tag{2.34} $$

On constate que le deuxième terme (vecteur) de l'équation 2.17 n'est pas nul tel qu'il devrait l'être (voir l'équation 2.13). C'est l'effet de la corrélation circulaire. Tel que mentionné plus haut, lorsque le filtre est assez long, ce terme devient nul après convergence. Cependant, si le filtre ne couvre pas tout le canal, le terme ${}_{u}\boldsymbol{w}(k)$ n'est

pas nul même après convergence. Dépendamment des conditions de canal, ce terme peut aider ou nuire à la convergence et ainsi permettre de réduire ou d'augmenter l'EQM. Au chapitre 3, nous verrons que pour un canal de type SUI, les performances de l'égaliseur UFBLMS sont approximativement les mêmes que l'égaliseur CFBLMS. On notera également que l'EQM est meilleure (plus faible) avec l'algorithme UFBLMS lorsque le filtre n'est pas assez long pour couvrir tous les échos.

2.6.4 Complexité de calcul de l'algorithme FBLMS

Jusqu'à présent, notre analyse nous a montrée que l'algorithme FBLMS est équivalent à l'algorithme LMS. On remarque cependant que son implémentation requière plus de manipulations. Son attrait provient de sa complexité de calcul qui croît selon $O\{\log_2(N)\}$ et ce, en tirant profit des algorithmes performants de calcul de transformée de Fourrier tandis que son équivalent temporel (BLMS) croît de manière $O\{N^2\}$.

2.6.4.1 Complexité de calcul d'une FFT/IFFT

Tel que nous l'avons mentionné plus haut, la convolution temporelle correspond à une multiplication point à point dans le domaine fréquentiel. Il est alors intéressant d'effectuer des opérations de filtrage dans le domaine fréquentiel pour réduire la complexité de calcul. Le passage temps-fréquence et fréquence-temps peut être accompli à l'aide d'une transformée de Fourier directe ou inverse. Le passage temps-fréquence s'écrit de façon discrète comme suit:

$$X(k)=\sum_{n=0}^{N-1} x(n)W_N^{kn} \qquad 0\leq k\leq N-1 \,. \tag{2.35}$$

Avec le facteur de phase W_N qui correspond à:

$$W_N=e^{-j2\pi l/N} \tag{2.36}$$

Rapport-Gratuit.com

et pour le passage du domaine fréquentiel au domaine du temps, les opérations sont quasi identiques si l'on exclue la mise à l'échelle $1/N$, la transformation discrète s'écrit:

$$x(n)=\frac{1}{N}\sum_{k=0}^{N-1} X(k)W_N^{-nk} \qquad 0\leq n\leq N-1 \,. \tag{2.37}$$

On note que pour effectuer la transformée de Fourier discrète, on a recourt à N multiplications complexes. Ceci correspond à $4N$ multiplications réelles par points. Ainsi, pour obtenir les N points, on doit accomplir $4N^2$ multiplications réelles. Afin d'augmenter les performances, on exploite entres autres, les deux propriétés suivantes des facteurs de phases W_N :

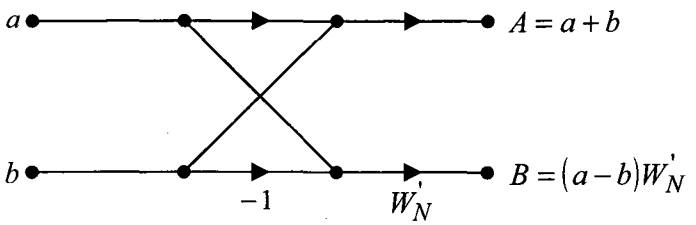
La symétrie:

$$W_N^{k+N/2}=-W_N^k \,. \tag{2.38}$$

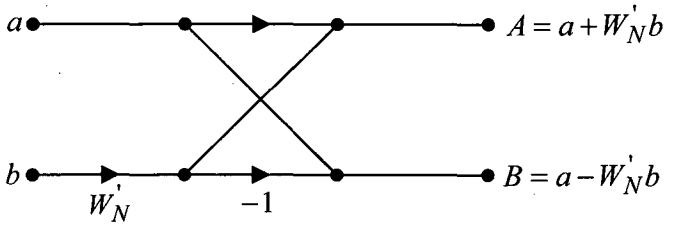
La périodicité:

$$W_N^{k+N}=W_N^k \,. \tag{2.39}$$

De ces propriétés, il ressort plusieurs méthodes rapides pour effectuer la transformation de Fourier, (Proakis & Manolakis, 1996). Lorsque le nombre de points N est de puissance 2 ou de puissance 4, on peut tirer profit des algorithmes bien connus dans la littérature, soit les algorithmes *radix-2* et *radix-4*. L'approche *radix-2* consiste à utiliser un bloc que l'on nomme *papillon*. Ce bloc effectue deux additions et une multiplication. L'opération *papillon* peut être réalisée en décimation temporelle ou en décimation fréquentielle. La figure 15 illustre graphiquement l'opération papillon *radix-2* à décimation en fréquence et à décimation en temps.



Décimation en fréquence



Décimation en temps

Figure 15 Opération papillon *radix-2*

Ainsi, pour accomplir une transformée de Fourier rapide, on combinera $\log_2 N$ opérations papillons pour obtenir $N/2 \log_2 N$ multiplications complexes et $N \log_2 N$ additions complexes. Notons que ces résultats sont obtenus en considérant une implémentation sans réutilisation de ressources. Dans le cadre de ce projet, nous utilisons l'algorithme $R2^2PC$ qui a été présenté par (Grandmaison, 2005). Cet algorithme utilise une méthode basée sur la méthode *radix-4* combinée à des techniques de réutilisations de ressources. L'algorithme *radix-4* permet de réduire davantage la complexité en terme de multiplications de la FFT/IFFT. En effet, par cette méthode, on combine $\log_4 N$ opérations papillons *radix-4* (voir figure 16) pour passer à $3N/8 \log_2 N$ multiplications. Au chapitre 4, nous aborderons plus en détail les motivations qui nous ont poussés à choisir l'architecture $R2^2PC$. Mais à ce stade-ci, nous allons considérer la complexité de calcul d'une FFT/IFFT *radix-2* et *radix-4* sans modifications. Le but de cette section étant de présenter et comparer la complexité de calcul théorique (sans simplifications d'implémentations) des algorithmes FBLMS et BLMS.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

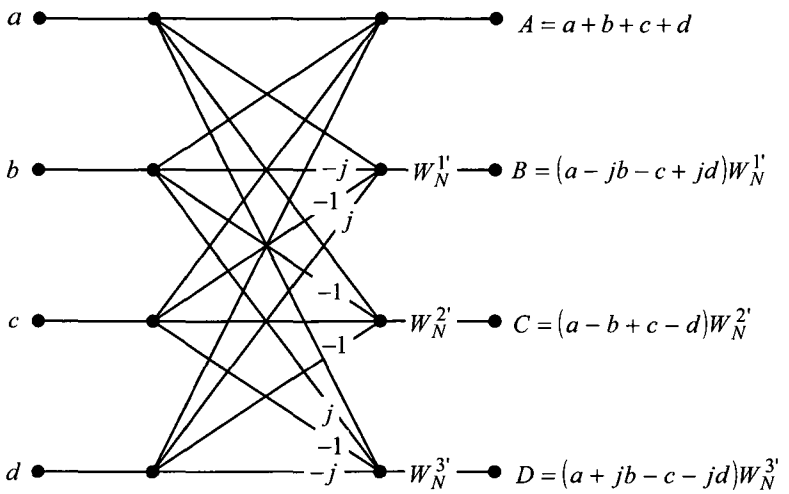


Figure 16 Opération papillon *radix-4* pour une décimation en fréquence.

Maintenant que nous connaissons plus spécifiquement la complexité de calcul d'une transformée de Fourier rapide, voyons plus en détail le nombre de multiplications requises par itérations pour l'algorithme FBLMS. En se rapportant au tableau IV, on remarque qu'une FFT est requise pour la mise en fréquence des symboles d'entrée. Une autre FFT est nécessaire pour transformer le vecteur d'erreur dans le domaine fréquentiel. Finalement, on a recourt à une autre FFT pour ramener en fréquence le gradient contraint. On note également qu'une IFFT est requise pour transformer le résultat de l'opération de filtrage fréquentiel dans le domaine du temps et une autre IFFT doit être accomplie pour réaliser la contrainte du gradient. L'un dans l'autre, on a recourt à cinq FFT/IFFT par bloc de $2N$ échantillons.

Ensuite, l'étape de filtrage requiert $2N$ multiplications complexes par itération. Il en est de même pour l'étape d'inter corrélation $\boldsymbol{r}_F^*(k) \otimes \boldsymbol{e}(k)$. Pour sa part, l'adaptation des coefficients requiert $4N$ multiplications par bloc de $2N$ échantillons. Le tableau V combine toutes les multiplications requises par itération de $2N$ symboles pour chaque étape. Tel qu'il a été mentionné plus haut, nous avons sectionné l'algorithme FBLMS en deux algorithmes distincts, soit l'algorithme CFBLMS et l'algorithme UFBLMS.

Tableau V

Sommaire des multiplications réelles requises dans l'algorithme FBLMS avec la méthode de chevauchement et sauvegarde à 50%

1.	**Mise en fréquence des données**	1 FFT:	***radix-2***	***radix-4***
			$4N\log_2 2N$	$3N\log_2 2N$
2.	**Filtrage**	$2N$ multiplications complexes: $8N$		
3.	**Calcul de la sortie temporelle**	1 IFFT	***radix-2***	***radix-4***
			$4N\log_2 2N$	$3N\log_2 2N$
4.	**Calcul de l'erreur en fréquence**	1 FFT	***radix-2***	***radix-4***
			$4N\log_2 2N$	$3N\log_2 2N$
5.	**Contrainte du gradient**	1 FFT et 1 IFFT:	***radix-2***	***radix-4***
			$8N\log_2 2N$	$6N\log_2 2N$
6.	**Inter corrélation**	$2N$ multiplications complexes: $8N$		
7.	**Adaptation des coefficients**	$2N$ multiplications d'un réel à un complexe: $4N$		

		CFBLMS	**UFBLMS**
Total	***radix-2***	$20N\log_2 2N+20N$	$12N\log_2 2N+20N$
	radix-4	$15N\log_2 2N+20N$	$9N\log_2 2N+20N$

2.7 Complexité de l'algorithme FBLMS versus l'algorithme LMS en bloc

Nous avons toutes les informations pour comparer la complexité en terme de multiplications des algorithmes FBLMS et BLMS. La figure 17 présente la courbe de complexité de ces deux algorithmes. L'algorithme FBLMS a été divisé en deux mode, soit le mode contraint (CFBLMS) et le mode non contraint (UFBLMS).

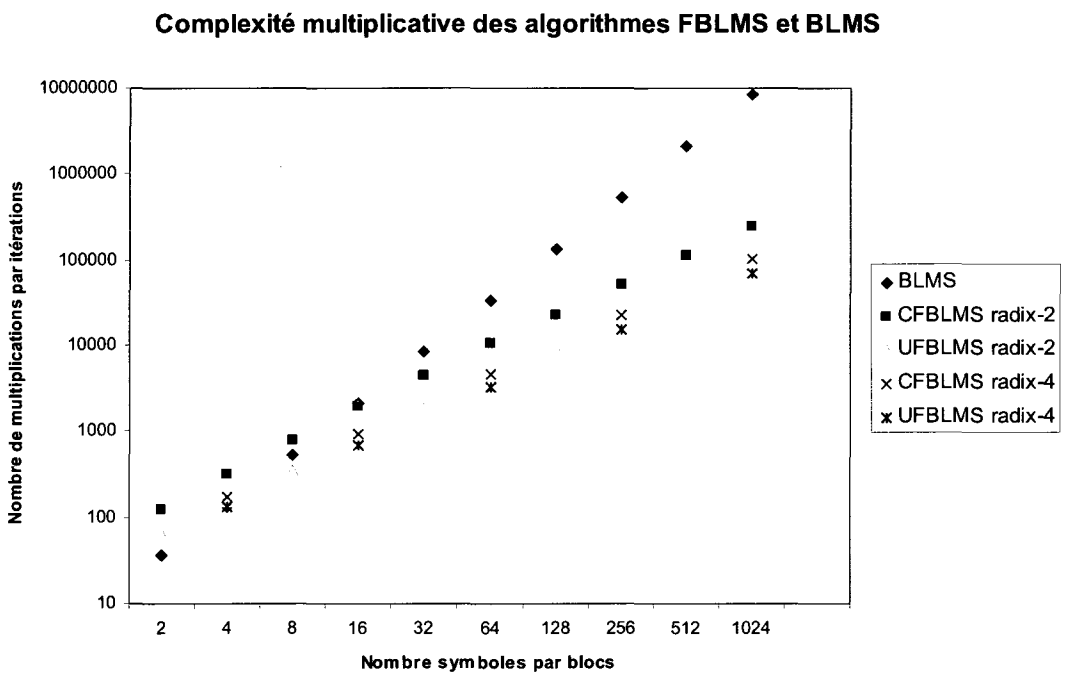


Figure 17 Complexité multiplicative des algorithmes FBLMS (chevauchement 50%) et BLMS

Sur la figure 17, on remarque que le nombre de multiplications requises par itérations croît selon $O\{N^2\}$ pour l'algorithme BLMS. Si bien qu'il est nécessaire de représenter l'histogramme sur une base logarithmique pour permettre de comparer visiblement les différents algorithmes. On note qu'à partir de 16 coefficients temporels par bloc, l'algorithme FBLMS devient plus efficace en terme de complexité de calcul. Plus on augmente le nombre de coefficients d'égaliseur, plus l'écart devient important. Il est

intéressant de noter que l'action de contraindre le gradient ajoute deux opérations de FFT/IFFT. Ainsi, du point de vue complexité de calcul, l'algorithme FBLMS est encore plus attrayant lorsqu'il est utilisé sans contrainte, soit en mode UFBLMS. Dans ce mode, on observe qu'à partir de seulement 8 coefficients, l'avantage se fait ressentir par rapport à son homonyme temporel. On note également que l'utilisation de l'algorithme *radix-4* rend encore plus attrayant l'égaliseur fréquentiel. Cependant, si l'on prévoit créer un égaliseur configurable, il serait plus intéressant de combiner les algorithmes *radix-2* et *radix-4*. De cette manière on obtient plus de choix et de résolution en ce qui à trait au nombre de coefficients d'égaliseur.

2.8 Conclusion

Dans ce chapitre, nous avons présenté les concepts du filtrage adaptatif avec une application directe, soit l'égalisation de canal. Nous avons porté notre attention sur des algorithmes d'adaptations basés sur la méthode *steepest descent*, soit les algorithmes LMS, BLMS et FBLMS. Nos recherches nous ont montrées que la probabilité de convergence de ces algorithmes est directement liée à la variation de puissance du signal d'entrée.

Nous avons porté un intérêt particulier sur la complexité de calcul des algorithmes mentionnés ci haut. Il a été démontré théoriquement que l'algorithme FBLMS est plus performant en terme de quantité de multiplications par bloc à partir de 8 coefficients en mode non contraint et 16 coefficients en mode contraint.

Nous avons également survolé les techniques d'égalisations aveugles en abordant les algorithmes CMA et MMA. Ces algorithmes sont particulièrement intéressants du point de vue efficacité spectrale. Il serait alors intéressant de trouver un moyen de combiner ces algorithmes avec l'égalisation fréquentielle. Au prochain chapitre, nous allons effectuer une conception d'un algorithme d'égaliseur fréquentiel qui nous permettra de combiner l'égalisation aveugle et entraînée dans le domaine fréquentiel.

CHAPITRE 3

CONCEPTION ET SIMULATION EN VIRGULE FLOTTANTE D'UN ÉGALISEUR FRÉQUENTIEL

3.1 Introduction

Avant de passer à l'étape d'implémentation matérielle de l'égaliseur fréquentiel, on établit la structure de ce dernier. Il faut en effet déterminer comment seront réalisés les blocs de décisions et de calcul d'erreur. L'objectif étant de créer un noyau d'égaliseur matériel le plus configurable possible avec une complexité relativement faible. Nous abordons donc ce chapitre en présentant les possibilités de configurabilité de l'égaliseur fréquentiel.

Par la suite, nous exposons quelques résultats simulations dans le but de caractériser les différents paramètres des architectures d'égaliseurs choisies. Nous intégrons donc l'égaliseur dans un environnement de communication numérique sans fil en le soumettant à des données altérées par un canal de type SUI. Nous utilisons des métriques de performance telles que l'EQM et le TEB pour évaluer l'égaliseur. Cette étape permet également de confirmer la fonctionnalité de l'algorithme développé en point flottant. Ainsi, nous avons employé ce modèle comme référence pour les étapes de quantifications et d'implémentation matérielles qui suivent.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

3.2 Environnement de simulation

L'objectif des simulations en point flottant est de vérifier la fonctionnalité de l'architecture d'égaliseur choisie. Pour y arriver, on cherche à reproduire les signaux typiques que l'on retrouve dans la pratique[1]. Dans un premier temps, on intègre un générateur de Bernoulli qui produit des données binaires distribuées de manière équiprobable. Ces données sont acheminées à un modulateur de symboles QAM. Tel que mentionné au chapitre 1, les symboles QAM doivent être mis en forme par un filtre

1 Les simulations en point flottant ont été réalisées à l'aide des outils Matlab/Simulink 7.1.

que mentionné au chapitre 1, les symboles QAM doivent être mis en forme par un filtre à cosinus surélevé pour éviter l'interférence entre symboles. Bien entendu, puisque nous utilisons un simulateur, les instants d'échantillonnage sont toujours parfaitement positionnés. En effet, nous avons choisi de ne pas simuler les boucles de génération d'horloges et de recouvrement de porteuses. Nous souhaitons effectuer des analyses comportementales en bande de base pour éviter les confusions et les problèmes de déverminage. De ce point de vue, le filtre de mise en forme ne semble pas utile puisque l'échantillonnage est contrôlé. Nous souhaitons tout de même inclure ces filtres afin de reproduire un signal plus près de la réalité que l'on retrouve dans une radio.

Par la suite, les symboles mis en forme par un filtre à racine de cosinus surélevé sont envoyés dans le canal de communication. Ce canal est d'abord constitué d'un générateur de bruit blanc Gaussien. Ce module est positionné à l'entrée du canal de sorte que nous puissions configurer un SNR par rapport au signal transmis, sans altérations engendrées par les interférences entre symboles inhérentes à notre canal. Notons que le SNR est posé à 30 dB pour toutes les simulations. Après, suivra un simulateur de multiples trajets basé sur les techniques SUI mentionnées au chapitre 1. Les étapes à accomplir pour générer les coefficients du canal SUI sont résumées à la figure 35, (IEEE 802.16, 2003).

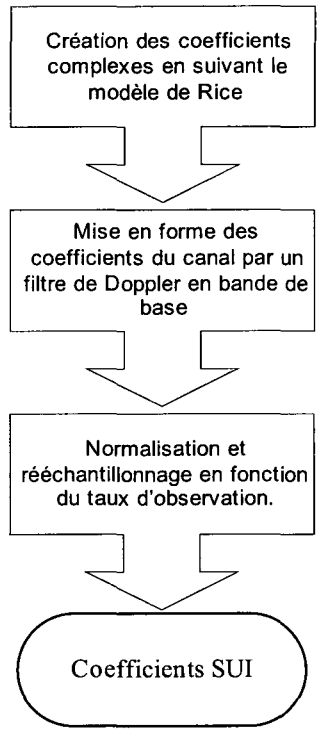


Figure 18 Étapes permettant de générer les coefficients du canal SUI

Il existe un nombre infini de possibilité de coefficients de canaux à tester. Au deuxième chapitre, nous avons montré que l'algorithme FBLMS est une traduction fréquentielle de l'algorithme LMS. Cet algorithme est bien connu et ses propriétés de convergence ont été étudiées exhaustivement dans la littérature. Il en est de même pour l'égalisation adaptative basée sur un filtre linéaire transversal avec les coefficients adaptés par la méthode LMS, (Widrow & Stern, 1984; Proakis, 2001; Bose, 2004; et plusieurs autres). Ainsi, nous avons conçu un modèle de simulation dans le but de vérifier la convergence des algorithmes que nous avons adaptés pour construire l'égaliseur fréquentiel. La première configuration de canal que nous avons choisie est composée de coefficients quasi fixes dans le temps. Nous avons configuré le générateur SUI de manière à ce qu'il puisse créer un modèle SUI-3 tel que montré au tableau VI. Pour la seconde configuration, nous avons opté pour un modèle SUI-5 dont les paramètres sont rapportés au tableau VII.

Tableau VI

Configuration du canal SUI-3

SUI-3				
	Trajet 1	**Trajet 2**	**Trajet 3**	**Unités**
Délai	0	1	2	us
Puissance	0	-8	-18	dB
Facteur K	19	0	0	-
Doppler	0.4	0.3	0.5	Hz
Antenne	Directionnelle, 30 degrés			

Tableau VII

Configuration du canal SUI-5

SUI-5				
	Trajet 1	**Trajet 2**	**Trajet 3**	**Unités**
Délai	0	4	10	us
Puissance	0	-8	-18	dB
Facteur K	7	0	0	-
Doppler	2	1.5	2.5	Hz
Antenne	Directionnelle, 30 degrés			

Les paramètres montrés au tableau VI et VII sont moyennés sur une seconde. Dans les deux cas, l'antenne émulée est de type directionnelle. Cette caractéristique a pour effet de stabiliser le trajet principal. Ainsi, la distribution des coefficients du trajet principal est basée sur le modèle de Rice avec un facteur K élevé. La fréquence de Doppler n'a donc pas beaucoup d'effet sur le premier trajet. On remarque que le modèle SUI-3 est composé de deux trajets secondaires dont les délais sont de 1 et 2 micro secondes par rapport au trajet principal et la variation de Doppler est très faible. Ceci fait en sorte que le taux de convergence de l'égaliseur n'a pas besoin d'être très élevé car les coefficients du canal ne varient pas très rapidement. Toutefois, pour le modèle SUI-5, on note que l'écart entre le premier trajet et les trajets secondaires sont de 4 et 10 micro secondes respectivement. Pour couvrir tous les échos, on doit augmenter la taille de l'égaliseur

Rapport-Gratuit.com

relativement à ce qui est requis pour le canal SUI-3. Théoriquement, il faut utiliser cinq fois plus de coefficients d'égaliseur pour le canal SUI-5. Ainsi, si on suppose un taux de transmission de un méga symboles par secondes, il faudra au moins 40 coefficients d'égaliseur pour couvrir tous les échos. Également à noter que les coefficients du canal SUI-5 varient plus rapidement puisque la fréquence de Doppler se situe autour de 2 Hz pour chaque trajet secondaire.

Au cours des simulations effectuées dans ce travail, nous avons vérifié les capacités de poursuite et de stabilité en simulant l'égaliseur fréquentiel sur une période d'une seconde. Pendant cette simulation, les coefficients varient dans le temps en amplitude et en phase. Leur comportement sur une seconde est montré à la figure 19 et 20. Notons que pour vérifier la convergence et la fonctionnalité des architectures fréquentielles, nous avons choisi de fixer les coefficients du canal. Les coefficients du canal SUI-3 et SUI-5 sont représentés graphiquement à la figure 21.

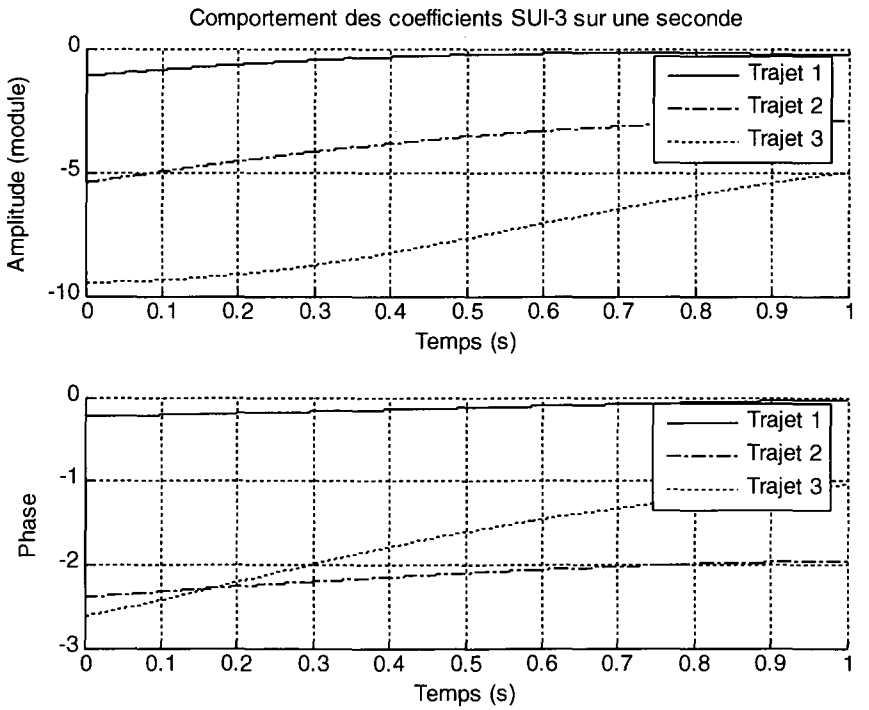


Figure 19 Amplitude et phase de coefficients SUI-3 sur une période de une seconde

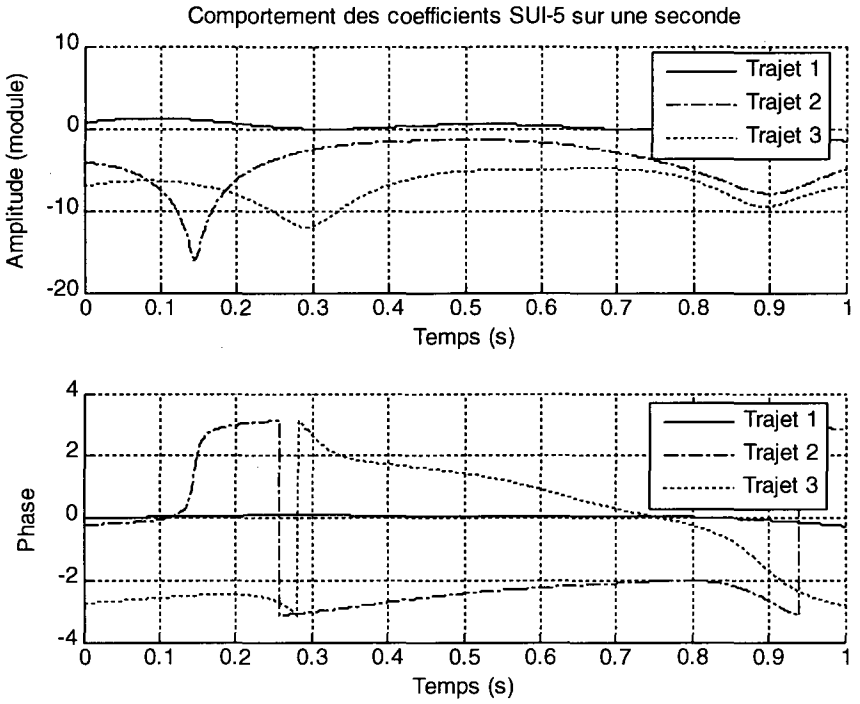
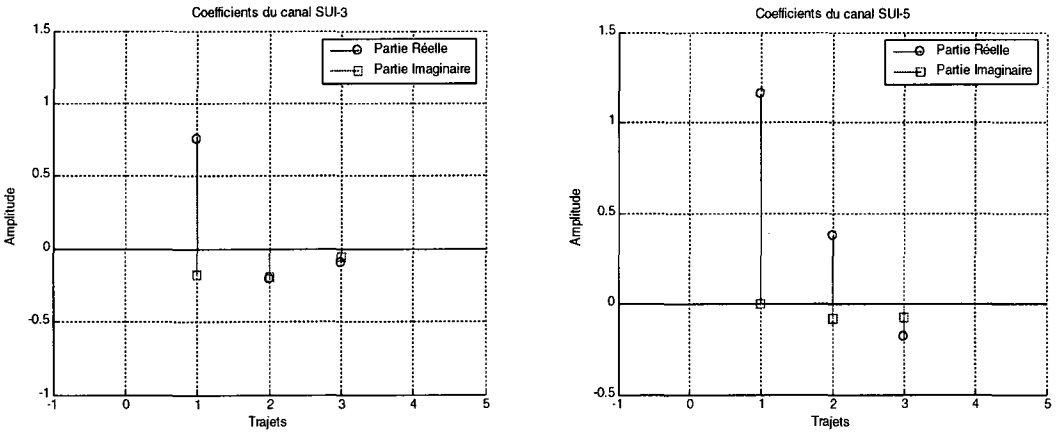


Figure 20 Amplitude et phase de coefficients SUI-5 sur une période de une seconde



a b

Figure 21 Amplitude réelle et imaginaire des coefficients des canaux SUI-3 et SUI-5 utilisés pour les analyses et vérification de l'algorithme FBMMA3M

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

À la sortie du canal de transmission, on retrouve à nouveau les filtres de mise en forme à racine de cosinus surélevé pour compléter la fonction de transfert de Nyquist. Ensuite, nous avons ajouté un module pour contrôler automatiquement les variations d'amplitude des symboles reçus. En pratique, on utilise un contrôleur automatique de gain numérique (CAGN) dans le but d'optimiser la plage dynamique de quantification. Ainsi, le CAGN effectue une première égalisation du canal tout en nous permettant de tirer le maximum de la plage dynamique pour les étapes de quantification en point fixe. La fonction de transfert du module CAGN est semblable à celle de l'algorithme LMS et s'écrit:

$$g(n)=g(n-1)+\mu\left(1-|r(n)|^2\right). \tag{3.1}$$

L'adaptation est dite aveugle puisque la référence est connue et fixe et ne nécessite pas de séquence d'entraînement. Le gain $g(n)$ est multiplié au signal reçu $r(n)$ pour ajuster l'amplitude moyenne autour de l'unité. La figure 22 illustre graphiquement les opérations accomplies dans le module CAGN. On note que le gain du *pas d'adaptation* est posé à 0.005 pour toutes les simulations. Ceci octroie une constante de temps de l'ordre du millième de seconde et permet ainsi de suivre aisément les variations du canal puisque nous utilisons un lien de communication basé sur le modèle SUI avec une fréquence de Doppler faible, soit 2.5 Hertz et moins.

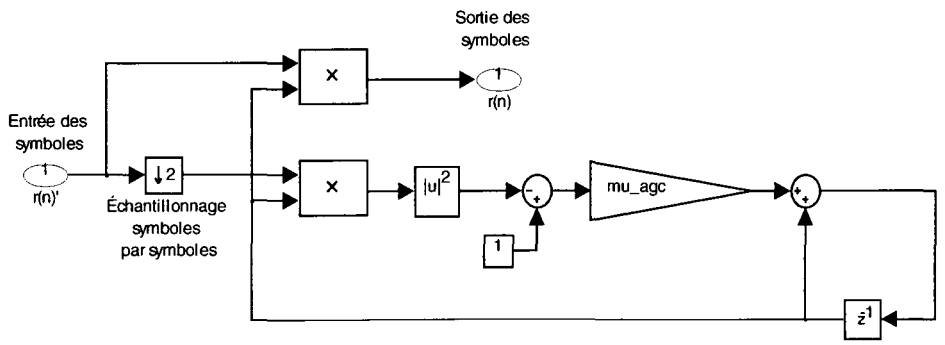


Figure 22 Schéma bloc du CAGN intégré dans le simulateur.

Sur la figure 22, le symbole de décimation par deux est employé pour permettre l'échantillonnage non fractionné au niveau de l'ajustement d'amplitude du CAGN. En effet, nous simulons un égaliseur à échantillonnage fractionnaire ($T_s/2$) puisque tel qu'il a été mentionné au chapitre 1, cette méthode offre de meilleures performances au niveau de la récupération d'horloge. Malgré que nous ne simulions pas ces étapes, nous effectuons l'échantillonnage fractionnaire afin d'allouer une comparaison plus juste du module VHDL avec le modèle de référence. Rappelons que l'objectif principal de ce travail est de créer un module configurable d'égaliseur fréquentiel ayant le potentiel d'être intégré dans une radio logicielle.

À la suite du module CAGN nous introduisons le module principal de cette présente étude, soit l'égaliseur fréquentiel. La figure 23 illustre un schéma bloc simplifié de l'environnement dans lequel nous vérifions le comportement des différentes architectures d'égaliseurs fréquentiels.

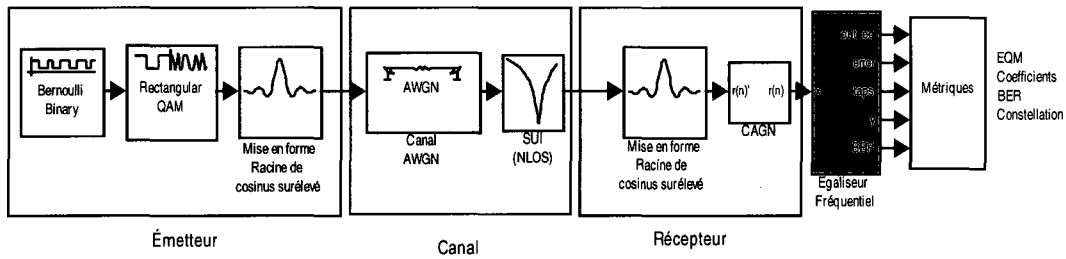


Figure 23 Modèle de simulation des différentes architectures d'égaliseurs fréquentiels

Sur la figure 23, on distingue le module *métriques*. Ce bloc est composé de fonctions permettant de générer différentes mesures telles que l'erreur quadratique moyenne (EQM), les coefficients dans le domaine temporel et fréquentiel, le taux d'erreur par symboles (TES) ainsi que la constellation égalisée. La métrique que nous préconisons pour vérifier et présenter la fonctionnalité de l'égaliseur est l'EQM. Pour calculer l'EQM,

on utilise un filtre afin de rendre la courbe plus lisse pour la présentation des résultats, (Dumais, 2004). L'équation de filtrage de l'erreur est:

$$eqm(n)=\lambda\, eqm(n-1)+(1-\lambda)|d(n)-\hat{s}(n)|^2 \tag{3.2}$$

où λ est le facteur de lissage et $d(n)$ représente le symbole désiré et peut être affecté à la séquence d'entraînement ou à la sortie du bloc de décision. En effet, nous employons deux méthodes pour mesurer l'EQM. Pour la première, on mesure l'EQM exacte, soit l'erreur entre le signal transmis et le signal égalisé. Et pour la seconde, on compare le signal détecté à la sortie du bloc de décision avec le signal égalisé. La seconde méthode est utile pour comparer les résultats de simulations avec ceux de l'implémentation matérielle.

3.3 Conception de l'architecture en point flottant de l'égaliseur fréquentiel

L'algorithme FBLMS est fonctionnel et a été vérifié dans plusieurs articles IEEE, (Ferrara, 1980; Clark, Mika & Parker 1981; Mansour & Gray,. 1982; Lee & Un, 1989; Shynk, 1992; Farhang-Boroujeny, 2000). Toutefois, puisque notre objectif principal est de réaliser un noyau matériel d'égaliseur fréquentiel, on cherche à créer une architecture la plus configurable possible tout en réduisant au mieux la complexité matérielle. À la figure 24, nous reprenons l'algorithme FBLMS.

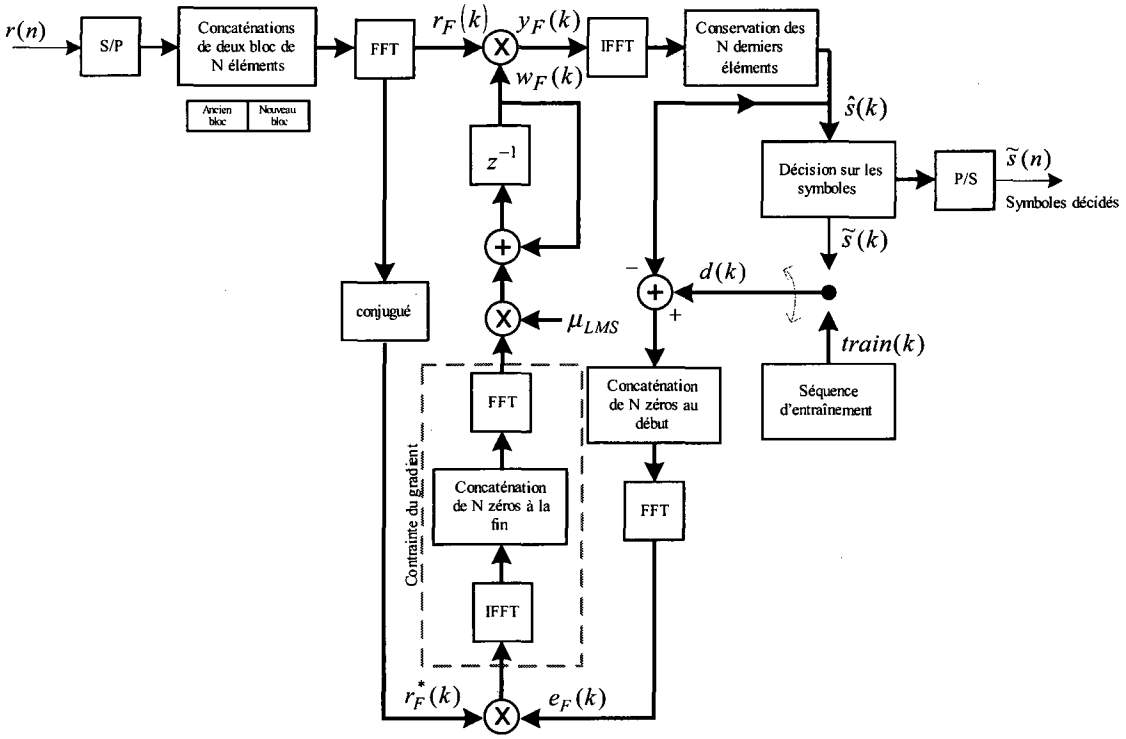


Figure 24 Schéma bloc de l'égaliseur FBLMS avec $N' = 2N$

3.3.1 Réduction de la complexité

Pour réduire la complexité, il serait possible de retirer la transformée de Fourier qui permet de faire le passage de l'erreur temporelle à l'erreur fréquentielle en calculant la séquence d'entraînement à l'avance. Les coefficients précalculés de la séquence d'entraînement seraient alors rangés dans une mémoire vive ou une mémoire morte. Puis, un mécanisme de contrôle permettrait de synchroniser le calcul de l'erreur directement dans le domaine des fréquences. La séquence d'entraînement est connue au récepteur. Elle peut donc être convertie en fréquence par l'équation suivante:

$$\boldsymbol{d}_F(k) = FFT\begin{Bmatrix} \mathbf{0} \\ \boldsymbol{d}(n) \end{Bmatrix}. \tag{3.3}$$

La figure 25 illustre schématiquement cet algorithme.

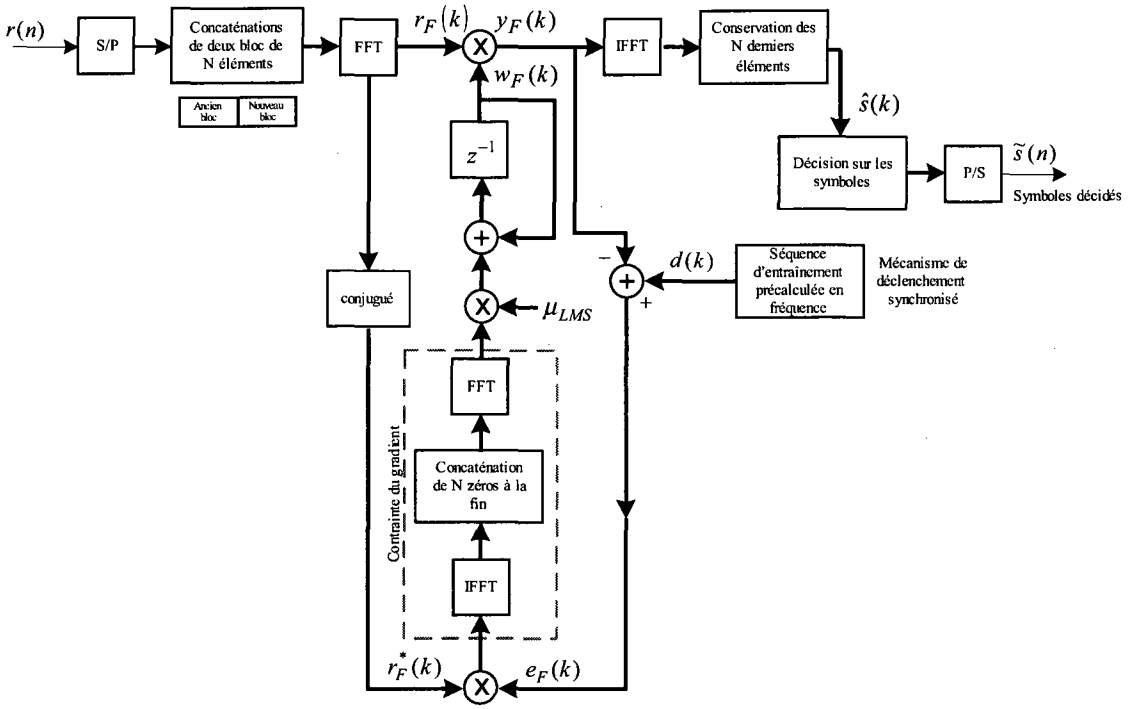


Figure 25 Schéma bloc de l'égaliseur FBLMS avec calcul de l'erreur directement en fréquence et et $N' = 2N$

La méthode présentée à la figure 25 est intéressante du point de vue complexité matérielle puisqu'elle permet de retirer un module de FFT. Cependant, en effectuant cette modification, on se contraint à faire converger l'égaliseur seulement en mode d'entraînement. Sachant qu'en général on n'utilise que 10% de la trame de données pour la séquence d'entraînement (Rappaport, 2000), on s'aperçoit que la convergence peut s'avérer très lente. De plus, même si à un instant donné, les coefficients de l'égaliseur sont optimaux, les paramètres du canal changent constamment. Ainsi, puisque la sortie du bloc de décision ne peut être utilisée avec un tel montage, cet égaliseur ne sera pas très performant en mode poursuite. D'autant plus que les coefficients ne pourront être adaptés qu'en mode d'entraînement, jamais en mode dirigé.

Dans (Shynk, 1992), on présente également une structure modifiée de l'algorithme FBLMS où l'on effectue le calcul de l'erreur directement dans le domaine fréquentiel tel

qu'illustré à la figure 25. En plus, on retire complètement le processus de temporisation par la méthode *overlap save*. Cette technique s'avère très intéressante du point de vue complexité matérielle puisqu'elle permet d'extraire toute la partie de contrainte du gradient, en plus d'allouer un traitement des données deux fois plus rapidement. En effet, sans le tampon de chevauchement à 50%, on peut doubler la fréquence de traitement des données. Cependant, après quelques essais, nous avons rapidement conclu que cette technique n'est pas du tout efficace pour réduire l'erreur quadratique moyenne avec un signal indépendant et corrompu par un canal également indépendant tel que ceux utilisés en communication numérique. En effet, avec un tel algorithme, on se trouve à négliger totalement les effets indésirables d'une convolution circulaire. Ceci fait en sorte que les coefficients de l'égaliseur sont toujours corrompus et il devient impossible d'effectuer une inter corrélation juste du gradient. La figure 26 illustre la constellation optimale de la structure sans chevauchement des données pour un signal 4-QAM avec les coefficients de canal SUI-3. On remarque que même pour une constellation de bas niveau, les points ne se distinguent pas des zones de décisions.

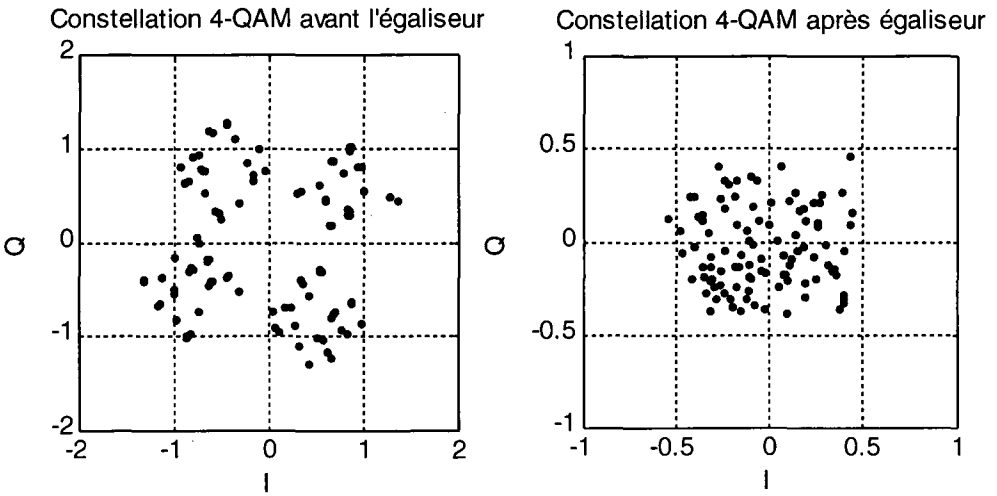
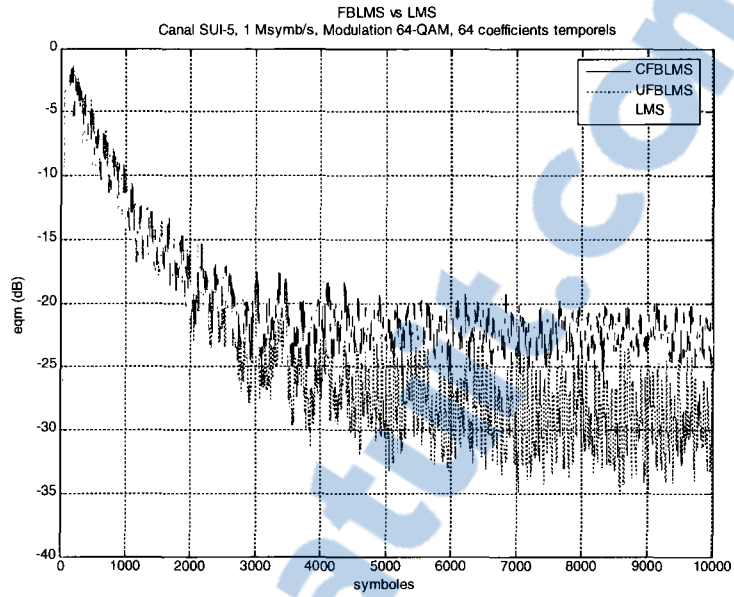


Figure 26 Constellation avant et après l'égaliseur FBLMS avec calcul de l'erreur directement en fréquence et $N' = N$

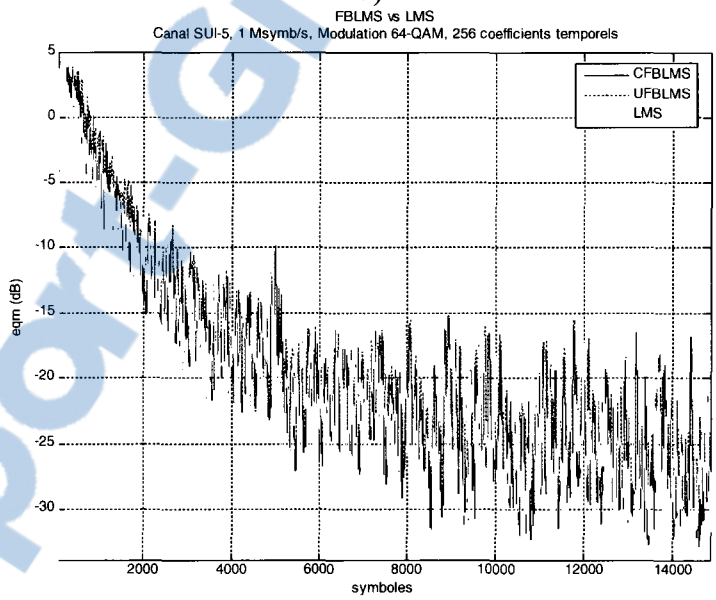
Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

Puisque cette structure d'égaliseur ne converge pas suffisamment lorsqu'on la configure en mode de convolution circulaire, on peut tout de suite éliminer cette méthode pour l'implémentation finale. Également, la stratégie de calcul de l'erreur directement dans le domaine fréquentiel est écartée puisqu'elle ne permet pas d'utiliser l'égaliseur en mode dirigé mais seulement en mode d'entraînement.

Si l'on poursuit sur la même idée de réduction de complexité, on peut penser à éliminer la contrainte du gradient. Cette omission abaisserait la complexité de calcul en nous permettant d'éliminer deux calcul de FFT/IFFT par bloc. Tel que nous l'avons vu au chapitre deux, l'algorithme UFBLMS se comporte sensiblement de la même manière que l'algorithme CFBLMS pour certaines conditions de canal. Nous avons vérifié et comparé le comportement de ces deux algorithmes avec le canal SUI-5 présenté plus tôt.



a)



b)

Figure 27 Courbe de convergence des algorithmes FBLMS contraint (CFBLMS) et non contraint (UFBLMS). a) Canal SUI-5, 64 coefficients d'égaliseur b) Canal SUI-5, 256 coefficients d'égaliseur.

Sur la figure 27, nous présentons la courbe de convergence de l'erreur quadratique moyenne pour les égaliseurs basés sur les algorithmes FBLMS contraint (CFBLMS), non

Rapport-Gratuit.com

contraint (UFBLMS) et l'homonyme de l'égaliseur fréquentiel de type FBLMS, soit l'égaliseur LMS. La valeur de l'indice λ est fixée à 0.1. Pour ces résultats, la séquence d'entraînement est transmise tout au long de la simulation. Celle-ci est d'une durée de 15 millisecondes. Le taux de transmission est fixé à un méga symbole par seconde. Ainsi, pendant toute cette simulation, on a simulé 15000 échantillons. Tel que nous l'avons indiqué au chapitre 2, l'algorithme CFBLMS est une traduction identique de l'algorithme LMS dans le domaine des fréquence. Ainsi, les deux algorithmes se comportent de manière très semblables. Sur la figure 27, la pratique confirme la théorie puisque la courbe d'EQM de l'égaliseur LMS est presque identique à celle de l'égaliseur CFBLMS.

On note que l'EQM des trois algorithmes est réduite sur la figure 27 b) par rapport à la figure 27 a). En effet, nous avons quadruplé le nombre de coefficients d'égaliseur. Ce changement fait en sorte que l'on peut couvrir plus d'échos et ainsi diminuer les probabilités d'erreurs engendrées par le canal de transmission. Lorsque le nombre de coefficients d'égaliseur n'est pas suffisamment grand pour couvrir tous les échos du canal de transmission, l'EQM n'est pas à son minimum. Sur la figure 27 a), on observe que l'EQM de l'égaliseur UFBLMS est inférieur aux deux autres. On attribue cette caractéristique au fait que le gradient n'est pas contraint. Effectivement, en ne contraignant pas le gradient, les coefficients du deuxième bloc de traitement ne sont pas nuls. Le processus d'adaptation se sert alors de ces coefficients pour égaliser le canal. La même remarque n'est pas applicable sur la figure 27 b) car la longueur effective de l'égaliseur est suffisamment grande. Un ajout de coefficients d'égaliseur n'a donc pas d'impact évident sur l'EQM. La figure 28 présente les coefficients temporels du bloc effectif et du bloc de chevauchement des égaliseurs CFBLMS et UFBLMS. Fidèle à son homonyme temporel, l'égaliseur CFBLMS possède des coefficients nuls dans la section de chevauchement. Pour sa part, l'égaliseur UFBLMS est muni de coefficients non nuls dans la section virtuelle (chevauchement). Ainsi, d'un point de vue pratique, l'égaliseur UFBLMS est virtuellement deux fois plus long que l'égaliseur CFBLMS. Cependant, les coefficients du deuxième bloc sont corrompus par l'effet d'inter corrélation circulaire.

Ce phénomène insère une certaine instabilité que l'on peut distinguer par de plus grandes variations de l'EQM sur la figure 27 a). Tel qu'il a été présenté au chapitre 2, cette particularité peut s'avérer constructive ou destructive, dépendamment des conditions de canal. Dans le cas d'un canal à variation lente tel que le modèle SUI, l'algorithme UFBLMS s'avère une option très intéressante car il se comporte sensiblement de la même manière que l'égaliseur CFBLMS mais il ne requiert que trois transformées de Fourier par itération. C'est une économie de deux blocs de transformée de Fourier par rapport à l'algorithme CFBLMS.

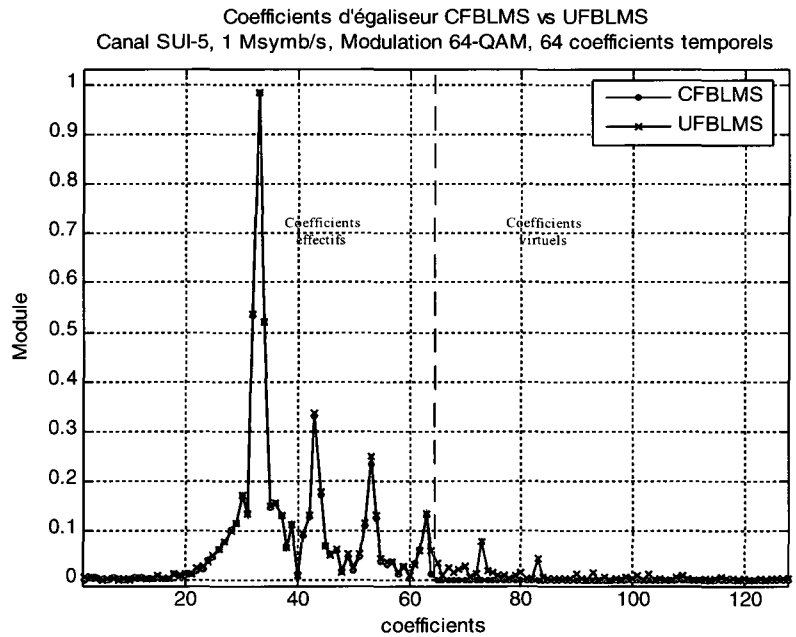


Figure 28 Coefficients des égaliseurs CFBLMS vs UFBLMS.

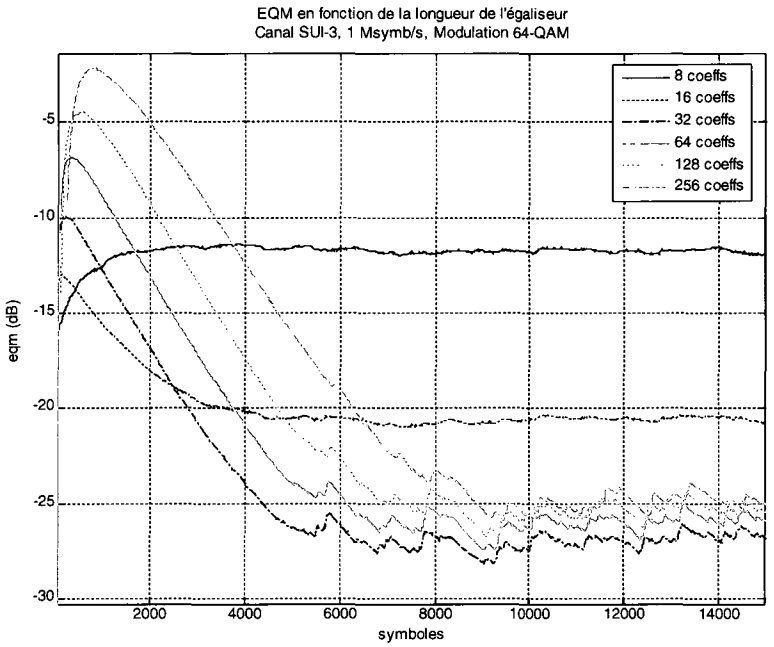
3.3.1.1 Effet de la longueur de l'égaliseur sur la convergence de l'EQM

Au premier chapitre, nous avons observé que la déconvolution (ou l'inversion) du canal résulte en une infinité de coefficients. Par le biais d'un exemple, on remarque que la puissance des résidus de la déconvolution diminue suivant N. Ainsi, à un certain seuil donné, il devient inutile d'augmenter la longueur N de l'égaliseur car la réduction de l'interférence entre symbole n'est plus perceptible. C'est ce que nous constatons sur la

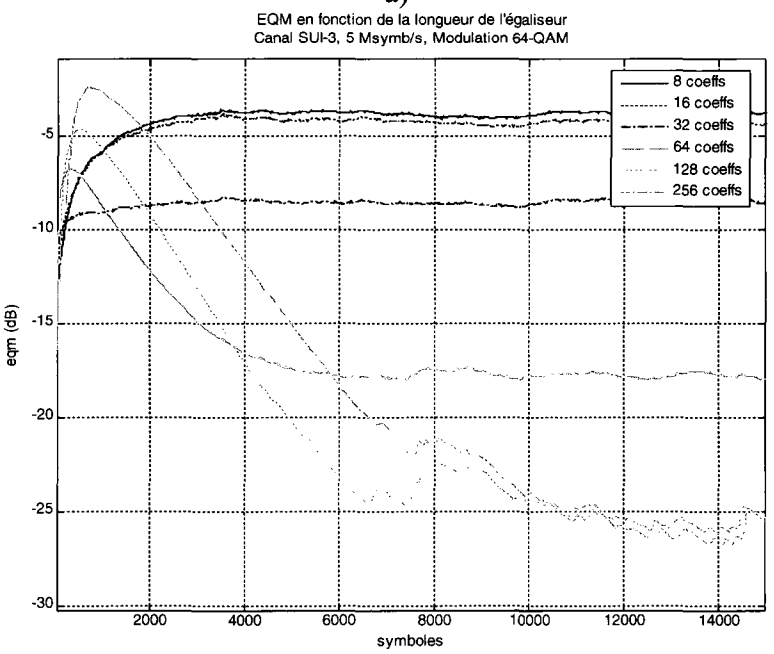
Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

figure 29 a). De ce fait, il est inutile de poser plus de 128 coefficients temporels pour le canal SUI-3 que nous avons présenté plus haut. Un égaliseur plus long ne réduirait pas l'EQM et serait plus susceptible à l'instabilité. En effet, n'oublions pas que la convergence d'un égaliseur linéaire dépend de la variation de la puissance des symboles reçus et de la longueur de l'égaliseur (voir chapitre 2). Au premier chapitre, nous avons aussi démontré que la longueur utile d'un égaliseur linéaire dépend des conditions de canal et également de la vitesse de transmission des symboles. Par l'équation 1.18 on remarque que la longueur relative des échos est linéairement dépendante du taux de symboles. Ainsi, si on quintuple le nombre de symboles transmis par secondes, il faudra aussi quintupler la longueur de l'égaliseur. Donc, pour un canal SUI-3 tel que présenté plus haut, il faudrait maintenant au moins 200 coefficients d'égaliseur pour couvrir tous les échos. Cette affirmation est renchérie par la figure 29 b), où l'on aperçoit que l'EQM est remarquablement réduite avec un égaliseur de 256 coefficients temporels.

Sur la figure 29, on observe que plus l'égaliseur est long, plus le temps de convergence augmente. Cette remarque renchérit l'idée qu'un égaliseur de type fréquentiel très long voit son utilité dans les environnements à faible variation dans le temps. Il est intéressant de noter que plus l'EQM est réduite, plus il est possible d'augmenter le niveau de modulation MQAM. En effet, au premier chapitre, nous avons vu que le rapport signal sur bruit doit être augmenté en fonction du niveau M de la modulation MQAM. Ainsi, plus on réduit l'EQM, plus on réduit la quantité de bruit présent dans le signal. On peut concrètement visualiser cette propriété en observant la figure 29 b). Dans ce contexte, un égaliseur de 64 coefficients ne pourrait pas être utilisé dans un modem 16-QAM si les spécifications exigeaient un taux d'erreur inférieur à 10^{-6}. Cependant, un égaliseur de 128 coefficient pourrait être utilisé. On note également qu'il est inutile, voir même nuisible d'augmenter la longueur de l'égaliseur à 256 coefficients puisque l'EQM n'est pas réduite et le taux de convergence est affecté à la baisse.



a)



b)

Figure 29 EQM de l'égaliseur FBLMS lorsque soumis au canal SUI-3.
a) 1 Msymb/s b) 5 Msymb/s

3.3.2 Égalisation fréquentielle autodidacte

Toujours d'un point de vue complexité, un noyau d'égaliseur fréquentiel configurable serait très attrayant s'il pouvait être utilisé en mode aveugle. En effet, cette approche permettrait d'éliminer la séquence d'entraînement et du même coup, augmenter l'efficacité spectrale. Dans la littérature, en se basant sur le théorème de Parseval, on montre qu'on peut convertir le calcul de l'erreur CMA entièrement dans le domaine des fréquences. Ainsi, on présente un égaliseur fréquentiel basé sur l'algorithme CMA. L'équation de l'erreur CMA est directement calculée dans le domaine des fréquences comme suit:

$$e_{FBCMA}(k)=y_F(k-1)\left(\frac{1}{N'^2}\sum_{j=0}^{N'-1}|y_F(k-1+j)|^2-R_{CMA}^2\right) \tag{3.4}$$

où nous rappelons que N' correspond à deux fois la longueur de l'égaliseur dans le cas où l'on effectue le filtrage par la méthode de chevauchement et sauvegarde à 50%. Par cette méthode, on peut éliminer le bloc de FFT servant à convertir l'erreur temporelle dans le domaine fréquentiel. Ce qui rend cette technique fréquentielle encore plus attrayante. Toutefois, on note que l'équation 3.4 n'est pas triviale. Elle requiert $N'+1$ multiplications complexes par bloc de N' symboles, en plus d'exiger un décalage de $\log_2 N'^2$ bits par bloc pour réaliser la division $1/N'^2$ avec N' qui s'exprime par 2^x. On remarque ainsi que la suppression d'un module de FFT ne produit pas de réduction de la complexité de traitement compte tenu de la lourdeur du calcul de l'erreur CMA dans le domaine des fréquences. Dans le cadre de leurs travaux présentés dans (Yang Yoon & Cho Nam, 1995), les auteurs ont effectué des simulations en point flottant concluantes. Les résultats de convergences sont comparables à l'égalisation temporelle basée sur l'algorithme CMA. Cependant, cette méthode comporte ses inconvénients. Dans un premier temps, tel que nous l'avons mentionné au deuxième chapitre, l'algorithme CMA est susceptible aux ambiguïtés de phases. De plus, l'algorithme développé dans cet article ne permet pas l'utilisation d'un mode d'adaptation auto dirigé puisque les données

sortantes du bloc de décision sont dans le domaine temporel. Pour utiliser les deux modes, soit aveugle et auto dirigé, il faudrait replacer le module de FFT pour convertir la sortie du bloc de décision dans le domaine des fréquences tel qu'on le fait avec l'algorithme FBLMS classique. Cette approche ferait augmenter la complexité de calcul de l'égaliseur justement à cause de l'étape de calcul de l'erreur CMA en fréquence.

3.3.2.1 L'algorithme FBMMA3M

C'est pour les raisons mentionnées ci dessus que nous proposons une nouvelle méthode d'égalisation fréquentielle autodidacte. Si l'on observe la figure 24, on note que le calcul de l'erreur de l'algorithme FBLMS est réalisé dans le domaine temporel avant d'être converti en fréquence pour l'adaptation des coefficients. Ainsi, puisque cette erreur est obtenue dans le domaine temporel, il n'y a rien qui nous empêche de remplacer ce signal d'erreur par un signal d'erreur de type MMA. Dans le domaine temporel, nous avons vu au chapitre 2 que les coefficients doivent être initialisés pour assurer la convergence de l'égaliseur. Cette initialisation peut être accomplie directement dans le domaine des fréquences par l'équation suivante:

$$w_{F\,init}(k) = FFT\left\{[\underbrace{0\ \cdots\ 0}_{N/2-1}\ 1\ \underbrace{0\ \cdots\ 0}_{N/2}][\underbrace{\mathbf{0}}_{N}]\right\}. \tag{3.5}$$

La figure 30 illustre l'algorithme FBMMA à trois modes d'adaptation que l'on nomme FBMMA3M. Avec cette structure, on peut utiliser les trois modes de calcul de l'erreur, soit le mode dirigé, le mode entraîné et le mode aveugle. La combinaison de ces trois méthodes permet d'améliorer les performances de convergence des coefficients de l'égaliseur.

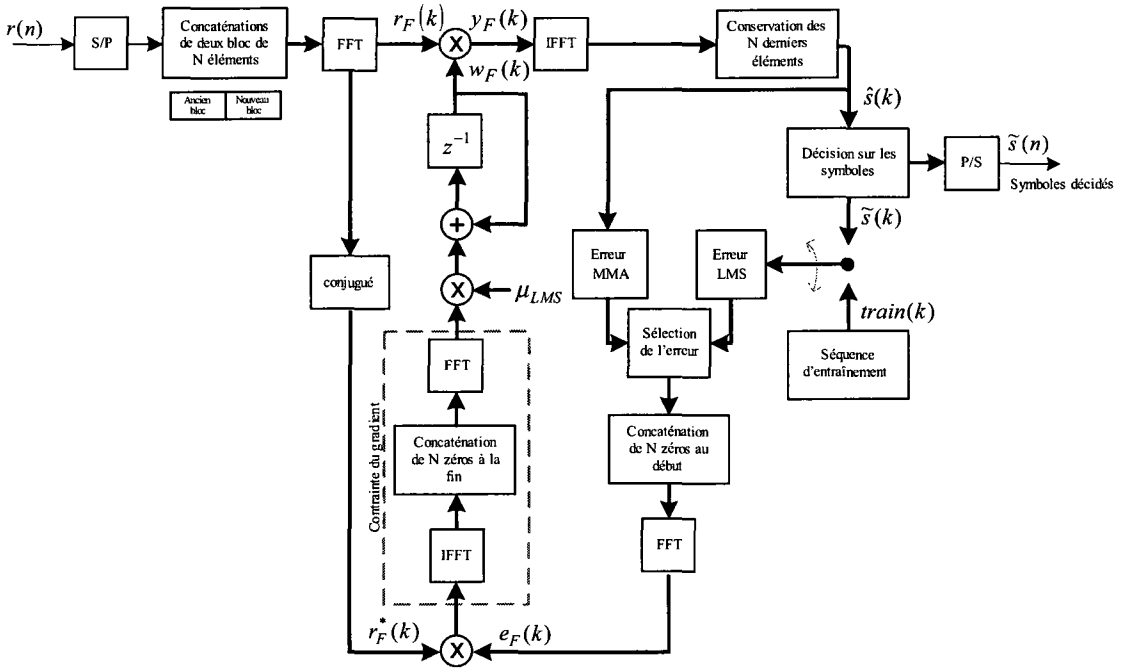


Figure 30 Algorithme FBMMA3M

Nous savons que les probabilités de convergence des coefficients sont supérieures lorsqu'on utilise une séquence d'entraînement, (Rappaport, 2003). Cependant, il est plus pratique de se servir d'un algorithme aveugle puisqu'on peut éviter les problèmes liés à la synchronisation de la trame et on peut également augmenter l'efficacité spectrale du modem en attribuant de l'information utile sur toute la trame de donnée, sans séquence d'entraînement. Dans la plupart des protocoles de communications, une section de la trame est attribuée pour des fins de contrôle entre l'émetteur et le récepteur. Cette dite section est généralement cyclique et correspond à plus ou moins 10 pourcent de toute la trame de donnée, (Rappaport, 2003). Ces données sont connues par le récepteur et peuvent être utilisées pour l'entraînement de l'égaliseur. Cette observation nous incite à conserver la fonction d'entraînement dans l'architecture de l'égaliseur FBMMA3M.

Sur la figure 30, on remarque le bloc de sélection de l'erreur. Ce module permet d'effectuer la sélection de l'erreur la plus vraisemblablement valide. Cette option est utile lorsqu'on utilise l'égaliseur en double mode, soit le mode aveugle et le mode auto

dirigé. En effectuant nos recherches dans la littérature, nous avons conservé deux algorithmes de sélection de l'erreur. Le premier est basé sur l'EQM, (Sayed, 2005), et le second sur la sortie de l'égaliseur $\hat{s}(k)$, (Weerackody & Kassam, 1994). Le premier algorithme consiste à relever l'erreur quadratique moyenne instantanée obtenue de la différence entre la sortie du bloc de décision et la sortie de l'égaliseur, puis de vérifier si le poids de cette EQM est assez faible pour utiliser l'erreur sur la décision ou si elle est trop élevée et il faut utiliser l'erreur MMA. Cet algorithme de sélection est résumé à l'équation 3.3. Le *seuil* est déterminé par essais et erreur. Ce dernier est fonction du niveau de modulation MQAM et du degré de distorsion engendré par le canal de transmission.

$$\begin{aligned} &si \quad 0 \leq EQM \leq seuil \quad alors \\ &\quad e(k) = e_{LMS}(k) \\ &autrement \\ &\quad e(k) = e_{MMS}(k) \end{aligned} \tag{3.6}$$

Pour le second algorithme de sélection, on utilise une zone de confiance dans laquelle on présume que le symbole à la sortie de l'égaliseur est correctement positionné dans sa région dédiée. Donc, le module de sélection de l'erreur vérifie si la valeur absolue de l'erreur réelle et imaginaire entre le symbole décidé et le symbole égalisé est inférieure ou égal à l'indice d. Et l'indice d est déterminé empiriquement en fonction du niveau de modulation MQAM et du canal de transmission. Cet algorithme de sélection est plus efficace que le premier mentionné à l'équation 3.3 puisqu'il prend distinctement en considération la partie réelle et imaginaire du symbole reçu. Ainsi, les probabilités que le module de sélection effectue la meilleure décision sur l'erreur sont rehaussées. Surtout lorsque le niveau de modulation est élevé. La figure 31 illustre la zone de confiance où l'on considère l'erreur LMS et la zone où l'on considère l'erreur MMA pour une constellation 16 QAM.

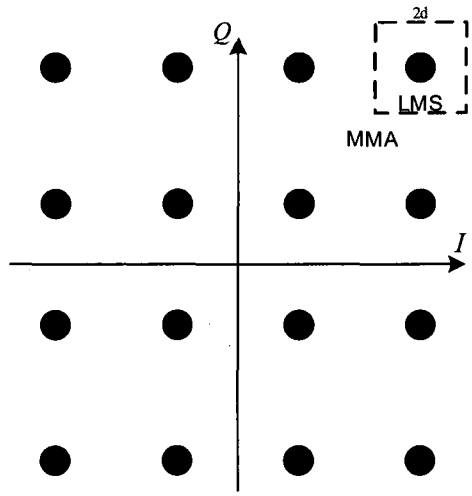


Figure 31 Sélection de l'erreur avec l'algorithme basé sur la sortie de l'égaliseur.

Le second algorithme de sélection, soit celui illustré à la figure 31, est plus performant en terme de convergence avec des constellations supérieures à 4-QAM pour les cas du canal SUI-3 et SUI-5 présentés en début de chapitre. Pour une constellation 64-QAM, seul le second algorithme nous a permis d'obtenir une convergence assez faible de l'EQM pour éviter les erreurs de décision sur les symboles. Ainsi, pour le reste de nos analyses et de la conception de l'architecture de l'égaliseur fréquentiel, nous employons l'algorithme de sélection de l'erreur illustré à la figure 31.

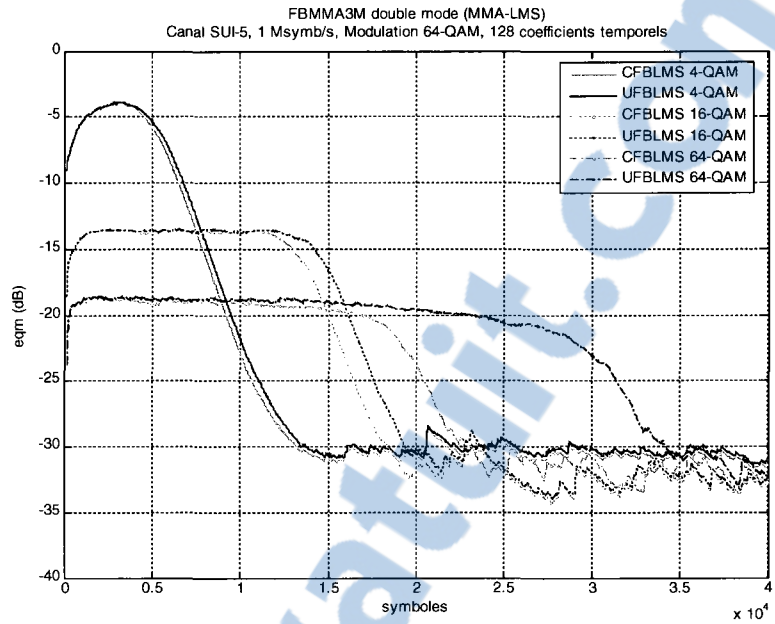


Figure 32 EQM estimée en mode MMA-LMS.

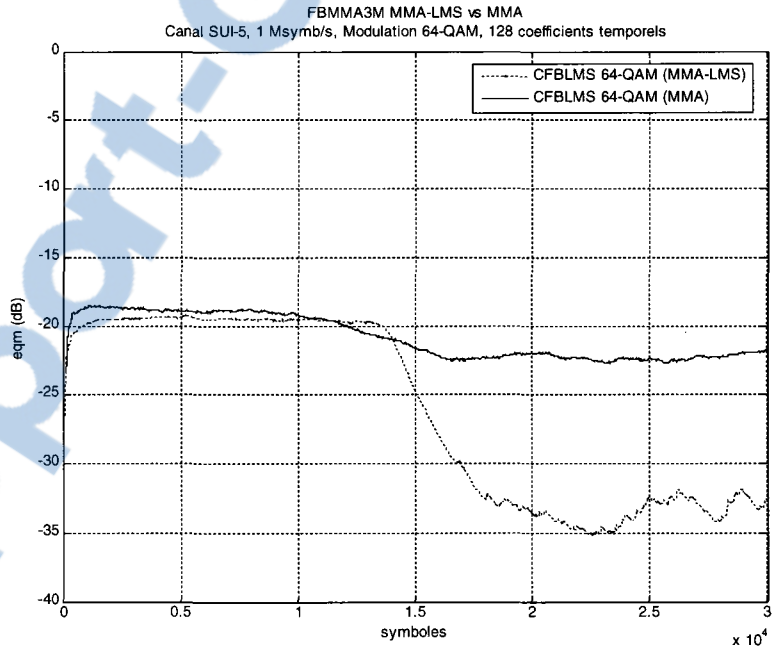


Figure 33 EQM estimée (MMA-LMS) vs MMA

Rapport-Gratuit.com

La figure 32 illustre la courbe de convergence de l'algorithme FBMMA3M. On observe que plus le niveau de modulation est élevé, plus la convergence lente. En effet, plus les points de la constellation se rapprochent, plus il y a risque d'erreur au niveau de la décision sur les symboles à en raison de la sensibilité accrue au bruit. Sur la figure 33, on constate de manière pratique l'effet de la combinaison des modes autodidactes et dirigés. Cette courbe démontre clairement que le taux de convergence de la combinaison des algorithmes MMA et LMS est de beaucoup amélioré. On constate même que sans cette combinaison, l'EQM n'est pas réduite à un niveau acceptable pour un taux d'erreur de l'ordre de 10^{-6}.

3.3.3 Pas d'adaptation variable

Tel que nous l'avons mentionné au chapitre deux, la convergence des coefficients d'un algorithme adaptatif basé sur la méthode *steepest descent* est fortement influencée par les valeurs propres de la matrice d'autocorrélation du signal d'entrée. Cette matrice est intimement liée à la variation de puissance des échantillons reçus. Ainsi, pour accentuer les performances de convergence, le poids du *pas d'adaptation* pourrait être variable en fonction de la puissance du signal d'entrée. Dans la littérature, les algorithmes NLMS et NFBLMS sont fondés sur cette technique, (Bourojeny, 2000). Pour l'égalisation fréquentielle, l'algorithme NFBLMS pourrait s'appliquer. La variabilité du *pas d'adaptation* s'écrit comme suit:

$$\mu_i(k) = \frac{\mu_0}{E\left[\left|r_F(k)\right|^2\right]} . \tag{3.7}$$

L'équation 3.2 requiert une opération de moyenne, une mise au carré complexe ainsi qu'une division pour chaque bloc de longueur N'. C'est très coûteux en terme de complexité de calcul. Pour cette raison, dans (Boroujeny, 1998; Dmochowski, 2001), on simplifie l'équation 3.2 par celle ci:

$$\mu_i(k)=\frac{\mu_0}{\beta\sigma_r^2(k-1)+(1-\beta)\sigma_r^2(k)} \tag{3.8}$$

où β représente l'indice de lissage ou de moyenne. Il doit être choisi entre 0 et 1. Et σ_r^2 désigne la puissance des symboles reçus à l'entrée de l'égaliseur. Nous avons vérifié si le taux de convergence est augmenté en utilisant le *pas d'adaptation* variable pour les canaux illustré en début de chapitre. Afin d'accroître la stabilité de l'algorithme et pour éviter les divisions par zéro, une valeur constante (0.1) a été ajoutée au dénominateur de l'équation 3.1. Nos résultats nous indiquent qu'il y a un gain de performance mais il n'est pas significatif pour les types de canaux que nous utilisons. En effet, si l'on mesure le degré de complexité de calcul ajouté à l'égaliseur versus le gain que l'on obtient avec cet ajout, il n'est pas très intéressant de retenir cette méthode. Cependant, il pourrait être avantageux d'intégrer la possibilité de varier le *pas d'adaptation* pendant le processus d'adaptation des coefficients. Cette technique permettrait au module de contrôle du modem d'augmenter le taux de convergence dans la phase adaptative et également de réduire l'erreur quadratique moyenne dans la phase stable. Puisque l'égaliseur que nous concevons dans ce travail est voué à des applications à liaison fixe, le taux de variation du canal ne sera pas très élevé. Ainsi, on pourrait intégrer un contrôleur externe, géré par un processeur fonctionnant à un taux plus faible pour contrôler la variabilité du *pas d'adaptation* en fonction de la puissance des symboles d'entrée. Sur les figure 34 et 18, on observe que l'intégration de quatre niveaux de μ suffit pour accomplir sensiblement la même performance que l'algorithme NFBLMS en continu.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

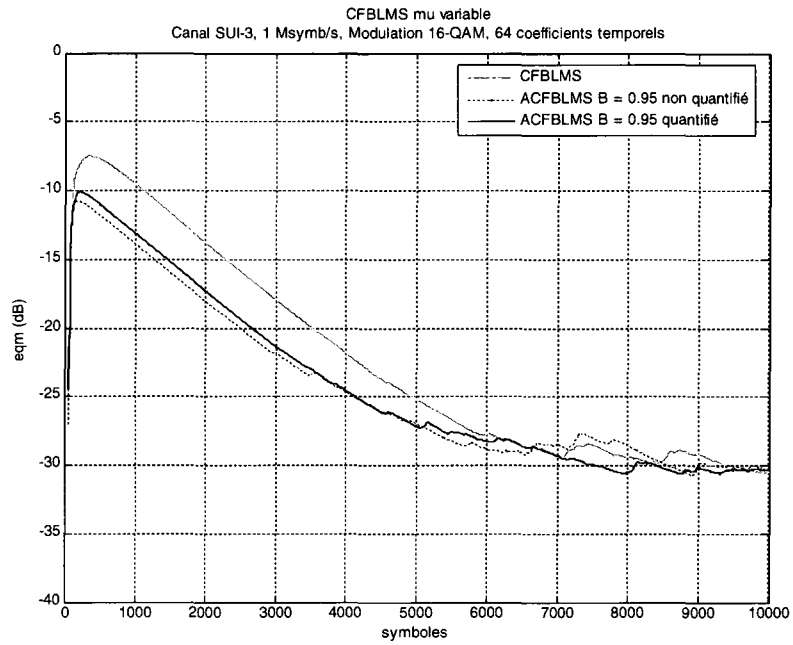


Figure 34 EQM estimé avec *pas d'adaptation* variable.

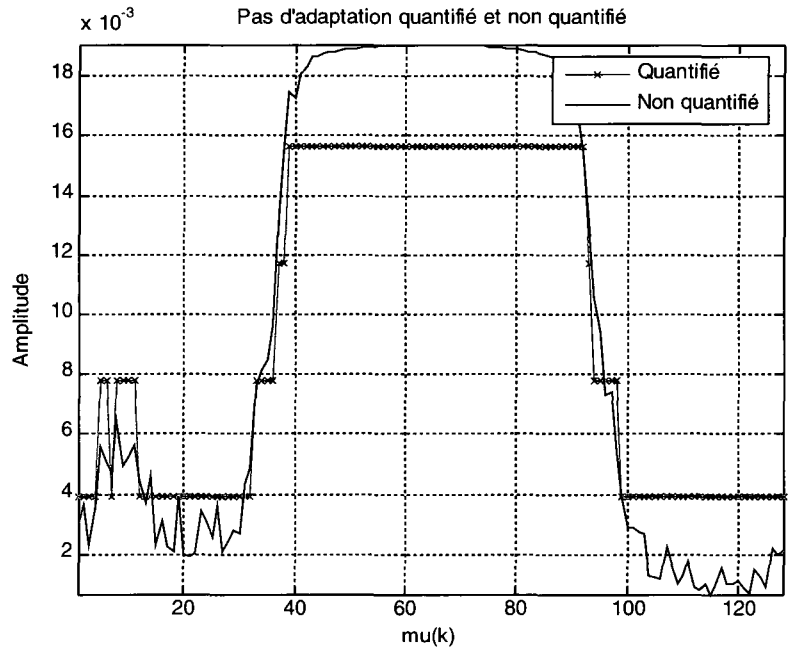


Figure 35 *Pas d'adaptation* variable quantifié et non quantifié

3.4 Conclusion

Au troisième chapitre, nous avons accompli la conception et l'analyse en point flottant de l'algorithme FBLMS. Nous avons confirmé que la convergence des coefficients est, entre autre, dépendante de l'étalement des délais et du taux de transmission des symboles. De ce fait, nous avons montré que la longueur de l'égaliseur est directement liée aux performances de ce dernier. Ainsi, plus le filtre de l'égaliseur est long, plus il est possible de réduire l'interférence entre symboles engendrées par le canal. Cependant, nous avons observé que le taux de convergence est réduit plus l'égaliseur est long.

Nous avons démontré qu'il est possible d'intégrer une combinaison des algorithmes de calcul d'erreur LMS et MMA avec un égaliseur fréquentiel. Nous avons baptisé cette nouvelle combinaison par l'acronyme FBMMA3M. Le terme (3M) indique que l'égaliseur peut être utilisé en trois modes différents. Soit le mode aveugle, le mode dirigé et le mode aidé. En mode aveugle, c'est l'erreur MMA qui est acheminée au processus d'égalisation. Le mode dirigé exploite l'erreur générée entre la sortie du bloc de décision et de l'égaliseur. Puis, le mode aidé représente le cas où l'on utilise une séquence d'entraînement.

En explorant les capacité de l'algorithme NFBLMS, nous avons retenu qu'il pourrait être intéressant de rendre le *pas d'adaptation* variable en temps réel pour permettre une meilleure convergence de l'égaliseur. De plus, en ce qui a trait au *pas d'adaptation*, nos simulations nous ont confirmé que les bornes doivent être inférieures ou égales à l'inverse du nombre de coefficients temporels d'égaliseur. Autrement, l'EQM n'atteint pas sont minimum et peut même diverger si μ est choisi trop fort.

Maintenant que nous connaissons la structure de l'égaliseur FBMMA3M en point flottant. Nous pouvons aborder l'étape de conception et réalisation en point fixe de l'égaliseur. C'est ce que nous présentons au chapitre quatre.

CHAPITRE 4

QUANTIFICATION ET LATENCE DE L'ÉGALISEUR FBMMA3M

4.1 Introduction

L'étape qui précède la réalisation du noyau d'égaliseur fréquentiel est la quantification. Il y a treize points à quantifier dans l'algorithme FBMMA3M. Chacun de ces points doit être analysé sous différentes conditions afin de déterminer une quantification générique et applicable pour différents contextes de télécommunications. En effet, le niveau de SNR ainsi que les coefficients du canal ne sont pas toujours les mêmes.

Nous rappelons ici que l'objectif principal de ce projet est de réaliser un égaliseur fréquentiel à espacement fractionnaire, configurable de 4 à 256 coefficients temporels pour des modulations allant de 4 à 64 QAM à des débits d'au moins 20 Msym/s. L'étape de quantification consiste donc à vérifier chaque configuration possible de l'égaliseur et de déterminer le nombre de bits d'entiers et de fractions nécessaires pour tous les cas. Pour transmettre à des débits de 20 Msym/s et plus, nous verrons également qu'il est nécessaire d'insérer un certain délai au niveau du calcul du gradient.

Dans ce chapitre, nous allons présenter notre stratégie de quantification tout en considérant l'impact de la latence sur les performances de l'égaliseur.

4.2 Stratégie de quantification de l'égaliseur FBMMA3M

Les signaux circulant dans l'algorithme FBMMA3M sont de nature complexes, aléatoires et bipolaires. Pour cette raison, nous avons choisi une représentation en complément à deux avec point fixe pour les quantifier. La variation d'amplitude des signaux qui se propagent dans l'égaliseur dépend des distorsions engendrées par le canal de transmission. En effet, le niveau de bruit blanc ainsi que le taux de variation des

coefficients du canal agissent directement sur le comportement de l'égaliseur. En conséquence, on ne peut pas déterminer la quantification exacte pour tous les contextes de communications. Dans notre démarche, nous avons choisi de quantifier l'égaliseur de manière à ce qu'il puisse être utilisé dans un contexte de télécommunication avec un trajet en vue non directe tel que le modèle SUI présenté au chapitre 3. Le rapport signal sur bruit est limité à un environnement où l'on peut transmettre à un taux d'erreur de 10^{-4}. Ainsi, les essais en simulation tiennent compte des cas où le bruit est élevé par rapport au niveau de modulation. Le tableau VIII rapporte les valeurs de SNR minimum que l'on injecte à l'égaliseur pour déterminer la plage d'entiers à configurer pour chaque point de quantification, excluant les distorsions engendrées par le canal SUI pour les modulations 4, 16 et 64 QAM.

Tableau VIII

Rapport signal sur bruit blanc inséré pour déterminer la quantification de l'égaliseur.

Modulation M-QAM	SNR (dB)
4	8
16	13
64	17

Il y a plusieurs points de quantification dans l'égaliseur qui nous concerne. Le nombre de ressources nécessaires pour transformer l'algorithme FBMMA3M en circuit dépend entre autre de la longueur des registres choisis. Il est donc important de trouver la quantification la plus juste afin de traduire fidèlement le comportement en point flottant, tout en minimisant les ressources requises.

Nous avons établi la quantification en deux étapes. D'abord, nous avons effectué une analyse théorique de l'amplitude des signaux pour chaque point de quantification. La figure 36 illustre tous les points à quantifier dans l'égaliseur. Au total, on en observe treize pour le cas où l'on contraint le gradient et onze pour le cas où l'on ne contraint pas

le gradient. En effet, dans ce dernier cas, deux modules de FFT sont éliminés dans le processus d'égalisation. Dans la littérature, plusieurs chercheurs ont traité la problématique de quantification des algorithmes adaptatifs, (Boroujeny, 1998, Bose, 2004). Deux sources d'erreur ressortent de ces analyses. La première concerne l'amplitude du signal, soit la partie entière du mot binaire. En effet, si à certaines étapes de l'algorithme, les résultats de calculs sont plus élevés que le nombre de bits disponibles pour la représentation, il y a risque de dépassement (ou *overflow* en anglais). Cette situation peut occasionner des changements drastiques dans le processus d'adaptation des coefficients. La seconde source majeure d'erreurs associées à la quantification concerne la précision de la partie fractionnaire du mot binaire. Si la représentation n'est pas assez fine, on peut se retrouver avec des performances d'égalisation amoindries par rapport à la théorie. Surtout au niveau de l'erreur quadratique moyenne. Il est donc primordial de choisir une précision adéquate.

Pour vérifier si la longueur des mots binaires et la précision de calcul de chaque point de quantification est suffisante, nous avons utilisé l'outil *Fixed Point Blockset* du logiciel Matlab. Comme point de départ, nous avons considéré que les symboles à l'entrée de l'égaliseur se situent entre -1 et 1. En effet, pour toutes les constellations M-QAM, nous proposons de normaliser la distance minimale entre chaque symbole par $2/\sqrt{M}$. Ainsi, les échantillons sans bruit sont toujours inférieurs à l'unité mais lorsqu'ils sont accompagnés de distorsions, on peut prédire que les amplitudes les plus élevées sont autour de l'unité. Ceci fait en sorte que la première transformée de Fourier (point (1) sur la figure 36) sera bornée par l'équation suivante:

$$|r_F(k)| \leq \sum_{l=0}^{2N-1} |r(n-l)| \leq 2N. \tag{4.1}$$

Par l'équation 4.1, on remarque que la quantification sera différente pour chaque longueur d'égaliseur puisque celle-ci est fonction de N, le nombre de coefficients temporels. Ainsi, comme point de référence théorique, nous avons posé la partie entière

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

de chaque mot binaire en sortie des cinq transformées de Fourier présentes dans l'égaliseur à $\log_2 2N+1$ bits. Le bit additionnel correspond au bit de signe.

Tel qu'il a été mentionné plus haut, le processus d'adaptation de l'égaliseur FBMMA3M est aléatoire et est fonction des distorsions engendrées par le canal de transmission. Il est donc difficile de déterminer théoriquement tous les points de quantification. Pour cette raison, nous avons eu recourt à notre modèle de simulation en point flottant pour déterminer les amplitudes maximales de chaque point de quantification. Ainsi, cette étape nous a permis de poser le nombre de bits d'entiers pour tous les points de quantifications de l'égaliseur. Nous avons stimulé notre modèle d'égaliseur en point flottant avec les canaux SUI-3 et SUI-5 pour une durée d'une seconde pour toutes les longueurs de coefficients tout en effectuant des essais avec différentes valeurs du *pas d'adaptation.* En simulant une seconde, on pouvait vérifier le comportement de l'algorithme sur un million de symboles. Ceci minimisait les probabilités que l'on manque des réactions particulilières de l'algorithme lorsqu'il est soumis à des variations de canal plus brusques. La vitesse de transmission a également été variée. Cette étape nous permettait de tenir compte de la longueur relative de l'égaliseur par rapport au taux de transmission. Nous soulignons également que la latence inhérente (voir plus bas dans ce chapitre) à l'architecture choisie a été considérée dans cette analyse. À l'aide d'un script de commande, les données sortant de chaque point de quantification ont été relevées. À partir de l'équation 4.2, nous avons déterminé le nombre de bits d'entiers nécessaires pour chaque configuration d'égaliseur en sélectionnant le pire cas recueilli de toutes les simulations accomplies.

$$(x)_{entier} = \lceil \log_2 \left(max\left[|\Re\{x\}|, |\Im\{x\}| \right] \right) \rceil \tag{4.2}$$

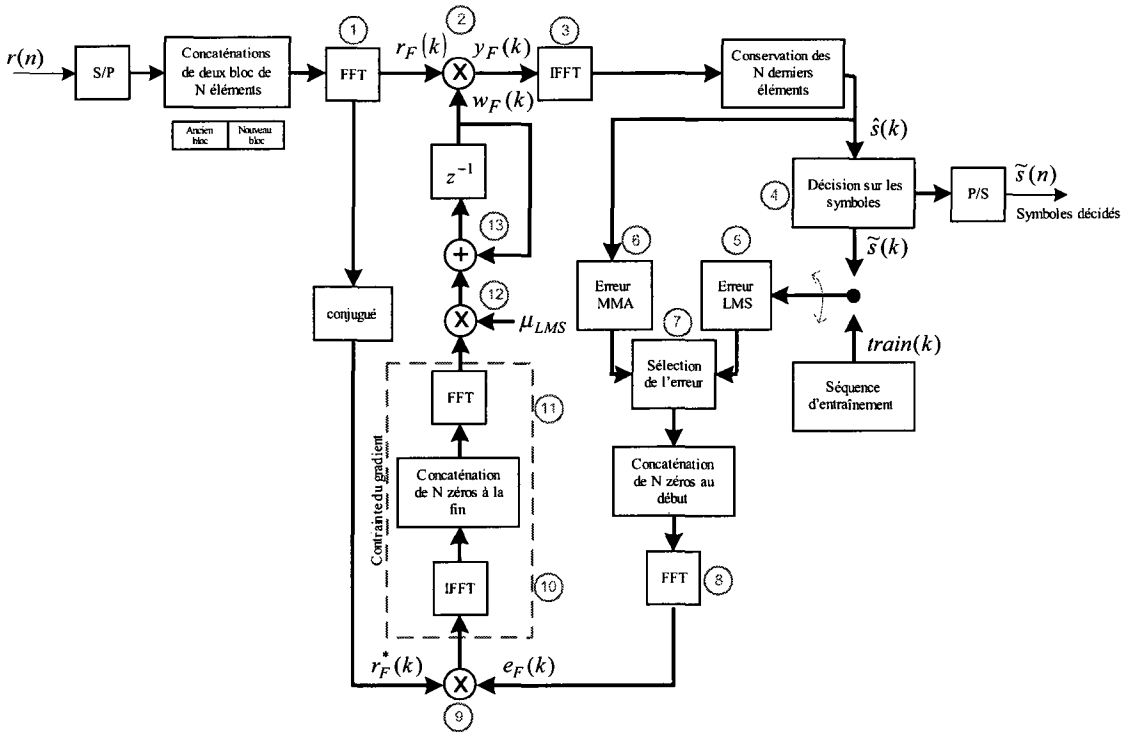


Figure 36 Points de quantification de l'algorithme FBMMA3M

Au cours des simulations dédiées à l'étape de quantification, nous avons observé que l'égaliseur tend à diverger lorsque l'on pose un *pas d'adaptation* trop élevé. Tel que nous l'avons présenté théoriquement au chapitre 2, le *pas d'adaptation* pour l'algorithme FBLMS doit être inférieur ou égal à l'inverse du nombre de coefficients temporels de l'égaliseur. On peut le poser légèrement plus grand mais c'est au risque de faire diverger les coefficients et de consommer une plus grande plage dynamique. Ainsi, pour réduire les probabilités de divergence lorsque le pas d'adaptation est trop élevé, nous avons inséré un limiteur borné à ± 2 à la sortie du point de quantification (13) afin de réduire les probabilités d'emballement des coefficients de l'égaliseur. Ce limiteur est fixé à la valeur maximale obtenue par l'équation 4.2 au point (13) pour tous les cas traités.

4.2.1 Quantification de l'erreur MMA

La fonction de l'erreur MMA présentée au deuxième chapitre est quadratique et peut tendre vers l'infini. La figure 37 illustre le comportement de cette fonction lorsque la sortie de l'égaliseur varie. Cette courbe est identique pour la partie réelle et la partie imaginaire. D'après les données accumulées en simulation, les symboles à l'entrée de l'égaliseur ne dépassent pas 1.5. Il en est de même pour la sortie de l'égaliseur. Ce n'est qu'en début du processus d'égalisation que l'on retrouve les amplitudes les plus élevées. Pour cette raison, nous avons posé un limiteur à la sortie de l'erreur MMA à ± 2. De cette manière, on limite les possibilités de divergences dues aux variations soudaines du canal de transmission. En effet, si les coefficients de l'égaliseur se mettent à grimper à la suite d'une variation subite du canal de transmission, il est possible que la sortie de l'égaliseur soit démesurément élevée pendant une courte période. La même problématique peut survenir si le *pas d'adaptation* choisi est trop élevé. Ces situations pourraient facilement occasionner une divergence irrécupérable, sachant que la fonction de l'erreur MMA dépend directement de la sortie de l'égaliseur et augmente quadratiquement vers l'infini.

Rapport-Gratuit.com

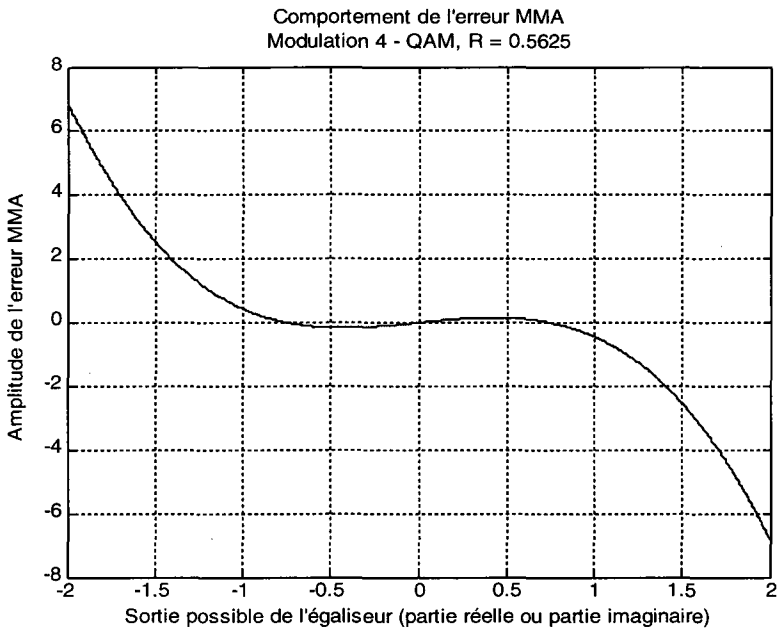


Figure 37 Erreur MMA

4.2.2 Quantification de l'erreur LMS

Au niveau de l'erreur LMS, le pire cas que l'on peut rencontrer est lorsque l'échantillon d'entraînement est directement l'opposé du symbole en sortie de l'égaliseur. Cet état engendre un résultat rarement plus grand que 2 puisque l'amplitude des symboles est normalisée à $2/\sqrt{M}$. Or, nous avons également posé un limiteur borné à ± 2 à la sortie du calcul de l'erreur LMS. Lorsque l'égaliseur est en mode dirigé, cette situation ne peut se produire puisque le bloc de décision effectue une sélection franche dans la zone où se trouve le symbole. Ainsi, en mode dirigé, l'erreur LMS est toujours inférieure à l'unité.

4.2.3 Quantification du seuil *d* pour la sélection MMA-LMS

Lors des simulations en point flottant, nous avons expérimenté plusieurs valeurs de seuil pour la sélection MMA-LMS. Le tableau IX rapporte les meilleures valeurs de *d* qui causent les taux de convergence les plus élevés pour les niveaux de modulations 4, 16 et 64 QAM. Pour une même puissance de transmission, la distance entre chaque point

d'une constellation MQAM diminue en fonction de M. Ainsi, la valeur du seuil de décision MMA-LMS (*d*) dépend du niveau de modulation M. Pour chaque niveau de modulation, il y a une certaine plage de valeur de *d* qui permet d'atteindre la convergence des coefficients. Donc, pour faciliter l'implémentation matérielle, notre choix s'est arrêté à des multiples de 2.

Tableau IX

Seuil de décision (d) entre l'algorithme MMA et LMS

Modulation M-QAM	d
4	0.5 ou 2^{-1}
16	0.125 ou 2^{-3}
64	0.0625 ou 2^{-4}

4.2.4 Impact de la précision sur la convergence de l'EQM

Les résultats obtenus au niveau de la précision de la partie fractionnaire des registres nous indiquent que la configurabilité des points de quantification est essentielle. En effet, la précision affecte directement l'erreur quadratique moyenne. Cette mesure est intimement reliée à la performance de l'égaliseur. De plus, nous savons qu'un signal transmis en 4-QAM requiert un SNR moins élevé qu'un signal transmis en 16-QAM. La même logique s'applique pour un signal transmis en 64-QAM et ainsi de suite. Donc, le nombre de bits requis au niveau de la fraction dépend du type de modulation utilisé ainsi que du taux d'erreur désiré. Pour toutes les configurations de longueur d'égaliseur, nous avons observé le même comportement. Pour des faibles constellations (4 et 16 QAM), le nombre de bit de la partie fractionnaire requis pour obtenir une EQM semblable à celle obtenue en point flottant et de l'ordre de dix. En dessous de cette valeur, des phénomènes d'instabilité apparaissent. En ce qui a trait aux modulations plus denses (64-QAM), la partie fractionnaire doit préférablement être posée à onze bits. Les figures

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

38 et 39 illustrent ce que nous venons de décrire. Pour cette simulation, nous avons élevé le SNR à 35 dB afin de bien visualiser la précision de l'égaliseur.

Cette configuration s'applique à tous les points de quantification montrés sur la figure 36 à l'exception des points (5) et (6) qui représentent respectivement l'erreur MMA et l'erreur LMS. Pour toutes les simulations accomplies, nous avons observé que la précision de la fraction ne requiert pas une granularité aussi rigoureuse que les autres points. En effet, 7 ou 8 bits fractionnaires suffisent pour obtenir une EQM semblable aux résultats du modèle en point flottant.

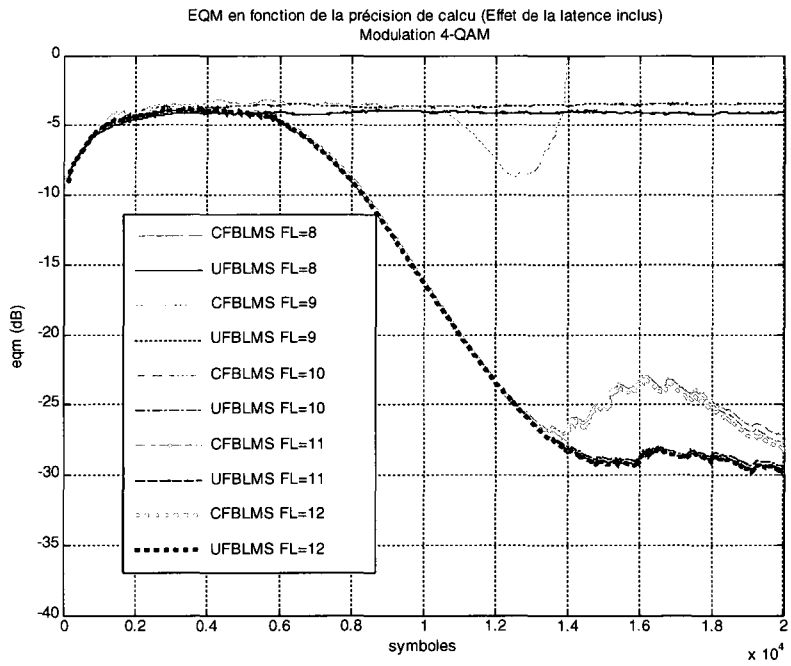


Figure 38 EQM en fonction de la précision fractionnaire pour une modulation 4-QAM

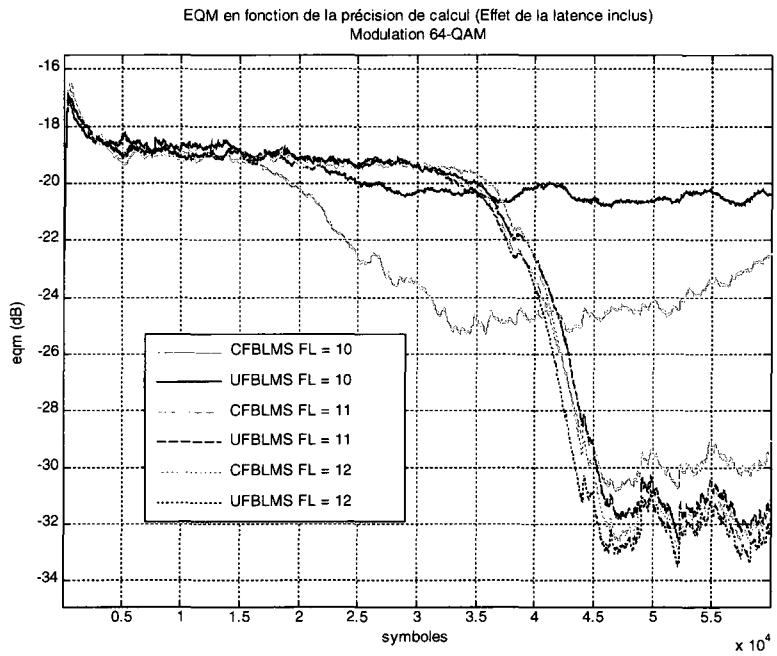


Figure 39 EQM en fonction de la précision fractionnaire pour une modulation 64-QAM

Tableau X

Détail des longueurs requises pour la partie entière des registres pour chaque point de quantification

Points de quantification	Nombre de coefficients temporels						
	4	**8**	**16**	**32**	**64**	**128**	**256**
1	3	4	5	5	7	7	7
2	4	4	5	6	7	7	8
3	2	2	2	2	2	2	2
4	N/A	N/A	N/A	N/A	N/A	N/A	N/A
5	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1
7	N/A	N/A	N/A	N/A	N/A	N/A	N/A
8	2	3	4	4	4	5	6
9	3	4	5	6	7	9	9
10	3	4	4	4	4	5	6
11	3	4	5	6	7	9	11
12	N/A	N/A	N/A	N/A	N/A	N/A	N/A
13	1	1	1	1	1	1	1

Au tableau X, on constate que le nombre de bits requis dans la partie entière augmente en fonction de la longueur de l'égaliseur. Pour les transformées de Fourier, on observe que l'équation 4.2 est une bonne référence pour quantifier ces modules. Toutefois, le fait que l'on exploite l'échantillonnage fractionnaire entraîne des petites différences par rapport à notre prévision théorique. En effet, la transformée de Fourier qui correspond au point (8) sur la figure 36 nécessite moins d'amplitude au niveau des entiers puisqu'on insère un zéro entre chaque symbole. La sortie du filtre (point (3) sur la figure 36) ne dépasse généralement pas 1 sauf en début de processus d'égalisation ou si le *pas d'adaptation* est choisi trop fort. C'est pourquoi la quantification est la même pour toutes les longueurs d'égaliseur. Tel qu'expliqué plus haut dans ce chapitre, l'erreur LMS ainsi que l'erreur MMA ont été bornées à ± 2 en simulation. Ainsi, nous avons choisi de poser un seul bit d'entier puisqu'avec huit bits de fraction, les valeurs en sorties

peuvent osciller entre ± 1.9961. Il en est de même pour la quantification des coefficients de l'égaliseur. Avec dix bits de fractions, il peuvent varier entre ± 1.999 avec une granularité de 0.0097656. Le point de quantification (4) représente le bloc de décision. Afin de représenter les symboles en sortie de l'égaliseur, nous les quantifions sur six bits. Soit cinq bits de fraction et un bit de signe. La quantification du module de sélection de l'erreur est présentée au tableau IX. Puis, le *pas d'adaptation*, (point (12)) est accompli par un décalage vers la gauche du point de la fraction. En effet, les valeurs de μ_{LMS} doivent être des multiples de deux. Cette stratégie permet d'économiser deux multiplicateurs réels au niveau de l'implémentation matérielle.

4.3 Latence de l'algorithme FBMMA3M

Afin de réaliser l'égaliseur FBMMA3M sous forme matérielle, nous avons dû effectuer quelques modifications à l'algorithme. En effet, dans son traitement, cet égaliseur requiert le calcul de 3 à 5 FFT/IFFT par bloc de N symboles, dépendamment que l'on contraigne ou non le gradient.

4.3.1 Complexité de l'algorithme FBMMA3M sans latence

Comme nous le savons, le traitement doit être accompli en temps réel. Or, pour y arriver sans latence, il faudrait que la fréquence de l'horloge principale de l'égaliseur soit au moins de quinze à vingt fois plus élevée que le taux de transmission des symboles. En effet, si on suit le cheminement des données montré à la figure 36, la première FFT/IFFT (point (1) sur la figure 36) doit procéder deux fois plus rapidement que le taux de symboles puis un tampon doit accumuler les données en sortie jusqu'à ce que les coefficients aient été rafraîchis. Ce même raisonnement s'applique aux trois autres FFT/IFFT, de sorte que la dernière opération de transformée de Fourier requiert un taux de traitement cinq fois plus rapide que le taux de symbole avec une mémoire tampon d'au moins $5N$ cases. Pour appliquer intégralement l'algorithme FBMMA3M, il est impératif que les signaux aient le temps de passer par toutes les opérations de ce dernier

avant de permettre l'échantillonnage d'un nouveau bloc de symboles. Une telle implémentation requiert une quantité de bloc mémoire et une stratégie de contrôle pour le moins complexe. Ceci bien sûr sans compter que la technique de sauvegarde et chevauchement nécessite un traitement deux fois plus rapide. Il en est de même pour l'application du principe d'échantillonnage fractionnaire. Si l'on regroupe toutes les contraintes mentionnées dans ce présent paragraphe, on constate que l'égaliseur doit fonctionner avec une horloge au moins douze à vingt fois plus rapide que le taux de symboles transmis. Pour figurer ce que cela représente, considérons un module de FFT/IFFT fonctionnant à une fréquence maximale de 200 MHz[4]. Si l'on souhaitait implémenter l'algorithme FBMMA3M intégralement, le taux de symboles ne devrait pas dépasser 1 Msym/s. Ce qui est bien loin des objectifs de 20 Msym/s établis en tout début de projet.

4.3.2 Effet de la latence sur l'égaliseur FBMMA3M

Nous avons donc choisi de modifier légèrement l'équation d'adaptation des coefficients pour éviter de quintupler le taux de traitement de l'égaliseur. Nous nous sommes basés sur le fait que le canal de transmission ne varie pas aussi rapidement que le taux de traitement de l'égaliseur. En effet, la variation de Doppler d'un canal SUI se situe entre 0 et 4 Hz. Le taux de traitement de l'égaliseur est de l'ordre des MHz. On peut donc présumer que les coefficients du canal sont quasi statiques aux yeux de l'égaliseur. Cette remarque nous laisse donc l'opportunité de retarder le calcul du gradient dans l'équation d'adaptation des coefficients de l'égaliseur. L'équation 4.3 exprime ce que nous avons modifié.

$$\boldsymbol{w}_F(k+1)=\boldsymbol{w}_F(k)+\mu_{LMS}\left[\boldsymbol{r}_F{}^*(k-D)\otimes\boldsymbol{e}(k-D)\right]_C \tag{4.3}$$

4 Fréquence maximale du module de transformée de Fourier R22PC lorsque la quantification des multiplicateurs requière 18 bits ou moins. Autrement, on observe que la fréquence maximale de traitement du module R22PC est de l'ordre de 150 Mhz.

D indique le délai en bloc de N échantillons. Dans (Proakis, 1989), il a été démontré qu'il est possible d'insérer un délai au niveau du calcul du gradient tout en conservant sensiblement les mêmes propriétés de convergences La stabilité et le taux de convergence sont affectés et dépendent de la grandeur de D. Les bornes du *pas d'adaptation* sont réduites. Ainsi, pour assurer la convergence ainsi que la stabilité en régime permanent, on propose de réduire le *pas d'adaptation*. L'article en question présente des résultats pour un algorithme basé sur la méthode LMS temporelle. Tel que nous l'avons présenté au chapitre 2, cette technique implique qu'à chaque symbole, on doit rafraîchir les coefficients. En ce qui nous concerne, l'adaptation est appliquée à chaque bloc de N échantillons. Un délai de D blocs correspond dans les faits à un délai de DN échantillons. On constate donc que plus l'égaliseur est long, plus l'impact du délai est prononcé. Pour notre application, chaque module de transformée de Fourier correspond à un délai de D blocs de N échantillons. Ainsi, lorsque nous appliquons l'algorithme CFBLMS, le délai est de cinq blocs. Puis pour l'algorithme UFBLMS, le délai est de trois blocs.

Ainsi, l'EQM est donc susceptible d'osciller davantage en régime permanent lorsqu'on contraint le gradient puisque le délai est plus élevé. C'est effectivement ce que nous constatons sur la figure 38 et 40. Pour illustrer l'impact du délai sur les performances de l'égaliseur, nous avons repris la simulation de la figure 29 présentée au troisième chapitre en ajoutant les délais à la suite du point de quantification (12) de la figure 36.

Si on compare les courbes de la figure 29 à celles de la figure 40, on remarque que le taux de convergence a légèrement diminué avec l'insertion des délais. Rappelons que cette simulation a été effectuée en mode MMA-LMS, sans séquence d'entraînement avec un canal SUI-5, un égaliseur de 128 coefficients temporels, un débit de 1 Msym/s et un *pas d'adaptation* $\mu_{LMS}=2^{-10}$. On remarque qu'il faut environ 5000 échantillons de plus pour atteindre le régime permanent lorsqu'on insère les délais. De plus, il faut réduire le *pas d'adaptation* pour atteindre la convergence lorsque on transmet en 64-QAM et que le gradient n'est pas contraint. On observe également que l'EQM oscille davantage en régime permanent lorsqu'on insère les délais.

On confirme toutefois que l'insertion des délais n'affecte pas significativement les performances de l'égaliseur FBMMA3M. Malgré que le taux de convergence soit réduit les avantages qui nous ont incités à aborder l'étude de l'égalisation fréquentielle sont toujours présents. En effet, nous avons montré que l'égaliseur FBMMA3M converge avec un long filtre même lorsqu'on insère un délai au niveau du gradient. De plus, cet ajout nous permet de réduire la complexité d'implémentation de l'égaliseur.

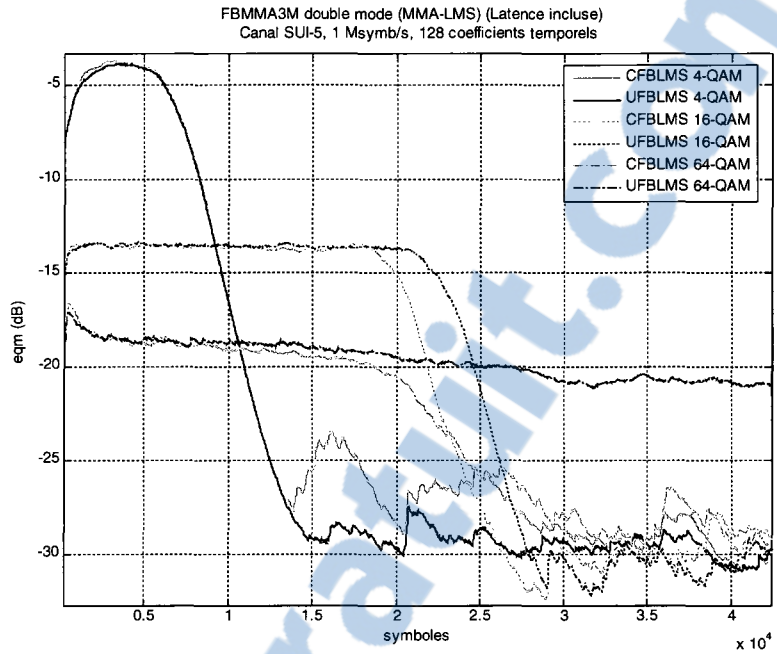


Figure 40 Comportement de l'EQM lorsque le gradient est retardé ($\mu_{LMS}=2^{-10}$)

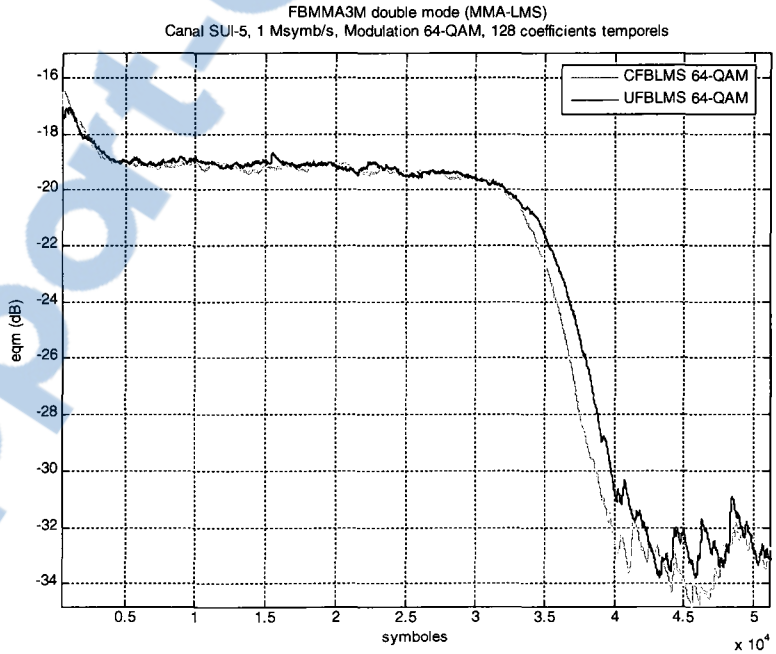


Figure 41 Comportement de l'EQM lorsque le gradient est retardé ($\mu_{LMS}=2^{-11}$)

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

4.4 Conclusion

La stratégie de quantification de l'égaliseur FBMMA3M ainsi qu'une version modifiée de l'adaptation des coefficients ont été abordés dans ce chapitre. Par le biais de simulations, nous avons déterminé les longueurs de registres requises pour chaque point de quantification. Les résultats ont montré qu'il faut au moins dix bits de fractions pour reproduire efficacement le comportement théorique de l'égaliseur. Et plus on augmente le niveau de modulation, plus la quantification doit être fine. Avec une modulation 64-QAM, il faut au moins onze bits de fraction.

Afin de répondre aux contraintes de taux de symboles fixées en tout début de projet, nous avons dû insérer un délai au niveau du calcul du gradient. Les résultats de simulations ont prouvé que cette modification permet d'obtenir une EQM semblable à la théorie en régime permanent. Cependant, l'ajout du délai provoque une réduction du taux de convergence de l'EQM. De plus, le *pas d'adaptation* doit être réduit pour éviter l'instabilité en régime permanent. L'avantage de cette modification nous permet cependant de quintupler la vitesse de l'horloge de l'égaliseur et également de réduire considérablement la quantité de mémoire tampon nécessaire au traitement.

CHAPITRE 5

RÉALISATION DE L'ÉGALISEUR FBMMA3M

5.1 Introduction

Les étapes de conception et de réalisation de l'algorithme FBMMA3M sont les suivantes:

1. Comprendre le principe d'égalisation dans le domaine temporel et fréquentiel.
2. Développer un modèle de simulation et déterminer la structure de l'algorithme à implémenter en point flottant.
3. Investiguer sur l'implémentabilité matérielle de l'égaliseur et déterminer la topologie finale à implémenter.
4. Revérifier et confirmer la fonctionnalité de la structure choisie en point flottant
5. Développer un modèle en point fixe et déterminer la quantification de chaque opération de sorte que la précision de calcul soit fidèle à l'algorithme en point flottant.
6. Transcoder l'algorithme dans un langage de description matérielle (VHDL).
7. Vérifier et valider le comportement en comparant avec le modèle développé en point flottant.
8. Effectuer le placement et le routage du code fonctionnel.
9. Vérifier et tester en laboratoire (Étape reportée à des travaux futurs, faute de temps).

Les quatre premiers chapitres ont été consacrés à la présentation des étapes 1 à 5. Le présent chapitre traite principalement des étapes 6, 7 et 8. On y expose les stratégies d'implémentation matérielle de l'égaliseur FBMMA3M. Nous expliquons également les méthodes de vérification comportementale de l'algorithme. La méthodologie employée a été appliquée pour toutes les configurations possibles de l'égaliseur, soit un nombre de 4 à 256 coefficients temporels, une gestion de l'erreur et des symboles capable d'accommoder des modulations de 4 à 64 QAM et un traitement des données au delà de 80 MHz pour permettre des taux de symboles allant jusqu'à 20 Msym/s, sachant qu'on

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

accomplit un échantillonnage fractionnaire à Tsym/2 et qu'on utilise la méthode de chevauchement et sauvegarde à cinquante pour cent.

5.2 Stratégie d'implémentation et de vérification

L'objectif final de ce travail de maîtrise est de développer un noyau d'égaliseur dans le domaine des fréquences facilement intégrable dans un système tel qu'un modem ou une radio. Les principaux objectifs à atteindre étaient les suivants:

1. Égaliseur configurable pré synthèse pour 4 à 256 coefficients temporels
2. Taux de symboles d'au moins 20 Msym/s.
3. Modulation 4-16-64-QAM.
4. Compromis entre vitesse de traitement et espace matérielle requise.

Pour atteindre nos objectifs de taux de transmission, nous avons choisi d'implémenter l'égaliseur dans une puce programmable FPGA. Chaque sous module de l'égaliseur est muni d'un signal d'activation (CE) pour permettre l'utilisation d'une horloge différente de la principale. Au niveau de l'écriture du code comportementale de l'égaliseur, nous avons opté pour une méthode universelle. Aucune fonction spécifique à une technologie (BRAM, BMULT, FIFO) n'a été imposée dans le code. De cette manière, l'égaliseur peut être implémenté dans de multiples plateformes (Xilinx, Altera) avec un minmum d'ajustements.

5.2.1 VHDL

La matérialisation de l'égaliseur est accomplie avec le langage VHDL (Very high speed integrated circuit High Description Language). Ce langage permet d'effectuer la description du système sous forme syntaxique, (VHDL tutorial, 2005). Cette description est évaluée par un compilateur qui traduit les lignes de codes structurées en circuits électroniques. Le code développé a été pensé de manière à ce que le noyau soit synthétisable et multi plate forme. En effet, la logique codée peut être synthétisée dans plusieurs technologies. Ainsi, les fonctions spécifiques à une compagnie tels que les

mémoires RAM ou les unités DSP n'ont pas été imposés dans le code. Les paramètres de configurations (voir plus bas) doivent être pensés et fixés avant la synthèse par l'usager de l'égaliseur. Le code est écrit de manière à limiter la dispersion (fanout) des signaux. Chaque module est synchrone avec l'horloge.

5.2.2 Implémentation

La procédure de conception de l'égaliseur fréquentiel FBMMA3M est illustrée à la figure 42. Les outils de conception utilisés sont les suivants: Matlab 7.1, Simulink, HDL designer 2003, Modelsim 5.7, Synplify Pro 7.7.1 et Xilinx ISE foundation 6.3. L'architecture en point flottant et en point fixe de l'égaliseur a été réalisée à l'aide des outils DSP blockset, Communications blockset et fixed point blockset de Matlab/Simulink. Lorsque l'architecture en point fixe a été validée et que nous avons atteint la précision désirée, nous avons effectué le transfert de l'architecture en langage VHDL entièrement à la main avec le logiciel HDL designer. Le comportement du code VHDL a été simulé à l'aide du logiciel Modelsim puis les résultats ont été comparés échantillons à échantillons avec le modèle Matlab/Simulink. Après que toutes les configurations possibles ont été vérifiées, le code VHDL fut synthétisé avec l'outil Synplify Pro. Afin de nous assurer la synthétisabilité du noyau d'égaliseur fréquentiel, nous avons accompli le placement routage du code VHDL à l'aide de Xilinx Foundation. Nous avons choisi d'utiliser le Virtex 2 Pro (XCV2P40-7) pour vérifier la vitesse de traitement maximale ainsi que l'espace requise en terme de BMULT, de BRAM et de *slices*.

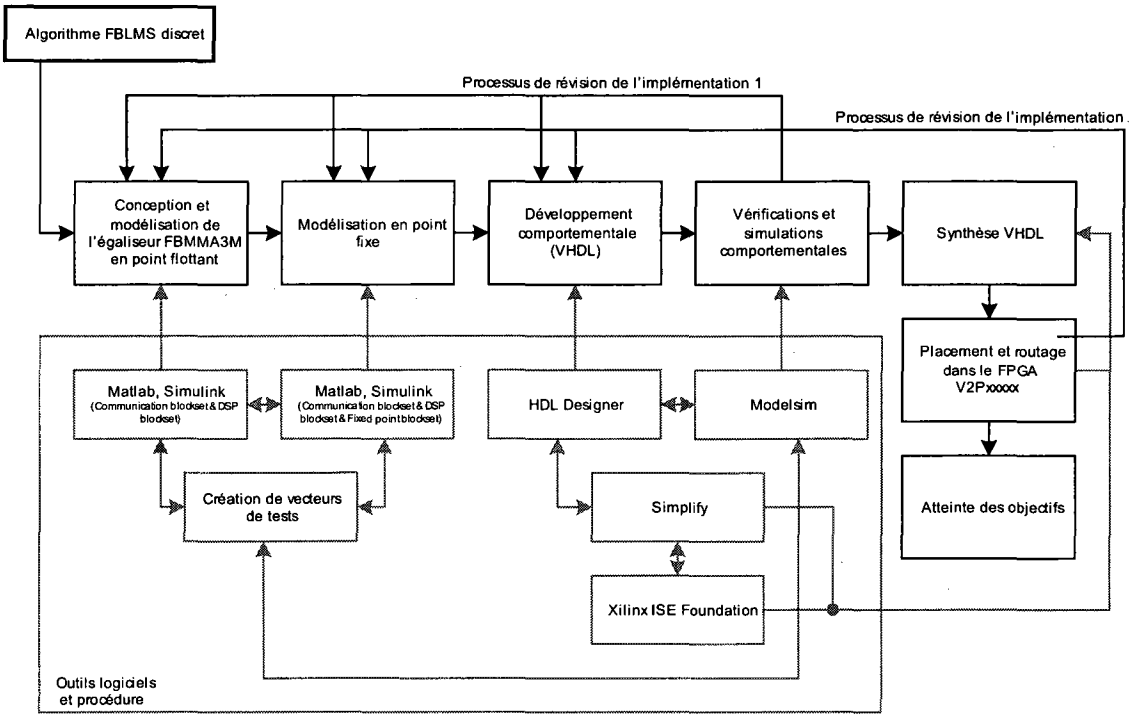


Figure 42 Diagramme bloc du processus d'implémentation et outils de CAO

Afin de réduire les ressources matérielles tout en augmentant le taux de traitement à son maximum, nous avons opté pour un traitement entièrement sériel. Pour y arriver, nous avons utilisé un module de FFT/IFFT déjà existant, fonctionnant entièrement de manière sérielle. Cette stratégie nous permet d'éviter les techniques de réutilisation de ressources. En effet, en accomplissant un traitement sériel, un seul multiplicateur complexe est requis pour le calcul de la corrélation fréquentiel (filtrage) et de l'inter corrélation (calcul du gradient). De plus, avec un traitement sériel, l'usage des blocs mémoires est réduit à son minimum.

5.2.3 Vérification pré synthèse

Une des difficultés rencontrées lors du développement matériel de l'égaliseur est l'initialisation. En effet, les données doivent être acheminées aux différents modules aux bons instants. Nous avons développé une stratégie de déclenchement combinatoire. C'est à dire que chaque module initialise le suivant.

La figure 43 illustre la méthodologie empruntée pour avancer le développement de l'égaliseur. Nous avons démarré avec le premier module de la chaîne de traitement, soit le module OSB. Après validation des formes d'ondes dans Modelsim, les résultats en simulation comportementale ont été comparés avec ceux en point flottant. Nous avons choisi de comparer le modèle VHDL avec le simulateur en point flottant car la quantification des multiplicateurs complèxes disponibles dans Simulink (Version Matlab 7.1) n'est pas configurable de la même manière que les multiplicateurs que nous avons implémentés en VHDL. En effet, la quantification des bits de la partie entière ainsi que celle des bits de la partie fractionnaire à la sortie du premier étage de la multiplication complexe n'est pas accessibles dans Matlab 7.1. Ceci étant dit, le comportement globale du simulateur en point fixe n'est pas identique à celui du modèle VHDL. Il y a des différences mineures de quantification. Le modèle en point flottant nous permet ainsi de confirmer notre choix de quantification lorsqu'on mesure le SQNR.

Pour vérifier chaque sous module, nous injections un vecteur de 50000 symboles provenant directement de la sortie du canal SUI (voir figure 23). Le script de vérification Matlab a été conçu pour détecter les erreurs de deux manières. La première méthode consistait à comparer le SQNR à un seuil acceptable. Pour une quantification de 10 bits fractionnaires, ce seuil était posé à $-20\log(1-\sum_{i=-10}^{-1} 2^{i})$, soit 60 dB. Advenant un résultat inférieur à ce seuil, un signal d'erreur était déclenché puis nous procédions aux modifications comportementales associées. La deuxième méthode

consistait en une vérification visuelle. Les résultats en points flottants étaient superposés aux résultats du modèle VHDL sur un graphique. À l'oeil, il nous était possible de distinguer les erreurs et également, l'instant exacte où elle se produisait. Les figures 44 et 45 illustrent respectivement les résultats en sortie du module *OSB* et du module *wfifo* (voir figure 46), comparés aux résultats du modèle en point flottant. Sur la figure 45, on observe que certains points du noyau VHDL sont légèrement distancés par rapport à ceux du modèle en point fixe. La raison étant que la quantification près de zéro n'est pas assez précises. Toutefois, le SQNR demeure inférieur à 60 dB. Nos simulations montrent que lorsque l'on respecte ce seuil, les propriétées de convergences ne sont pas affectées.

Au niveau du filtrage, (voir point 2 de la figure 36), nous avons dû alimenter l'entrée (d*w_re, dw_im*, figure 46) du module *s_cplx_mult1* avec les résultats en point flottant. En fonctionnant en boucle ouverte, il nous était possible de vérifier la fonctionnalité ainsi que la synchronisation de tous les blocs jusqu'à la sortie du module *wfifo*. Lorsque toutes les configurations établies ont été validées, nous avons fermé la boucle et observé le comportement de l'erreur quadratique moyenne estimée avec celle obtenue en sortie avec le modèle théorique. La même vérification a été faite au niveau de la constellation en sortie de l'égaliseur. Quelques résultats en boucle fermées sont présentés en fin de chapitre.

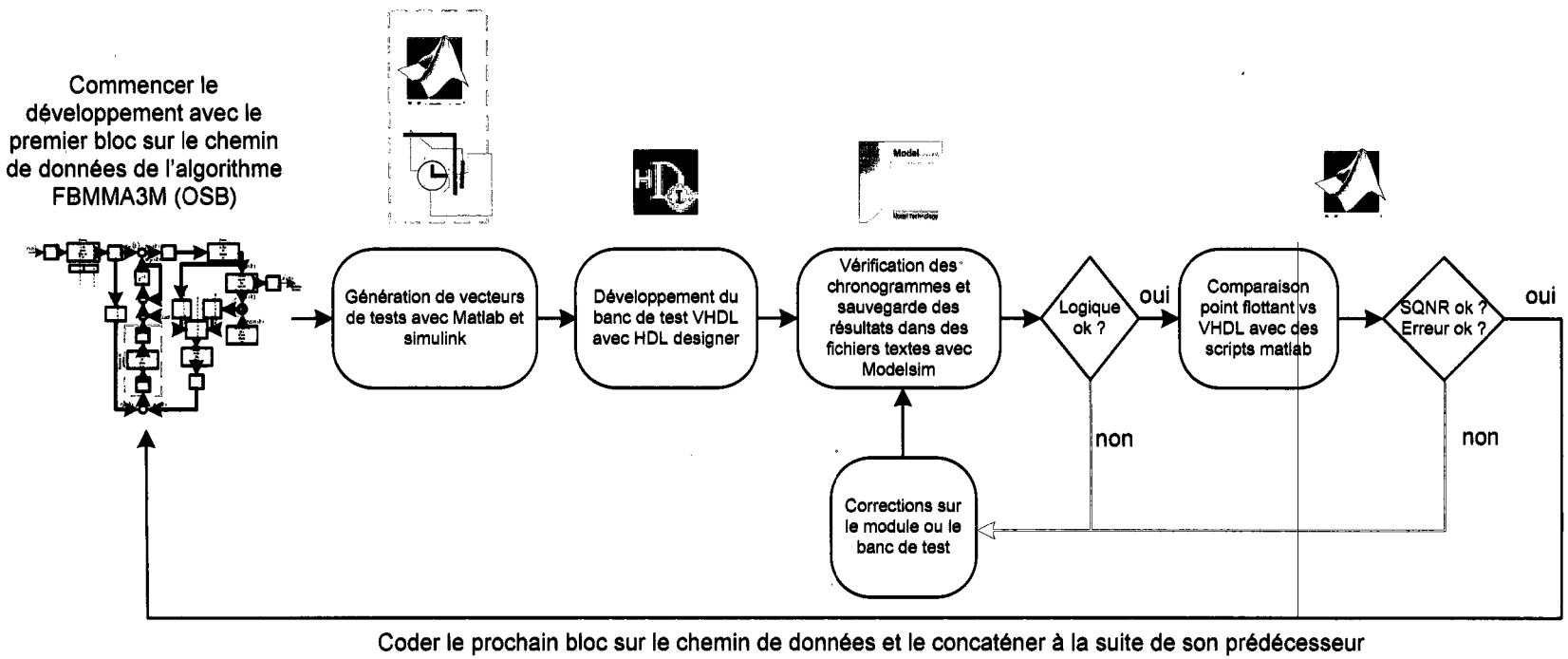


Figure 43 Diagramme bloc du processus d'implémentation et outils de CAO

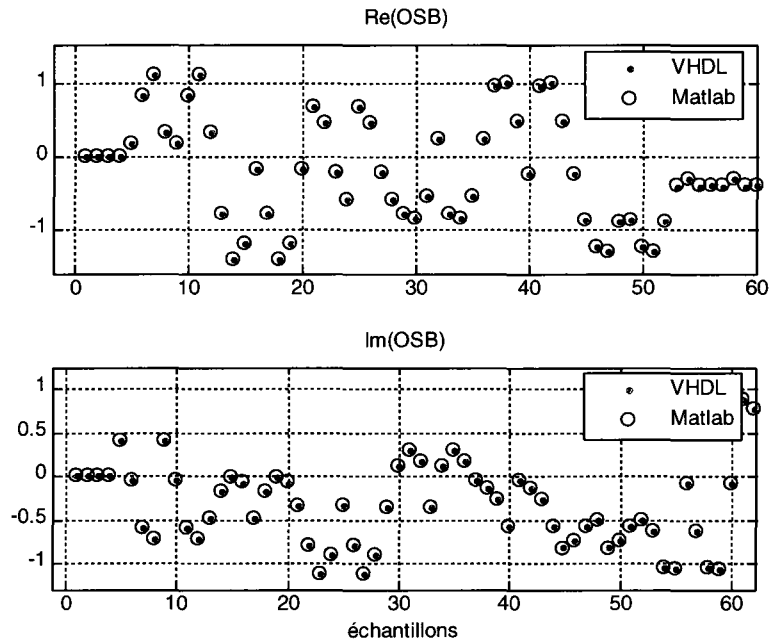


Figure 44 Exemple de courbe de vérification point flottant vs VHDL pour le module *OSB*.

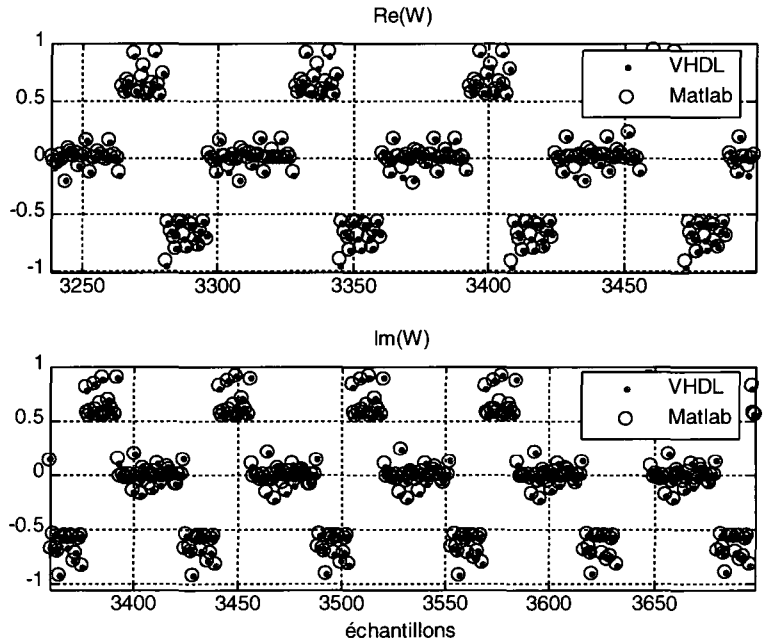


Figure 45 Exemple de courbe de vérification point flottant vs VHDL pour le module *update_coeffs*.

5.3 Description des modules implémentés

Le schéma bloc d'implémentation matérielle de l'égaliseur FBMMA3M est illustré à la figure 46. Les symboles corrompus à l'entrée sont marqués en gras et se distinguent en deux parties: réelles et imaginaires. Ils entrent d'abord dans le module *OSB* de manière sérielle, puis ils sont remaniés et disposés en deux blocs de *N* échantillons qui accomplissent le chevauchement et sauvegarde à 50%. Le signal d'activation, *ce*, doit être cadencé à 4*Tsym/s*. La totalité du traitement d'égalisation fréquentielle s'effectue de manière sérielle. La plus grande difficulté rencontrée lors de l'implémentation de l'égaliseur a été l'initialisation et la synchronisation des modules. Pour chaque configuration, soit de 4 à 256 coefficients temporels en mode contraint ou non contraint, cette opération est transparente au yeux de l'usager de ce module. Nous avons employé une stratégie combinatoire. C'est à dire que chaque module initialise le suivant. Le détail des différents modules est présenté dans les sous sections qui suivent.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

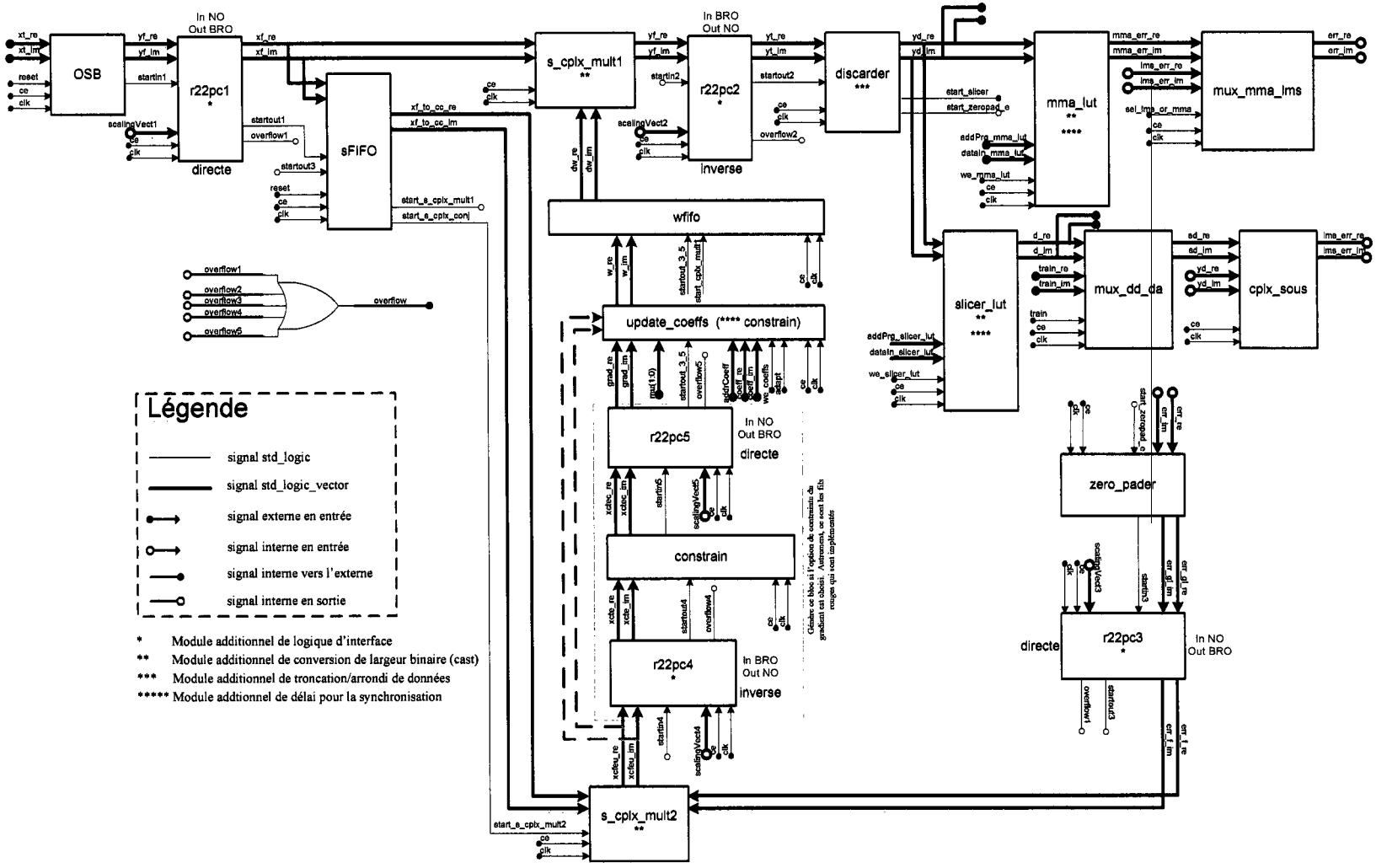


Figure 46 Schéma d'implémentation de l'égaliseur FBMMA3M

5.3.1 Module de chevauchement et sauvegarde

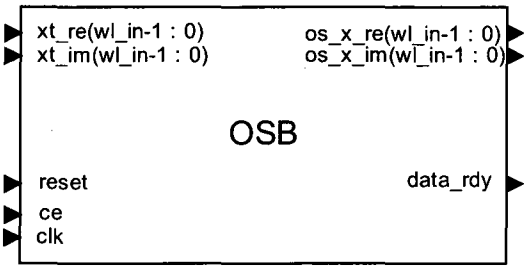


Figure 47 Module de chevauchement et sauvegarde.

Génériques

wl_in	integer	Indique le nombre de bits alloués pour un échantillon en entrée.
log2n	positive	Spécifie le logarithme en base deux associé au nombre de coefficients fréquentiels de l'égaliseur.
ntaps	positive	Spécifie le nombre de coefficients temporels de l'égaliseur.

Signaux en entrée

clk	Horloge.
ce	Activation du module.
reset	Réinitialisation de la MEF.
xt_re	Partie réelle du symbole en entrée.
xt_im	Partie imaginaire du symbole en entrée.

Signaux en sortie

os_x_re	Partie réelle des échantillons en sortie.
os_x_im	Partie imaginaire des échantillons en sortie.
data_rdy	Indique que le prochain échantillon en sortie est valide.

Le module de chevauchement et sauvegarde est présenté à la figure 47 sous l'acronyme OSB (Overlap Save Buffer). L'entrée et la sortie des données sont de nature sérielle et le

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

taux de traitement est 4xTauxSymboles. La temporisation des symboles est contrôlée à l'aide d'une machine à dix états. Quatre blocs de mémoires complexes (partie réelle et partie imaginaire du symbole) sont nécessaires à l'implémentation. Donc en matériel, huit blocs de mémoires volatiles à *N* cases sont utilisés. Le tableau XI illustre la succession des étapes de la machine à états finis qui contrôle le flot des données dans ce module pour le cas où on configure l'égaliseur avec un seul coefficient. Notez que nous n'avons pas implémenté l'option à un seul coefficient. Toutefois, ce cas particulier simplifie la présentation de la MEF. De plus, le raisonnement s'applique pour toutes valeurs de *N*.

Tableau XI

Succession des états de la MEF du bloc de chevauchement et sauvegarde

Entrée	**Sortie**	**Tampon (entr.)**	**Tampon (sort.)**	**État**
1	0	T1	X	INIT1
	0			
2	0	T2		INIT2
	1		T1	E1
3	1	T3		E2
	2		T2	E3
4	2	T4		E4
	3		T3	E5
5	3	T1		E6
	4		T4	E7
6	4	T2		E8
	5		T1	E1

Ainsi qu'on peut le remarquer au tableau XI, les échantillons en sortie sortent deux fois plus rapidement qu'ils entrent en raison du chevauchement à 50 %. Aux états initiaux, on transmet des zéros en sortie pendant que les tampons se remplissent. Lorsqu'un tampon est en entrée, un autre doit être disponible en sortie. C'est le principe de mémoire à double port, (Xilinx, 2005). Chaque tampon est monopolisé en entrée ou en

sortie pendant $2N$ coups d'horloge. Donc, afin de ne pas écraser d'information, quatre blocs de mémoires complexes sont requis. Après l'initialisation, les huit états (E1 à E8) se répètent cycliquement.

5.3.2 Choix du module de FFT

Le module de FFT/IFFT accomplit une fonction très importante dans l'égaliseur FBMMA3M. En effet, on a recourt à ce module à cinq reprises (voir figure 46) dans le processus d'égalisation. L'implémentation et la vérification fonctionnelle d'un module de FFT/IFFT peuvent constituer une charge de travail de l'ampleur d'un projet de maîtrise, (Grandmaison, 2005). Ceci est d'autant plus vrai lorsqu'il est requis, tel que dans ce projet, de pouvoir configurer le nombre de points de la FFT. Pour cette raison, nous avons choisi d'utiliser un noyau de FFT/IFFT déjà existant. Sur le marché, il existe une grande variété de noyau à propriété intellectuelle accomplissant la fonction de FFT/IFFT. La plupart sont basées sur l'algorithme *divide and conquer* de Cooley-Tukey, (Proakis & Manolakis, 1996). Le nombre d'opérations papillons (voir chapitre 2) dépend du nombre de points de la FFT. En général, ces algorithmes sont optimisés pour N logarithmes de 2 ou de 4.

Dans un modem, tel qu'il a été abordé au tout début de ce mémoire, le processus d'adaptation doit être accompli en temps réel. Si un nouveau symbole est disponible à chaque cycle à l'entrée de l'égaliseur, une donnée doit être également valide en sortie. Cependant, l'opération de transformée de Fourier requiert un bloc de N données avant d'être accomplie. Il y a deux tendances dominantes pour l'implémentation matérielle des modules de FFT/IFFT fonctionnant en temps réel. La technique parallèle et la technique pipelinée. Les critères principaux pour la sélection du module de FFT/IFFT sont la vitesse de traitement, la latence, la complexité matérielle et la configurabilité.

La technique parallèle est souvent conçue avec un seul papillon *radix-4*. Ainsi, seules les transformées de puissance 4 sont possibles. Cette méthode requiert trois tampons

doubles ports de grandeur N. Un pour paralléliser l'entrée, un pour le traitement et un autre pour la sortie. Ceci implique que la latence est de l'ordre de $3Nf$ cycles. Où Nf représente $2N$ pour notre application puisque nous utilisons la méthode de chevauchement et sauvegarde à 50%. Nous rappelons que N représente le nombre de coefficients temporels de l'égaliseur.

La technique pipelinée utilise $\log_{2\ ou\ 4} Nf$ papillons interreliés sériellement. Il n'y a donc pas de parallélisation à accomplir et les données entrent et sortent du module de manière sérielle. Ainsi, la mémoire requise ainsi que la latence sont de l'ordre de Nf.

En suivant les critères de sélection mentionnés plus haut, notre choix d'architecture s'est arrêté sur la méthode pipelinée basée sur l'algorithme $R2^2PC$ qui a été réalisé et exhaustivement analysé dans (Grandmaison, 2005). Dans son mémoire, Mme Grandmaison a effectuée une analyse des différents noyaux de FFT/IFFT disponibles sur le marché. Au moment où nous avons accompli ce projet de maîtrise, le module $R2^2PC$ était muni des meilleures propriétés sur le marché.

5.3.3 Modules r22pcx

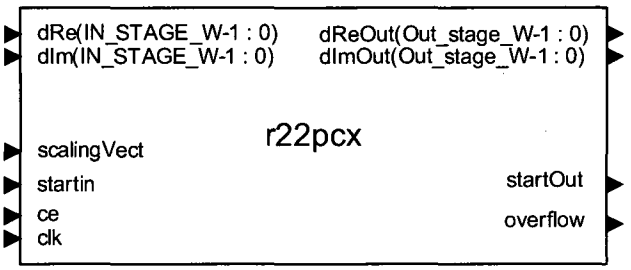


Figure 48 Module de transformée de Fourier directe et inverse.

Génériques

in_fft_w	positive	Largeur binaire du bus en entrée (partie réelle et imaginaire).

out_fft_w	positive	Largeur binaire du bus en sortie (partie réelle et imaginaire).
tf_w	positive	Largeur binaire des facteurs de phases.
log2n	positive	Logarithme base 2 du nombre de coefficients fréquentiels. ($2N$)
scaling	positive	Indique la mise à l'échelle à réaliser.
inverse	boolean	Indique la sélection d'une transformée directe ou inverse. (vrai pour naturel et faux pour inverse)
bit_rev_in	boolean	Indique si les échantillons en entrée sont en ordre naturel ou inverse. (vrai pour naturel et faux pour inverse).
w	width_typ	Indique la largeur des étages intermédiaires dans le processus de calcul de la transformée.

Signaux en entrée

clk	Horloge.
ce	Activation du module.
startin	Réinitialisation des compteurs.
scalingVect	Mise à l'échelle des données en sortie.
dRe	Partie réelle de l'échantillon en entrée.
dIm	Partie imaginaire de l'échantillon en entrée.

Signaux en sortie

startout	Drapeau indiquant l'arrivée du premier résultat valide
overflow	Drapeau indiquant un débordement survenu pendant le processus de calcul.
dReOut	Partie réelle des échantillons en sortie
dImOut	Partie imaginaire des échantillons en sortie

Le module $R2^2PC$ a été conçu dans le cadre d'un autre projet de maîtrise, (Grandmaison, 2005). Il permet d'effectuer des transformées de Fourier directes (FFT) et inverses (IFFT) pour des blocs de 4 à 4096 points. Ce noyau est hautement configurable. De plus, la structure interne du noyau $R2^2PC$ a été pensée en terme de portabilité. En

effet, il est possible d'implémenter ce module dans plusieurs plate formes (Ex.: Altera, Xilinx) puisque seuls les blocs dédiés de multiplications 18x18 ainsi que les blocs de mémoires ont été imposés et en général, ces dispositifs sont disponibles dans la plupart des puces programmables. En ce qui concerne la réalisation du noyau d'égaliseur fréquentiel, cinq blocs de transformées de Fourier sont requis. Le nombre de points de chaque FFT/IFFT, *Nf*, correspond au nombre de coefficients fréquentiels de l'égaliseur, soit 2*N*.

5.3.3.1 Ordonnancement

La variable *bit_rev_in* permet de configurer l'ordonnancement du module $R2^2PC$. Dans la littérature, la plupart des algorithmes de FFT/IFFT sont de type DIF (décimation in frequency) et/ou DIT (decimation in time), (Proakis & Manolakis, 1998). Ce principe est appliqué dans la structure de l'algorithme $R2^2PC$. Ainsi, lorsque l'information à l'entrée est transmise en ordre naturel NO (natural order), le résultat ressort en ordre inverse BRO (bit reverse order). Le tableau XII illustre le concept d'ordonnancement pour une transformée de 8 points. L'ordre des échantillons est présenté en binaire pour bien montrer le principe de l'ordre inversé. Par exemple, pour une entrée en ordre naturel, le deuxième échantillon (001) se retrouvera en sortie à la cinquième position et le cinquième point (100) en entrée sera positionné au deuxième rang en sortie. L'ordre du mot binaire associé à la position de l'échantillon en entrée est inversé.

Tableau XII

Illustration du concept d'ordonnancement pour une transformée de Fourier de 8 points

Entrée Naturelle	**Sortie Inversée**		**Entrée Inversée**	**Sortie Naturelle**
000	000		000	000
001	100		100	001
010	010		010	010
011	110		110	011

Entrée Naturelle	Sortie Inversée
100	001
101	101
110	011
111	111

Entrée Inversée	Sortie Naturelle
001	100
101	101
011	110
111	111

Rappelons que tout le traitement dans l'égaliseur est accompli de manière sérielle. Il faut donc s'assurer qu'à chaque opération, la rencontre des différents signaux soit dans le bon ordre. Pour ce faire, nous avons configuré l'ordonnancement comme suit:

Tableau XIII

Configuration de l'ordonnancement des cinq transformées de Fourier dans l'égaliseur FBMMA3M

r22pc1		r22pc2		r22pc3		r22pc4		r22pc5	
Entrée	Sortie	Entrée	Sortie	Entrée	Sortie	Entrée	Sortie	Entrée	Sortie
NO	BRO	BRO	NO	NO	BRO	BRO	NO	NO	BRO

Au tableau XIII et à la figure 46, on constate qu'il n'y pas de manipulation de données servant à gérer l'ordonnancement. En effet, la configuration montrée ci haut nous permet de profiter du fait que pour chaque opération, les données circulent dans le même ordre. Plus précisément, la corrélation et le filtrage sont effectués en ordre inverse. Puis lorsqu'on contraint le gradient, l'opération de contrainte s'effectue en ordre naturel. Finalement, la conversion fréquence/temps permet de retrouver l'ordre naturel pour le traitement de l'erreur de même que pour la sortie de l'égaliseur.

5.3.3.2 Largeur des bus internes et mise à l'échelle

Tel que nous l'avons mentionné plus haut, l'algorithme $R2^2PC$ utilise une stratégie pipelinée. Le nombre d'étage de papillon dépend de la taille de la FFT/IFFT selon le logarithme base deux du nombre de coefficients fréquentiels de l'égaliseur. On doit

spécifier la largeur des bus internes au niveau pré synthèse. En général, il y a $\lceil N/2 \rceil$ largeur de bus à configurer dont deux sont associées à l'entrée et à la sortie *(in_fft_w, out_fft_w)*. On recommande de poser la taille des étages internes supérieure à la plus grande valeur entre le bus d'entrée et de sortie pour augmenter les performances de rapport signal sur bruit de quantification (SQNR). Nous avons donc choisi de poser la taille des bus internes en suivant l'équation suivante:

$$IW_{FFT} = max\ (in_fft_w,\ out_fft_w) + 1 \tag{5.1}$$

La mise à l'échelle a été originalement intégrée au module $R2^2PC$ pour permettre d'éviter les débordements pendant le processus de calcul de la transformée de Fourier. En ce qui nous concerne, nous l'utilisons à cet égard mais également dans le but d'ajuster le signal conformément aux résultats du modèle de simulation. Nous avons aussi tenté d'utiliser la mise à l'échelle pour varier le pas d'adaptation. En effet, les transformées *r22pc3, r22pc4* et *r22pc5* (voir figure 46) sont directement liées au calcul du gradient. Cependant, nous avons abandonné l'idée puisque lorsqu'on supprimait des divisions par deux pour augmenter virtuellement le *pas d'adaptation*, il apparaissait sporadiquement des erreurs de débordement. Ce phénomène s'explique par deux principes. Dans un premier temps, lorsqu'on augmente le *pas d'adaptation* par le biais des signaux *scalingVect*, on se retrouve en fait à doubler la plage des valeurs possibles en sortie des modules de transformées de Fourier ainsi que du module de multiplication complexe (*s_cplx_mult2*). Ceci a un impact direct sur la quantification. Pour s'assurer qu'il n'y ait pas de dépassement, il faudrait ajouter $\log_2 2N$ bits d'entiers à chaque module affecté. Ce qui n'est pas souhaitable du point de vue ressources matérielles. La deuxième source d'erreur tirée de cette stratégie survenait lorsque l'on variait, à la hausse ou à la baisse, la mise à l'échelle entre deux blocs de traitement. Cette situation engendre des erreurs dans le calcul de la transformée. Sachant que chaque échantillon d'un bloc de traitement d'une transformée de Fourier affecte tous les points en sortie, on comprend bien cette problématique.

5.3.3.3 Latence

Les échantillons en entrée et en sortie du module $R2^2PC$ sont acheminés de manière sérielle. Chaque bloc peut se succéder un à la suite de l'autre sans interruption et sans logique de contrôle. Ainsi, lorsque le processus de traitement est enclenché, l'usager de ce module n'a pour seule responsabilité que de recueillir les résultats en sortie et de vérifier s'il y a dépassement. Cependant, lors de l'initialisation, il faut attendre le temps d'une transformée complète plus le nombre d'étages de papillons avant de récolter les données en sortie. En d'autres mots, la latence de chaque module de transformée de Fourier correspond à $2N + \log_2 2N$ cycles d'horloges. Le deuxième terme de cette équation correspond au nombre d'étages de papillons inclus dans le module. On constate que la latence est fonction de la longueur de l'égaliseur. En combinant les trois à cinq FFT/IFFT à implémenter dans l'algorithme FBMMA3M, on comprend pourquoi nous avons opté de retarder le gradient au niveau de l'équation d'adaptation des coefficients en simulation. Cette étape nous a permis de confirmer que l'égaliseur fréquentiel peut très bien fonctionner même avec cette modification de l'algorithme FBMMA3M (voir chapitre 4).

5.3.3.4 Initialisation du module r22pc

Le module $R2^2PC$ doit être initialisé par le signal *startin*. Ce signal agit en quelque sorte comme un *reset* puisqu'il réinitialise tous les compteurs de ce module. L'information en entrée n'est valide que lorsqu'un front montant suivi d'un front descendant n'ait été détecté au niveau du signal *startin*. Il faut également que le signal *ce* soit actif haut. Ce dernier (signal *ce*) permet à l'usager de faire fonctionner le module à une fréquence différente que celle du signal d'horloge. Les résultats en sortie ne sont valables que $2N$ (taille de la transformée de Fourier) cycles après que le signal *startout* ait été activé. La figure 49 illustre le chronogramme d'initialisation d'un module $R2^2PC$ configuré à huit points. Sur le schéma d'implémentation de la figure 46, nous

Rapport-Gratuit.com

avons ajouté une étoile sous le nom de chaque module de transformée de Fourier. Ceci permet d'indiquer que pour chacun de ces modules, nous avons intégré de la logique d'interface pour permettre une liaison homogène entre les modules suivants. Le module *gl_logic_r22pc* n'est pas montré sur le schéma d'implémentation pour alléger la présentation. Sa fonction est de gérer les signaux *startin* et *startout*. Il est souvent intégré à même les autres modules de l'algorithme.

Sur le chronogramme de la figure 49, on peut également observer que le signal *overflow* est activé tant que la sortie n'est pas valide. Ce n'est que deux cycles d'horloge plus tard qu'il se désactive.

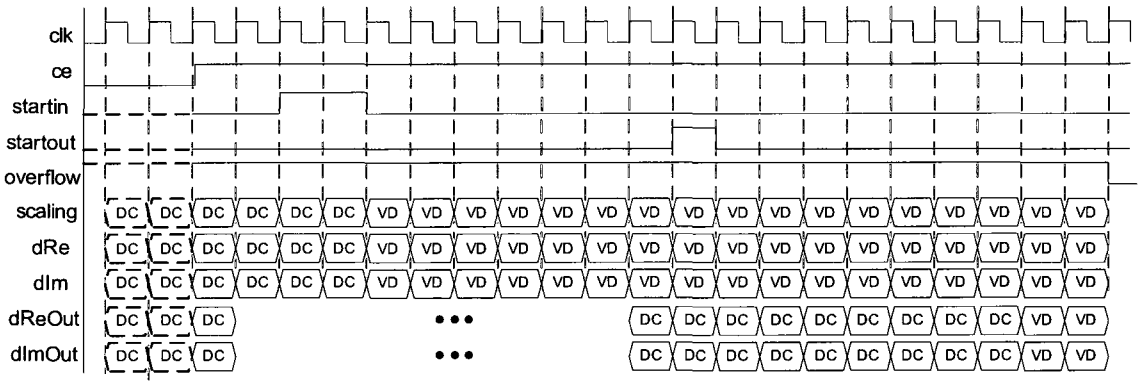


Figure 49 Chronogramme du module r22pc pour $N = 8$.

5.3.3.5 Complexité multiplicative du module r22pc

Au chapitre deux, nous avons abordé la complexité multiplicative théorique d'une transformée de Fourier. Ici, maintenant que nous avons choisi et présenté le module pour cette opération, nous pouvons établir plus précisément le nombre de multiplicateurs requis pour chaque bloc r22pcx. Les équations 5.2 et 5.3 montrent le nombre de multiplicateurs dédiés nécessaires à l'implémentation de ce module.

$$nMult_{radix2} = 2\log_2 2N - 2 \tag{5.2}$$

$$nMult_{radix4} = 2\log_2 2N - 4 \tag{5.3}$$

Donc, pour les configurations d'égaliseur où $2N$ peut s'exprimer par 2^x, on obtiendra la complexité multiplicative de l'équation 5.2 puis lorsque $2N$ peut être obtenu par 4^x, on aura la complexité multiplicative de l'équation 5.3.

5.3.4 Module de temporisation des données

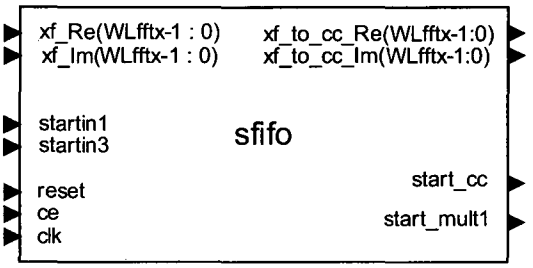


Figure 50 Module de temporisation des données.

Génériques

wl_fftx	integer	Indique le nombre de bits alloués pour un échantillon.
log2n	positive	Spécifie le logarithme en base deux associé au nombre de coefficients fréquentiel de l'égaliseur. Cette donnée permet, par le biais d'une fonction présynthèse, de déterminer le nombre de cases mémoires des tampons ainsi que la taille des compteurs (pointeurs).

Signaux en entrée

clk	Horloge.
ce	Activation du module.
reset	Réinitialisation des compteurs (pointeurs)
startin1	Signal d'activation de la mise en mémoire des données
startin3	Signal qui active un compteur dédié au contrôle des données en

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

	sorties de la mémoire.
xf_Re	Partie réelle des échantillons en entrée.
xf_Im	Partie imaginaire des échantillons en entrée.

Signaux en sortie

xf_to_cc_re	Partie réelle des échantillons en sortie.
xf_to_cc_im	Partie imaginaire des échantillons en sortie.
start_mult1	Drapeau indiquant l'arrivée du premier échantillon fréquentiel dans la chaîne de filtrage. Ce drapeau est utilisé par le module wfifo.

Le module *sfifo* permet de conserver les données sortantes de la transformée de Fourier *r22pc1* jusqu'à ce que la sortie du bloc d'erreur fréquentielle (*r22pc3*) soit valide. En effet, la multiplication qui suit (*s_cplx_mult2*) correspond à une inter corrélation entre le vecteur d'erreur et le signal d'entrée. C'est pourquoi chaque échantillon doit correspondre au même indice.

5.3.4.1 Synchronisation à l'initialisation

Lorsque le signal *startout1* est activé, un compteur est enclenché jusqu'à ce que la valeur $2N$ soit atteinte. À cet instant, on commence l'emmagasinage des données dans une mémoire tampon. Lorsque le signal *startout3* s'active, cela indique que l'erreur fréquentielle sera valide dans $2N$ cycles. Le même compteur permet de gérer l'instant où l'information contenue dans la mémoire tampon peut être acheminée vers le module de *s_cplx_mult2*. Lorsque ce moment est venu, le module *sfifo* est en régime permanent. Un pointeur est utilisé pour l'entrée des données dans la mémoire tampon puis un autre pointeur est employé pour la sortie des données. Ainsi, puisque deux pointeurs sont requis pour gérer l'entrée sortie de l'information, le circuit généré lors de la synthèse sera une mémoire à double port. Chaque dimension (partie réelle et partie imaginaire) requiert son bloc mémoire. Un jeu de deux pointeurs circulaires contrôle l'entrée et la sortie des échantillons.

5.3.4.2 Configuration du module *sfifo*

La taille de la mémoire tampon dépend de la taille de l'égaliseur. À partir de l'activation du signal *startin1*, $3(2N + \log_2 2N$ + latence des autres blocs) cycles doivent être accomplis avant que l'information soit autorisée en sortie. Ainsi, la taille requise est $7N$. En réalité, $(6N + 3\log_2 2N$ + latence autres blocs) sont requis mais il faut choisir une dimension multiple de deux pour l'implémentation. Que ce soit en mode contraint ou non, la profondeur de la mémoire reste la même puisque la contrainte du gradient survient après l'inter corrélation.

5.3.5 Module de multiplication complexe

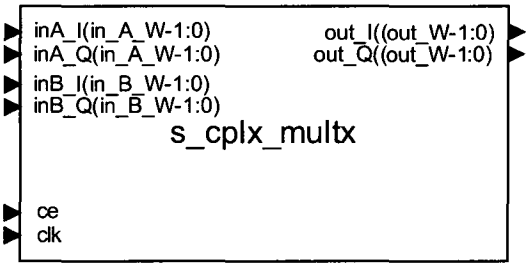


Figure 51 Module de multiplication complexe.

Génériques

in_A_W	positive	Indique la largeur du mot binaire de l'entrée A
in_B_W	positive	Indique la largeur du mot binaire de l'entrée B
inter_W	positive	Indique la largeur du mot binaire en sortie de chacune des quatre multiplications réelles.
out_B_W	positive	Indique la largeur du mot binaire du résultat de la multiplication complexe en sortie
frac_A_W	positive	Indique où le point de la fraction se situe sur le signal d'entrée A.
frac_B_W	positive	Indique où le point de la fraction se situe sur le signal d'entrée B.

frac_inter_W	positive	Indique où le point de la fraction se situe sur le signal de chacune des quatre multiplications réelles.
frac_out_W	positive	Indique où le point de la fraction se situe sur le signal de sortie de la multiplication complexe.

Signaux en entrée

clk	Horloge.
ce	Activation du module.
inA_I	Partie réelle (In phase) du signal de l'entrée A.
inA_Q	Partie imaginaire (Quadrature phase) du signal de l'entrée A.
inB_I	Partie réelle (In phase) du signal de l'entrée B.
inB_Q	Partie imaginaire (Quadrature phase) du signal de l'entrée B.

Signaux en sortie

out_I	Partie réelle des échantillons en sortie.
out_Q	Partie imaginaire des échantillons en sortie.

Puisque le traitement adaptatif de l'égaliseur fréquentiel est sériel, il n'y a que deux multiplicateurs complexes à implémenter matériellement pour réaliser l'adaptation de l'égaliseur FBMMA3M. Un pour accomplir le traitement de l'inter corrélation entre les symboles d'entrée et l'erreur puis un autre pour effectuer le filtrage fréquentiel. On distingue ici une réduction nette de la complexité multiplicative par rapport à la théorie. En effet, théoriquement, il faudrait implémenter $4N$ multiplicateurs complexes si le traitement était accompli par bloc. Bien sûr, en utilisant des stratégies de réutilisation de ressources, ce nombre pourrait être réduit mais, tel que mentionné au chapitre 4, il ne faut pas perdre de vue que la fréquence de traitement serait affectée si le traitement était accompli par bloc de données.

Rappelons qu'une multiplication complexe requiert quatre multiplications réelles ainsi que deux additions réelles. Dans le but de maximiser la fréquence de traitement, nous avons implémenté le multiplicateur complexe avec une latence de trois cycles d'horloge.

Un cycle est attribué à l'ajustement des données. C'est le module *C* (voir figure 52) qui accomplie cette tâche. Le signal d'entrée (A ou B) possédant le plus court mot binaire est ajusté à la même longueur que le signal d'entrée le plus long. Au deuxième cycle d'horloge, les quatre multiplicateurs réels accomplissent leur fonction. À ce niveau, les vecteurs sont de longueur *inter_W*. Ce paramètre est configurable et il est recommandé de le fixer à $2in_A_W$ pour plus de précision et moins de risque de dépassement. Par la suite, le troisième cycle permet de finaliser la multiplication complexe avec l'étage d'addition/soustraction. Au court du même cycle d'horloge, le module *CR* effectue un arrondi et ajuste les données en sortie à *out_W*. Les valeurs de configurations des variables *in_A_W*, *in_B_W* et *out_W* sont fixées avec les résultats du tableau X présenté au chapitre 4. Ainsi, les signaux d'entrée prennent la quantification du bloc précédent et les signaux en sortie sont fixés à la valeur qui leur est associée au tableau X.

En ce qui a trait au module *s_cplx_mult2*, nous avons inversé la position des additionneurs montrés à la figure 52. En effet, l'opération est $\boldsymbol{r}_F^*(k)\otimes\boldsymbol{e}(k)$. La multiplication du conjugué complexe fait en sorte que la soustraction doit être accomplie sur la partie imaginaire. Cette modification nous a permis d'effectuer deux opérations en une. Soit le conjugué complexe de $r_F^*(k)$ ainsi que la multiplication complexe.

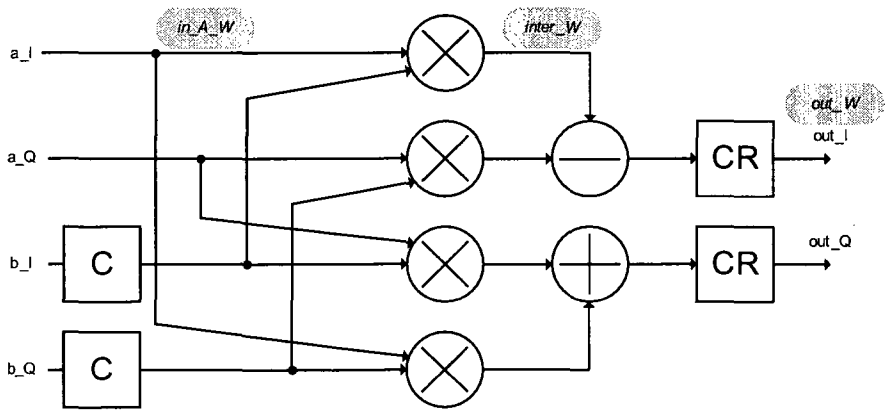


Figure 52 Schéma d'implémentation du module de multiplication complexe.

5.4 Module de sélection des données

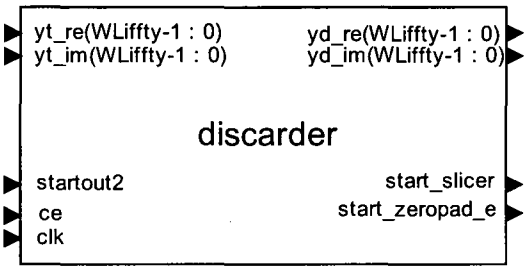


Figure 53 Module de sélection des données.

Générique

log2n	positive	Logarithme base 2 du nombre de coefficients fréquentiels (2*N*). La largeur binaire du compteur servant au contrôle de la sortie est *log2n*-1

Signaux en entrée

clk	Horloge.
ce	Activation du module.
startout2	Signal provenant du module *r22pc2*. Il déclenche l'initialisation du module. Ce signal agit comme un *reset* puisque lorsqu'il est enclenché, les compteurs ainsi que les signaux de contrôle interne sont remis à zéros.
yt_re	Partie réelle (In phase) du signal de l'entrée provenant de la sortie du module *r22pc2*.
yt_im	Partie imaginaire (Quadrature phase) du signal de l'entrée provenant de la sortie du module *r22pc2*.

Signaux en sortie

yd_re	Partie réelle (In phase) du signal en sortie
yd_im	Partie imaginaire (Quadrature phase) du signal en sortie
start_slicer	Drapeau qui indique que la sortie est valide. Ce signal est transmis à la sortie de l'égaliseur de sorte que l'usager puisse gérer l'échantillonnage des données en sortie du module

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

	FBMMA3M.
start_zeropader	Signal de déclenchement du module *zeropader*.

Le module *discarder* fait la sélection des données en sortie du module de IFFT (*r22pc2*). En effet, puisque nous employons une méthode de filtrage fréquentiel basée sur la technique de chevauchement et sauvegarde à 50%, la première moitié des résultats de chaque bloc en sortie de la IFFT sont corrompus. Ainsi, seul les N derniers éléments sont valides.

La logique est implémentée sous forme de MEF à trois états. Soit un état *init* qui permet de détecter l'instant où l'information est valide en sortie du module *r22pc2* lors de l'initialisation. Au moment où l'initialisation est terminée, le signal *start_zeropader* est activé pour déclencher le module *zero_pader*. Les états *send* et *discard* sont employés en régime permanent de manière cyclique, se partageant chacun N cycles d'horloge par bloc de $2N$ échantillons. En mode *send*, les données sont acheminées à la fois en sortie de l'égaliseur puis vers les modules de gestion de l'erreur. Également dans ce mode, le signal *start_slicer* est monté au niveau haut pour indiquer que les données en sortie sont valides. Ce signal peut être utilisé par l'usager de l'égaliseur à des fins de gestions de données en sortie. En mode *discard*, le signal *start_slicer* est désactivé et les informations en sortie sont figées à la dernière donnée valide.

5.4.1 Modules de gestion de l'erreur

Ce qui distingue l'algorithme FBMMA3M de l'algorithme FBLMS se situe au niveau de la gestion de l'erreur. En effet, l'égaliseur FBMMA3M peut générer trois types de signaux d'erreurs. Une erreur basée sur une séquence d'entraînement, une autre obtenue à partir d'une décision sur le symbole égalisé puis une erreur autodidacte basée sur l'algorithme MMA. Puisque l'égaliseur FBMMA3M doit être programmable pour accommoder les modulations 4, 16 et 64 QAM, nous avons choisi d'utiliser des tables de références (LUT) pour implémenter le module de décision ainsi que le module de l'erreur MMA.

5.4.1.1 Module de décision

Le module de décision est implémenté à l'aide de deux LUTs à contenu identique. Le résultat en sortie dépend de la valeur du signal de sortie filtré (*yd_re, yd_im,* figure 46) de l'égaliseur. Les symboles QAM peuvent être représentés à l'aide d'une combinaison de deux signaux PAM. Ainsi, la décision sur chaque dimension est obtenue par une table de référence dont les données sont distribuées équitablement pour chaque niveau possible.

Nous proposons une configuration de 1024 possibilités. Ceci signifie que le signal d'entrée doit être sur dix bits. Pour ce faire, nous avons inséré un module qui accomplit une troncature du signal filtré en sortie de l'égaliseur. Cette disposition permet une granularité sur les décisions de $1024/\sqrt{(M)}$ points. Au niveau de la représentation des symboles, nous suggérons une quantification sur six bits, soit cinq bits de fractions et un bit de signe. Ainsi, avec cette proposition de configuration, les deux dimensions requièrent un bloc de mémoire RAM de 1024 registres de 6 bits.

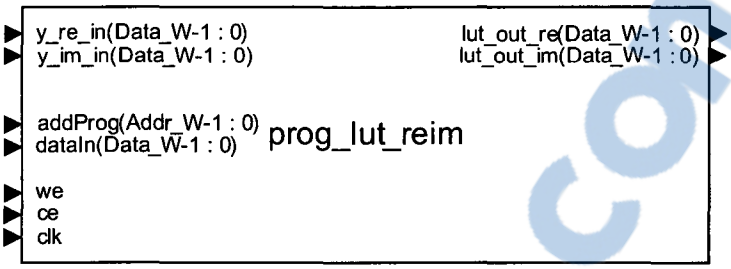


Figure 54 Modules de décision et de l'erreur MMA

Génériques

Data_W	positive	Cette valeur est affectée directement à la largeur des registres du bloc de mémoire RAM. Pour le module de décision, elle indique la quantification des symboles décidés. Puis, pour le module de l'erreur MMA, elle indique le nombre de bits requis pour représenter l'amplitude de l'erreur MMA de chaque dimension.
Addr_W	positive	Indique la largeur du bus d'adresse du bloc mémoire RAM. Par le fait même, le nombre de registres requis pour le bloc de mémoire RAM. Indique également combien de bits doivent être tronqués sur chaque dimension des symboles filtrés à la sortie du module *discarder*.

Signaux en entrée

clk	Horloge.
ce	Activation du module.
we	Signal permettant de configurer le bloc de mémoire RAM en lecture (niveau bas) ou en écriture (niveau haut).
y_re_in	Partie réelle (In phase) du signal de l'entrée provenant de la sortie du module *discarder*.
y_im_in	Partie imaginaire (Quadrature phase) du signal de l'entrée provenant de la sortie du module *discarder*.

Signaux en sortie

lut_out_re	Partie réelle (in phase) du signal en sortie de la table de

Rapport-Gratuit.com

	référence. Le nom indiqué diffère de ce qui est affiché sur la figure 46 car nous généralisons la présentation pour les deux modules (décision et erreur MMA). Au niveau du module de décision, ce signal est dirigé vers la sortie de l'égaliseur FBMMA3M et également vers le module de soustraction pour générer l'erreur LMS. En ce qui concerne le module de l'erreur MMA, la sortie est acheminée vers le module de sélection de l'erreur.
lut_out_im	Partie imaginaire (quadrature phase) en sortie. Le nom indiqué diffère de ce qui est affiché sur la figure 46 car nous généralisons la présentation pour les deux modules (décision et erreur MMA). Au niveau du module de décision, ce signal est dirigé vers la sortie de l'égaliseur FBMMA3M et également vers le module de soustraction pour générer l'erreur LMS. En ce qui concerne le module de l'erreur MMA, la sortie est acheminée vers le module de sélection de l'erreur.

Sachant que l'amplitude des signaux véhiculant dans l'égaliseur FBMMA3M est représentée en complément à deux, il faut porter une attention particulière à la manière dont on insère les données dans la table de référence. En effet, les valeurs négatives doivent être rangées dans l'ordre logique inverse puisqu'avec une représentation en complément à deux, la graduation est du plus grand nombre jusqu'au plus petit lorsqu'on l'observe par le biais d'une quantification entière non signée. La figure 54 illustre l'ordonnancement des valeurs de registres sous forme graphique pour une constellation 64 QAM. À noter que la graduation de l'amplitude est en entier. Il faut donc diviser par 2^{-5} pour retrouver les valeurs réelles des amplitudes.

La sortie de ce module est à la fois acheminée à la sortie de l'égaliseur et au module *cplx_sous*. Tel que nous l'avons mentionné au chapitre 4, la sortie de ce module est quantifiée avec un bit d'entier et huit bits de fractions. Soit dix bits en considérant le bit de signe.

Notons que le module de décision ainsi que le module de l'erreur MMA peuvent être configuré autrement. Par exemple, si l'usager envisage n'utiliser que du 4-QAM, la

granularité requise sera beaucoup moins exigeante. Ainsi, la taille des bloc mémoire pourra être réduite.

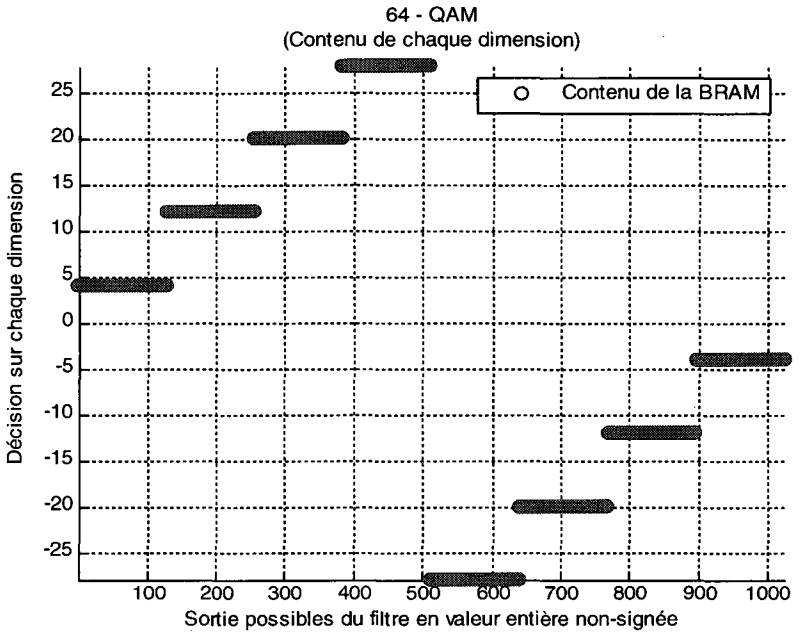


Figure 55 Configuration de la table de de décision pour une modulation 64-QAM

5.4.1.2 Module de l'erreur MMA

La construction du module de l'erreur MMA est identique à celle du module de décision. En effet, puisque l'erreur MMA ne dépend que de la variable y en sortie du filtre (sortie du module *r22pc2*) , on peut précalculer les résultats et les placer dans une mémoire RAM. Nous proposons à nouveau une table de référence à 1024 registres. Cependant, la taille des registres est posée à dix bits. Soit un bit d'entier et huit bits de fractions. Au chapitre quatre, nous avons vu que l'erreur MMA peut tendre vers l'infini si on ne la restreint pas. Ainsi, en posant cette quantification, nous restreignons intrinsèquement l'amplitude de sortie à $\pm 1{,}99609375$. Donc, les valeurs en sortie seront plafonnées par cette plage. La figure 53 illustre la plage de donnée possible en sortie. À noter que la graduation de l'amplitude est en entier. Il faut donc diviser par 2^{-8} pour retrouver les valeurs réelles des amplitudes. Au chapitre 4, nous avons également observé qu'avec

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

une quantification telle que celle proposée ici, les résultats de convergence ne sont pas très loin de ce qu'on obtient en théorie. Au niveau de la programmation de la mémoire RAM, les données négatives doivent être placés dans l'ordre inverse pour correspondre avec la notation en complément à deux.

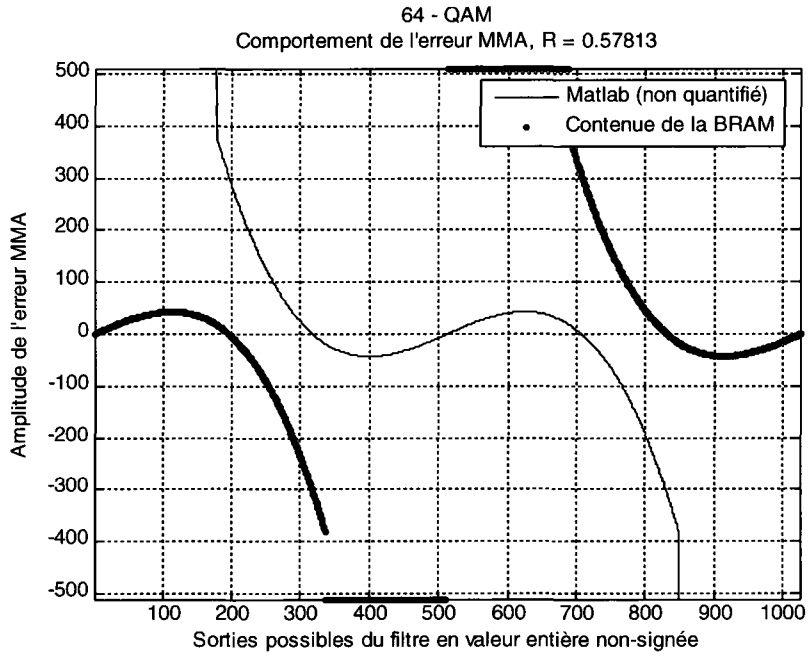


Figure 56 Configuration de la table de référence de l'erreur MMA pour une modulation 64-QAM

5.4.1.3 Module de sélection de l'erreur (*mux_mma_lms*)

Au quatrième chapitre, nous avons présenté les seuils optimaux pour accomplir la sélection entre l'erreur MMA et LMS. Pour les modulations 4-16-64 QAM, les valeurs sont respectivement 0,25, 0,125 et 0,0625. On observe que ces valeurs sont des multiples de deux. Or, pour comparer, on regarde si un des bits de poids supérieurs au seuil est actif. Le cas échéant, cela signifie que l'erreur LMS est supérieure au seuil. Ainsi, si la partie réelle et imaginaire est au dessus seuil, on conserve l'erreur LMS en sortie. Autrement, c'est l'erreur MMA que l'on achemine vers le module *zero_pader*. Le même raisonnement peut être appliqué si le signal d'erreur est négatif. Cependant, on regarde cette fois si les bits de poids supérieurs au seuil sont au niveau bas.

5.4.2 Module d'insertion de zéros

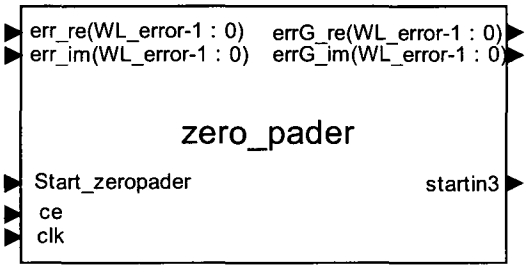


Figure 57 Module d'insertion de zéros.

Signaux en entrée

clk	Horloge.
ce	Activation du module.
start_zeropader	Réinitialisation de des compteurs.
err_re	Partie réelle de l'échantillon en entrée.
err_im	Partie imaginaire de l'échantillon en entrée.

Signaux en sortie

startin3	Drapeau qui déclenche le module *r22pc3*
errG_re	Partie réelle des échantillons en sortie
errG_im	Partie imaginaire des échantillons en sortie

Le module d'insertion de zéros, *zero_pader*, permet de transformer le vecteur d'erreur de manière à considérer le filtrage fréquentiel. En effet, puisqu'on utilise la méthode de chevauchement et sauvegarde à 50%, les *N* premiers échantillons d'erreurs doivent être nuls. De plus, sachant que l'égaliseur FBMMA3M fait de l'échantillonnage fractionné à *Tsym*/2, on doit insérer des zéros à un échantillon d'erreur sur deux. En effet, les résultats d'erreur prises entre deux symboles ne sont pas valides. Il est donc préférable de les annuler en les remplaçant par des zéros.

5.4.3 Module de contrainte du gradient

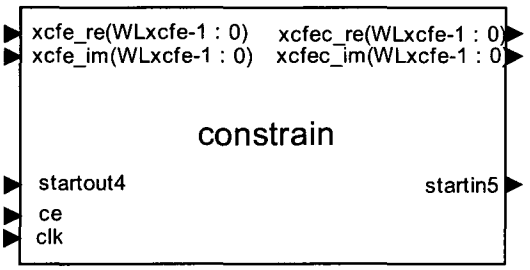


Figure 58 Module module de contrainte du gradient

Signaux en entrée

clk	Horloge.
ce	Activation du module.
startout4	Réinitialisation des compteurs du module.
xcfe_re	Partie réelle de l'échantillon en entrée.
xcfe_im	Partie imaginaire de l'échantillon en entrée.

Signaux en sortie

startin5	Drapeau qui déclenche le module *r22pc5*
xcfec_re	Partie réelle des échantillons en sortie
xcfec_im	Partie imaginaire des échantillons en sortie

Le module de contrainte du gradient n'est implémenté que lorsque l'on configure cette option. Le résultat de l'inter corrélation fréquentiel entre le signal d'erreur et celui de l'entrée est converti dans le domaine temporel par le module *r22pc4* et ensuite, on retranche les N derniers échantillons du bloc de taille $2N$ pour les remplacer par des zéros. Ceci dans le but d'éliminer la circularité engendrée par la transformée de Fourier. Si on ne choisit pas l'option de contrainte du gradient, c'est l'algorithme non contraint qui est mis en circuit. Ce dernier ne requiert pas de transition vers le domaine du temps et permet ainsi d'économiser deux modules de transformée de Fourier. Sur la figure 46, les modules *r22pc4*, *r22pc5* et *constrain* sont remplacés par les signaux en pointillés.

5.4.4 Module d'adaptation des coefficients

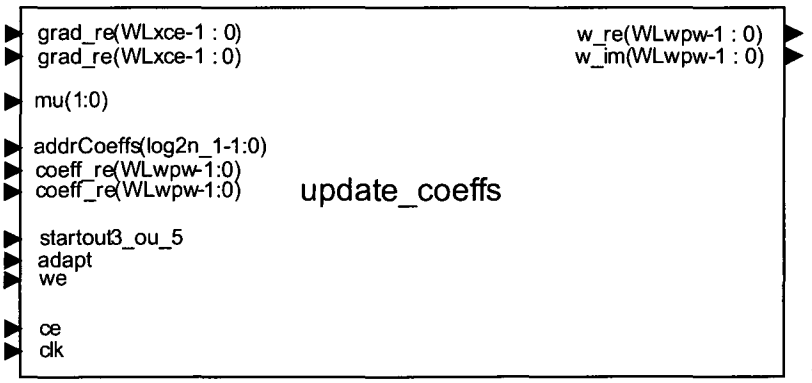


Figure 59 Module d'adaptation des coefficients

Génériques

wlxce	integer	Indique le nombre de bits alloués pour un échantillon en entrée. (quantification en sortie du module *r22pc5* ou s_*cplx_mult2* selon que l'égaliseur soit configuré en mode contraint ou non contraint).

log2n	positive	Spécifie le logarithme en base deux associé au nombre de coefficients fréquentiels de l'égaliseur.

Signaux en entrée

clk	Horloge.
ce	Activation du module.
we	Active le bloc mémoire en mode lecture (niveau bas) ou écriture (niveau haut).
adapt	Active (niveau haut) ou désactive (niveau bas) l'adaptation des coefficients.
startout3_ou5	Réinitialise tous les compteurs. (agit comme un *reset*). Dans le cas où l'on implémente la contrainte du gradient, c'est le signal *startout5* provenant du module *r22pc5* qui est utilisé. Dans l'autre cas, on prend celui du module *r22pc3*, soit *startout3*.
mu	Signal programmable qui aiguille la sélection du *pas d'adaptation*. Quatre choix sont possibles et configurables pré synthèse.
addrCoeffs	Vecteur d'adresse pour la programmation des coefficients dans le bloc de mémoire RAM.
coeffs_re	Partie réelle du registre utilisé pour programmer ou initialiser les coefficients dans le bloc de mémoire RAM.
coeffs_im	Partie imaginaire du registre utilisé pour programmer ou initialiser les coefficients dans le bloc de mémoire RAM.
grad_re	Partie réelle du gradient en entrée.
grad_im	Partie imaginaire du gradient en entrée.

Signaux en sortie

w_re	Partie réelle des coefficients en sortie.
w_im	Partie imaginaire des coefficients en sortie.

Le module d'adaptation des coefficients accomplit l'opération suivante:

$$w_F(k+1)=w_F(k)+\mu_{LMS}\left[X_C E(k-D)\right]. \tag{5.4}$$

Le terme $X_c E$ représente le gradient. D désigne le délai engendré par la latence de l'algorithme FBMMA3M. Sa valeur se chiffre autour de trois pour l'algorithme non contraint et autour de cinq lorsqu'on choisit l'option de contrainte du gradient. Deux blocs de mémoire RAM à double port de $2N$ registres sont requis pour emmagasiner les coefficients. Après chaque traitement de $2N$ échantillons, les coefficients de l'ancien bloc sont additionnés au gradient que l'on pondère à l'aide du *pas d'adaptation*. Le résultat est acheminé en sortie puis est également inscrit à la place de l'ancienne valeur dans les registres Ainsi que tous les autres modules, le traitement se fait entièrement de manière sérielle. De cette manière, l'opération 5.4 peut être accomplie en un seul cycle d'horloge, un échantillon à la fois.

La pondération avec le *pas d'adaptation* est accompli par le biais d'un décalage du point virtuel de la fraction vers la gauche. Ainsi, seul les nombres divisibles par deux peuvent être choisi comme valeur de *pas d'adaptation*. Cette restriction nous permet d'éliminer l'usage de deux multiplicateurs réels. Il est possible de programmer la valeur du décalage. Quatre choix sont disponibles et configurables dans le fichier de configuration pré synthèse de l'égaliseur FBMMA3M.

En tout temps, il est possible de reprogrammer la valeur des coefficients dans la mémoire RAM. Cette option peut être utilisé lors de l'initialisation de l'égaliseur afin de poser le coefficient central à l'unité (voir équation 3,5). À noter que puisqu'on est en fréquence, les coefficients doivent être initialisés comme suit: On peut également accomplir cette fonction pendant le traitement, advenant une situation où les coefficients divergeraient à cause d'un changement trop brusque des coefficients du canal de transmission.

5.4.5 Module de temporisation des coefficients

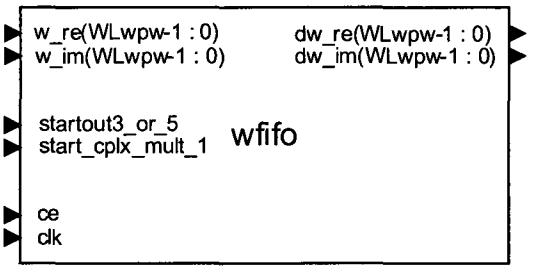


Figure 60 Module de temporisation des coefficients

Signaux en entrée

clk	Horloge.
ce	Activation du module.
startout3ou5	Réinitialise tous les compteurs. (agit comme un *reset*). Dans le cas où l'on implémente la contrainte du gradient, c'est le signal *startout5* provenant du module *r22pc5* qui est utilisé. Dans l'autre cas, on prend celui du module *r22pc3*.
start_cplx_mult1	Signal qui déclenche un compteur dédié à la synchronisation entre l'arrivée des coefficients et celle du signal $r_F(k)$
w_re	Partie réelle des coefficients en entrée.
w_im	Partie imaginaire des coefficients en entrée.

Signaux en sortie

wd_re	Partie réelle des coefficients en sortie.
wd_im	Partie imaginaire des coefficients en sortie.

Le module de temporisation des données permet d'aligner les signaux $r_F(k)$ et $w_F(k)$. En effet, la latence introduite par les différents modules de l'égaliseur n'est pas un multiple de N et varie selon le nombre de coefficients désirés. Ainsi, le module *wfifo* est constitué de deux (partie réelle et imaginaire) mémoires RAM à double port de $2N$ registres. À l'initialisation, les coefficients y sont emmagasinés jusqu'à l'arrivée du

premier échantillon $r_F(k)$ d'un bloc de $2N$. Par la suite, un jeu de deux pointeurs circulaires contrôlent l'entrée et la sortie des échantillons.

5.5 Configuration et programmation de l'égaliseur FBMMA3M

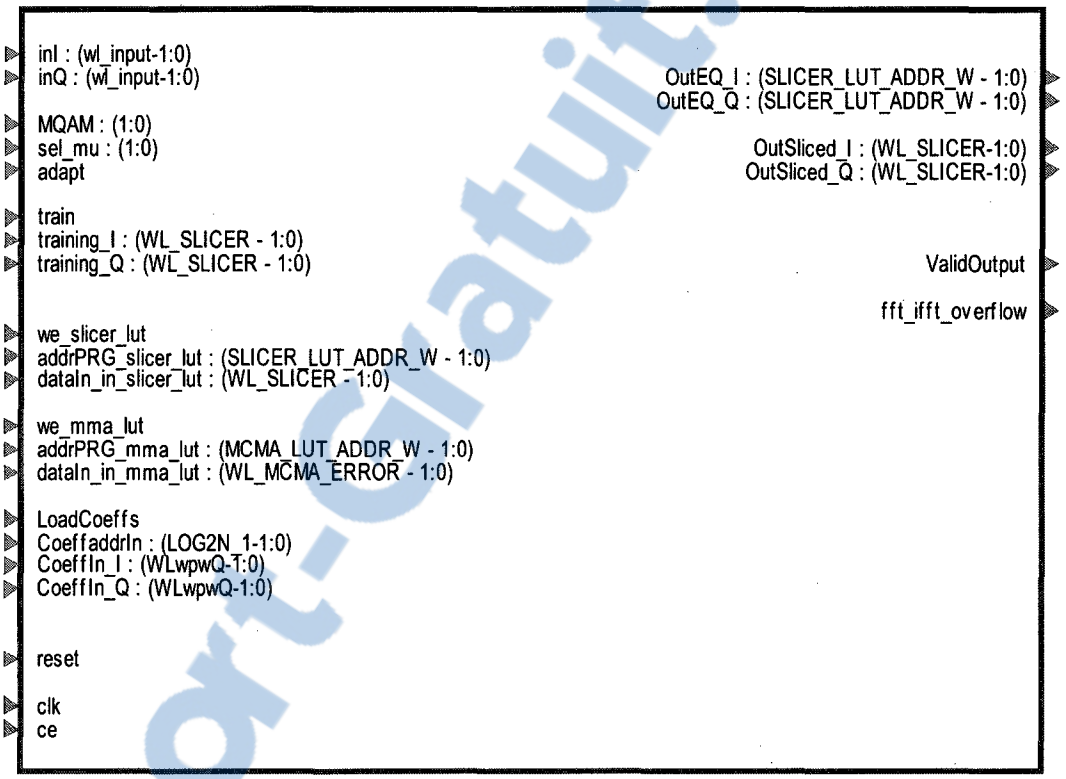


Figure 61 Module FBMMA3M

Signaux en entrée

clk	Horloge.
ce	Activation du module. Ce signal peut être utilisé comme horloge à $4Tsym/s$.
Load_Coeffs	Signal de contrôle de lecture (niveau bas) et écriture (niveau haut) des deux blocs de mémoire RAM contenant les coefficients de l'égaliseur.
Coeff_addrin	Vecteur d'adresse des deux blocs mémoire RAM contenant les coefficients de l'égaliseur.

Rapport-Gratuit.com

Coeffin_I	Partie réelle des coefficients de l'égaliseur. On utilise ce signal lorsqu'on souhaite programmer la partie réelle des coefficients dans l'égaliseur.
Coeffin_Q	Partie imaginaire des coefficients de l'égaliseur. On utilise ce signal lorsqu'on souhaite programmer la partie imaginaire des coefficients dans l'égaliseur.
we_mma_lut	Signal de contrôle de lecture (niveau bas) et écriture (niveau haut) des deux blocs de mémoire RAM contenant les valeurs précalculées de l'erreur MMA.
addrPGR_mma_lut	Vecteur d'adresse des deux blocs mémoire RAM contenant les données de l'erreur MMA
datain_in_mma_lut	Vecteur de données de la LUT MMA. On utilise ce signal lorsqu'on souhaite programmer la table de référence MMA.
we_slicer_lut	Signal de contrôle de lecture (niveau bas) et écriture (niveau haut) des deux blocs de mémoire RAM contenant les zones de décision du module de décision.
addrPGR_slicer_lut	Vecteur d'adresse des deux blocs mémoire RAM contenant les données du module de décision.
datain_in_slicer_lut	Vecteur de données de la LUT contenant les zones de décision. On utilise ce signal lorsqu'on souhaite programmer la table de référence du module de décision.
Train	Signal d'activation du mode entraînement. Au niveau haut, l'erreur est calculée en fonction de la séquence d'entraînement. Puis au niveau bas, la sortie du module de décision est choisie comme référence pour le calcul de l'erreur LMS.
training_I	Partie réelle de la séquence d'entraînement.
training_Q	Partie imaginaire de la séquence d'entraînement.
MQAM	Vecteur de deux bits qui détermine quel seuil utiliser en mode LMS-MMA. Les seuils sont programmés pour accommoder les modulations 4-16-64-QAM. 00 signifie le mode LMS seulement. 01 signifie le seuil pour une modulation 64-QAM. 10 signifie le seuil pour une modulation 16-QAM et 11 signifie le seuil pour une modulation 4-QAM.
sel_mu	Vecteur de deux bits qui détermine le poids du *pas d'adaptation*. Quatre valeurs sont configurables et leur choix est programmable.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

Adapt	Signal qui permet d'activer (niveau haut) ou de désactiver (niveau bas) le rafraîchissement des coefficients de l'égaliseur.
In_I	Partie réelle des symboles en entrée de l'égaliseur.
In_Q	Partie imaginaire des symboles en entrée de l'égaliseur.

Signaux en sortie

out_EQ_I	Partie réelle des symboles filtrés par l'égaliseur.
out_EQ_Q	Partie imaginaire des symboles filtrés par l'égaliseur.
outSliced_I	Partie réelle des symboles décidés par l'égaliseur.
outSliced_Q	Partie imaginaire des symboles décidés par l'égaliseur.
ValidOutput	Signal qui indique que la séquence d'information en sortie est valide. L'usagé peut se servir de ce signal pour se synchroniser avec la séquence d'entraînement.
fft_ifft_overflow	Signal qui indique qu'un débordement est survenu dans un des module de transformée de Fourier.

L'égaliseur FBMMA3M que nous avons implémenté est configurable et programmable. Il peut être synthétisé avec le gradient contraint ou non contraint. Le tableau XIV rapporte les paramètres programmables ainsi que les variables configurables accessibles dans le fichier *define_params.vhd.* Les indices *C* et *P* désigne respectivement *configurable* et *programmable.* Les variables configurables sont associées aux différents génériques des sous modules de l'égaliseur que nous avons présentés plus haut dans ce chapitre.

Tableau XIV

Paramètres programmables et configurables de l'égaliseur FBMMA3M

Paramètres	C	P	Description
MQAM		√	Indique quel seuil utiliser au module de sélection de l'erreur en mode LMS-MMA. Si le module de décision est programmé pour une modulation 4-QAM, le signal doit transmettre «11». Pour du 16-QAM c'est «10» et pour du

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

			64-QAM c'est «01». Si l'usager désire un traitement uniquement en mode LMS, le signal doit transmettre «00».
mu	√	√	Le plus haut *pas d'adaptation* désiré doit être configuré préalablement dans le fichier *define_params.vhd.* Puis, lorsque l'égaliseur est en fonction, il est possible de varier le *pas d'adaptation.* Un «00» programme le *pas d'adaptation* originale fixé dans le fichier de configuration. Un «01» programme le pas d'adaptation originale sur deux. Et ainsi de suite jusqu'à «11».
log2n	√		Indique le logarithme base deux du nombre de coefficients fréquentiels désiré. N coefficients temporels requièrent $2N$ coefficients fréquentiels.
WL_input	√		Indique la largeur binaire des symboles corrompus à l'entrée de l'égaliseur.
FL	√		Indique le nombre de bits de fractions requis pour tous les points de quantifications à l'exception des coefficients, du module de décision et du module de l'erreur MMA qui sont eux même configurables. Pour cette valeur, nous recommandons $FL \geq 10$. (Voir chapitre 4).
FL_W	√		Indique le nombre de bit de fractions requis pour les coefficients. Pour cette valeur, nous recommandons $FL \geq 10$. (Voir chapitre 4).
TF_W	√		Indique le nombre de bits requis pour les facteurs de phases de chaque transformée de Fourier. Cette valeur peut être posée entre ces bornes: $FL \leq TF_W \leq 18$. Une quantification plus élevée augmentera dramatiquement la complexité de l'égaliseur en terme de multiplicateurs requis.
WL_slicer	√		Indique le nombre de bits requis pour représenter les symboles en sortie de l'égaliseur. Nous recommandons six bits dont un bit de signe et cinq bits de fractions.
FL_slicer	√		Indique le nombre de bits de fractions requis pour représenter les symboles en sortie de l'égaliseur.
Slicer_LUT_ADDR_W	√		Indique le nombre de registres requis pour la table de référence du module de décision. Nous recommandons dix bits. Plus cette valeur est élevée, plus la granularité sera fine.
FL_Slicer_ADDR_W	√		Indique la partie fractionnaire requise pour le module de décision. Cette variable configurable permet également

			d'effectuer la juste troncature à la sortie du module *r22pc2*.
WL_MMA_ERROR	√		Indique le nombre de bits requis pour représenter l'erreur MMA en complément à deux. Nous recommandons dix bits dont huit bits de fractions. (Voir chapitre 4).
FL_MMA_ERROR	√		Indique le nombre de bits de fractions requis pour représenter l'erreur MMA. Nous recommandons huit bits.
MMA_ADDR_W	√		Indique le nombre de registres requis pour la table de référence du module de l'erreur MMA. Nous recommandons dix bits.
FL_MMA_ADDR	√		Indique la partie fractionnaire requise pour le module de l'erreur MMA. Cette variable configurable permet également d'effectuer la juste troncature à la sortie du module *r22pc2*.

Les tables de référence du module de décision et de l'erreur MMA ainsi que la table des coefficients peuvent être programmées en tout temps. Par exemple, si pendant le traitement on détecte un débordement irrécupérable par le biais du signal *fft_ifft_overflow*, la table des coefficients pourrait être reprogrammée. Cette propriété peut être intéressante pour l'usager qui souhaite utiliser la modulation adaptative. En effet, nous avons vu que la constellation 4-QAM converge plus rapidement que celle de 64-QAM en mode autodidacte. On pourrait donc tirer profit de cette réalité puis utiliser la modulation 4-QAM pour faire converger les coefficients et lorsque l'adaptation est accomplie, on pourrait reprogrammer les tables de décision ainsi que de l'erreur MMA pour accommoder du 64-QAM.

La figure 51 illustre un exemple d'initialisation de l'égaliseur FBMMA3M. Au départ, on remplit les tables de référence et on programme le coefficient central dans le domaine des fréquences. Puis, lorsque cette étape est accomplie, on réinitialise tous les compteurs en activant le signal *reset*. Après cette opération, on active le processus d'égalisation. Notez que dans cet exemple, l'égaliseur fonctionne avec l'horloge principale puisque nous posons le signal d'activation *ce* constamment actif.

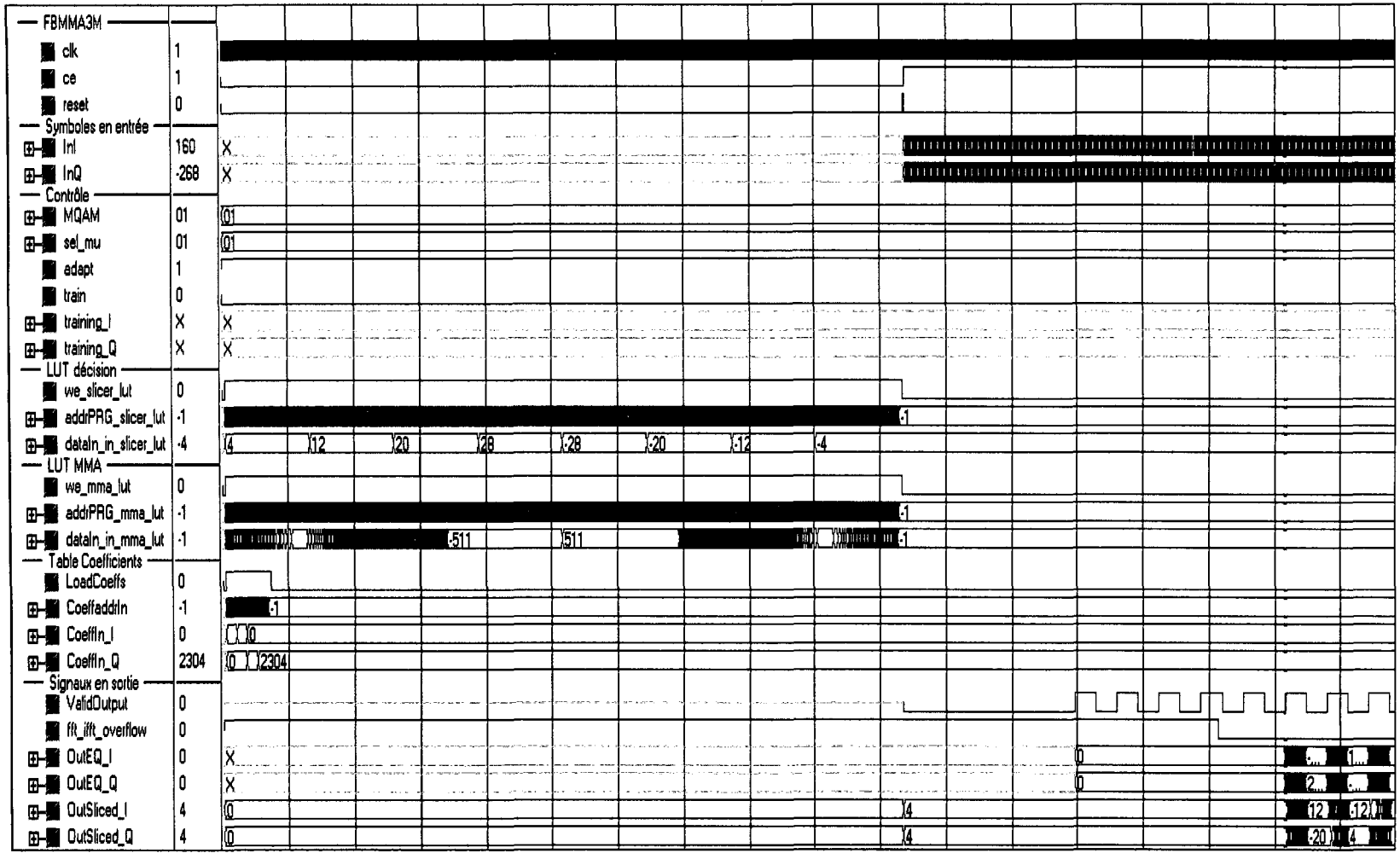


Figure 62 Exemple d'initialisation du module FBMMA3M.

5.5.1 Complexité de l'égaliseur

5.5.1.1 FPGA XCV2P40-7

La technologie choisie pour effectuer la synthèse et le placement routage est une puce programmable de type FPGA (Field Programmable Gate Array). Pour couvrir toute la complexité matérielle d'un égaliseur fréquentiel programmable et configurable de 4 à 256 coefficients, nous avons sélectionné le VirtexII Pro numéro XCV2P40-7 de Xilinx (Xilinx, 2005). Le suffixe 7 indique le plus haut grade de vitesse disponible. Ce FPGA possède 192 multiplicateurs embarqués de 18 par 18 bits (BMULT). Il est également munis de 192 mémoires dédiées (BRAM) de 18 kilo bits que l'on peut configurer manuellement ou automatiquement par le compilateur en mode simple ou double ports avec la disposition 16Kx1 bits jusqu'à 512x36 bits. Les unités logiques que l'on nomme *slices* (voir figure 50) sont munies de générateurs de fonction avec quatre entrées, de logiques de retenues, de multiplexeurs et de deux registres. On peut programmer les générateurs de fonctions sous forme de tables (LUT) et/ou sous forme de mémoire de 16 bits et/ou sous forme de registre à décalage de 16 bits. Le XCV2P40-7 est construit avec 19392 *slices*.

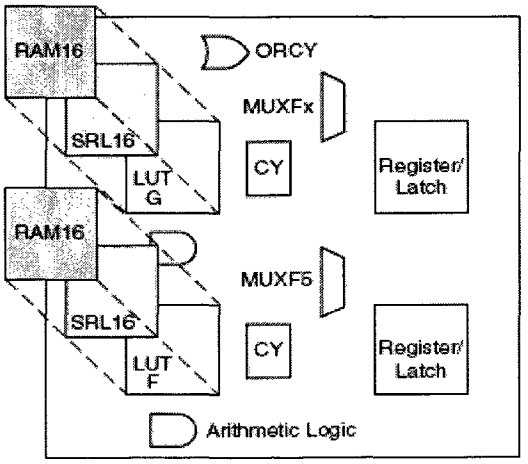


Figure 63 Configuration d'une *slice* d'un Virtex II Pro. (Tiré de (Xilinx, 2005))

Plusieurs autres ressources sont disponibles telles que des processeurs embarqués PowerPC RISC et des bus RocketIO. Toutefois, nous ne les utilisons pas dans ce travail.

5.5.1.2 Estimation théorique des ressources requises

Avec les équations 5.2 et 5.3, on peut calculer le nombre théorique de multiplicateurs requis pour chaque configuration de l'égaliseur FBMMA3M. En effet, nous savons qu'un multiplicateur complexe est requis pour le filtrage et un autre pour l'inter corrélation. Ce qui nous porte à huit multiplicateurs dédiés pour le traitement de l'adaptation. Sachant que trois modules de transformée de Fourier sont requis pour le mode non contraint et cinq pour le mode contraint, nous pouvons écrire:

$$cplxtyMult_{FBMMA3M(U)} = 3\left(2\log_2 2N - (2_{OU}4)\right) + 8\,, \tag{5.5}$$

$$cplxtyMult_{FBMMA3M(C)} = 5\left(2\log_2 2N - (2_{OU}4)\right) + 8\,. \tag{5.6}$$

Ainsi, lorsque les échantillons sont quantifiés pour tout le processus sous la barre des 18 bits, le nombre de multiplicateurs requis suivra l'équation 5.5 lorsque le gradient n'est pas contraint et l'équation 5.1 lorsque le gradient est contraint.

Au niveau du nombre de BRAM, on peut s'attendre à ce que les modules de l'erreur MMA et de décision en consomment deux (partie réelle et imaginaire) chacun. On peut également prévoir que le module *sfifo* en consommera deux puisqu'il requiert $7N$ cases mémoires. Les modules de FFT/IFFT consomment des BRAMS à partir de 128 points. La quantité dépend de la quantification des étages internes ainsi que de la configuration de l'ordonnancement, (Grandmaison, 2005).

Et le nombre de *slices* sera supérieur à ce que consomment les FFT/IFFT tous ensemble. Lorsque les mémoires requises sont de l'ordre de *N* cases mémoires, elles sont fabriquées avec des *slices* configurées en LUTs. On peut donc s'attendre à un ratio plus élevé de consommation de *slices* pour les configurations d'égaliseurs à petite quantité de coefficients, soit de 4 à 32 coefficients temporels.

5.5.1.3 Ressources requises après placement routage

Nous avons utilisé l'outil de synthèse Synplify Pro pour traduire le code VHDL en circuit dans le FPGA XCV2P40-7. Aucune des options n'a été sélectionnée (*pipelining, retiming, ressources sharing, modular desing, FSM compiler*) dans le but d'accélérer la vitesse de compilation. Seule la contrainte d'horloge a été imposée. Nous l'avons fixée à 5 nano secondes, soit 200 MHz. Pour les configurations 4 à 32 coefficients temporels, cette contrainte a été atteinte après placement et routage. Cependant, pour les configurations plus élevées, la vitesse maximale est d'environ 150 Mhz (voir chapitre 4). Le goulot d'étranglement se situe au niveau de la transformée de Fourier inverse avec entrée des échantillons en ordre inverse (*r22pc2* et *r22pc4*). Un des papillons retarde le traitement. En effet, puisque la quantification des multiplicateurs de la FFT/IFFT (r22pc2 et r22pc4) est supérieure à 18 bits, le compilateur utilise deux blocs dédiés de multiplicateurs 18x18. Cette architecture réduit les performances de dispersion du signal dans le FPGA et donc diminue la vitesse maximale de traitement du noyau d'égaliseur. Toutefois, 150 MHz constitue une fréquence amplement suffisante pour atteindre l'objectif de 20 Msym/s. En effet, si on considère qu'il faut quadrupler la fréquence d'horloge par rapport au taux de symboles pour accommoder l'échantillonnage fractionnaire à Tsym/2 ainsi que pour accomplir l'opération de chevauchement et sauvegarde à 50%, on obtient un égaliseur qui peut traiter un taux de symboles de l'ordre de 37.5 MHz.

Le placement et routage ont été accomplis à l'aide de l'outil Xilinx ISE Foundation version 6.3 avec les options d'optimisation de la vitesse d'opération maximale. La figure

48 rapporte les ressources matérielles requises en terme de BMULT, de BRAM et de *slices* pour toutes les configurations possibles. Les indices C et U indiquent respectivement les résultats pour un égaliseur fréquentiel à gradient contraint et non contraint. Nous avons juxtaposé les résultats de ressources BMULT théoriques requises. On observe que jusqu'à 64 coefficients fréquentiels, les résultats pratiques concordent avec la théorie. Au delà de cette quantité, on obtient des résultats plus élevés puisque la quantification des modules *r22pc3*, *r22pc5*, *s_cplx_mult1* et *s_cplx_mult_2* est posée plus élevée que dix-huit bits (voir tableau X, chapitre quatre). Ceci fait en sorte que le compilateur utilise deux multiplicateurs embarqués par multiplication réelle. Le nombre de multiplicateurs réellement implémentés dans le XCV2P40-7 s'obtient donc par:

$$cplxtyMult_{FBMMA3M(U)} = (3+1)\left(2\log_2 2N - (2_{OU}4)\right) + 2*8 \tag{5.7}$$

et

$$cplxtyMult_{FBMMA3M(C)} = (5+2)\left(2\log_2 2N - (2_{OU}4)\right) + 2*8\,. \tag{5.8}$$

Ainsi, tout se passe comme si on avait besoin de sept et quatre transformées de Fourier pour l'algorithme contraint et non contraint respectivement. De plus, deux multiplicateurs complexes de plus sont requis pour le filtrage et l'inter corrélation. Encore une fois, on observe que l'algorithme non contraint permet de réduire considérablement le nombre de ressources matérielles requises. Cette observation est également valide au niveau du nombre BRAM requises. Avec huit et seize coefficients fréquentiels, ce sont seulement les tables de référence du module de décision ainsi que celles de l'erreur MMA qui consomme des BRAMS (deux chacune). À trente-deux et soixante-quatre coefficients fréquentiels, s'ajoutent deux autres BRAMS pour l'implémentation du module *sfifo*. Puis, à partir de 128 coefficients fréquentiels, les FFT/IFFT consomment des blocs de mémoires embarqués. La quantité dépend de la quantification et de la taille, (Grandmaison, 2005). À noter que la non linéarité que l'on observe sur la courbe de consommation en *slices* est justement due au fait que les FFT/IFFT se mettent à employer des BRAM.

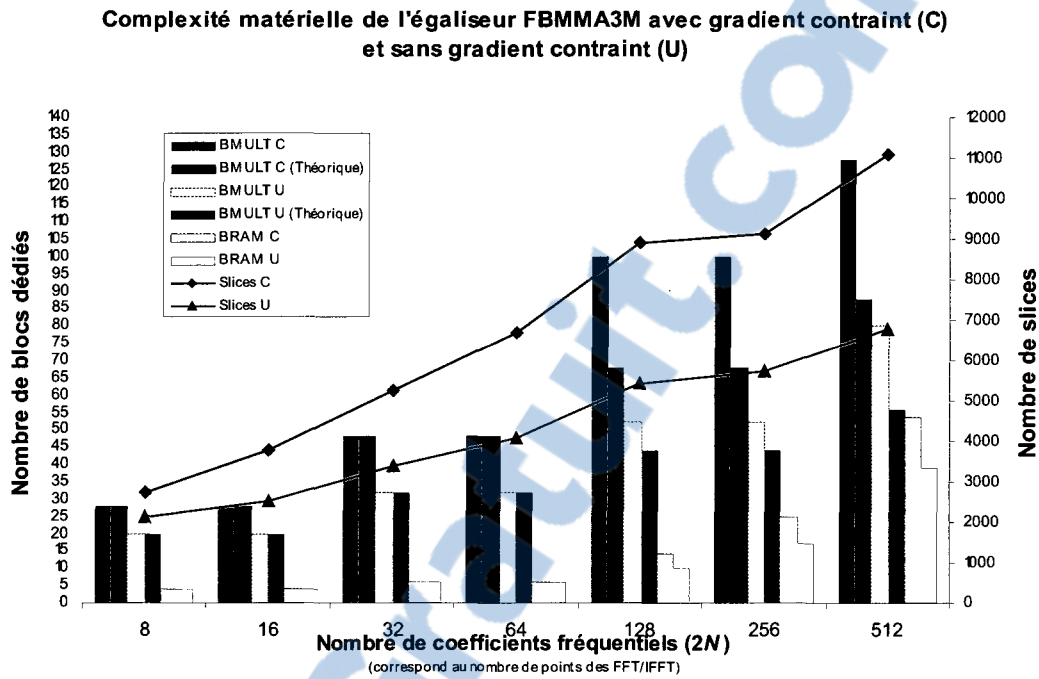


Figure 64 Complexité matérielle de l'égaliseur FBMMA3M avec gradient contraint et non contraint.

5.6 Quelques résultats de performances de l'égaliseur FBMMA3M

Le haut degré de configurabilité de l'égaliseur FBMMA3M fait en sorte qu'il y a un très large éventail de résultats que l'on pourrait présenter. Afin de couvrir le plus possible les différentes fonctions du noyau implémenté, nous avons choisi d'exposer la constellation en sortie de l'égaliseur, les coefficients après convergence ainsi que l'erreur quadratique moyenne combinée MMA-LMS. Les figures 65, 66 et 67, affichent les courbes mentionnées ci haut après convergence de l'égaliseur lorsqu'il est soumis au canal SUI-3 présenté au chapitre trois avec un SNR de l'ordre de 35 dB. La quantification de l'égaliseur FBMMA3M est identique à ce qui a été présenté au chapitre quatre. Dix bits sont affectés à la fraction de tous les points de quantifications à l'exception des modules de décision et de l'erreur MMA. Ces derniers sont affectés de huit bits de fraction au niveau du bus d'adresse de la table de référence.

Rapport-Gratuit.com

Pour chaque figure présentée, la configuration de l'égaliseur est rapportée au tableau XV.

Tableau XV

Configuration de l'égaliseur FBMMA3M pour les figures 65, 66 et 67

	Taux de symboles (MHz)	**MQAM**	**Contraint (C) ou non contraint (U)**	**mu**	**N**
Figure 65	**1**	**4**	**C et U**	2^{-4}	**4**
Figure 66	**10**	**16**	**C et U**	2^{-8}	**64**
Figure 67	**37**	**64**	**C et U**	2^{-10}	**256**

Avec quatre coefficients (figure 65), on observe que l'erreur quadratique moyenne diffère légèrement de ce qu'on obtient avec le modèle en point flottant. Nous expliquons ce phénomène par le fait que la véritable latence est plus grande que ce qui a été simulé. En effet, chaque bloc de FFT/IFFT consomme $2N + \log_2 2N$ cycles d'horloges. De plus, la latence des autres blocs tels que le multiplicateur et la gestion de l'erreur tourne autour de sept cycles d'horloge. Ceci fait en sorte que le délai D est de l'ordre de $8N$ et non $5N$ tel que nous l'avons simulé. La différence est moins marquée lorsqu'on configure l'égaliseur en mode non contraint puisqu'il n'y a que trois modules de FFT/IFFT.

Au premier chapitre, nous avons vu que la dispersion des symboles est directement liée au taux de symboles. Ainsi, à 1 MHz, un égaliseur de quatre coefficient peut à la limite couvrir le canal SUI-3. Toutefois, si on augmente la vitesse de transmission par un facteur dix, les échos affectent au moins les vingt symboles de part et d'autre du coefficient centrale. Ainsi, en posant un égaliseur de soixante quatre coefficients, on peut espérer réduire l'EQM suffisamment pour établir une conversation entre deux points de communications. C'est effectivement ce que nous illustrons à la figure 66.

À la figure 67, nous présentons les résultats de convergence lorsqu'on pose le taux de transmission à 37 Msym/s, tout près de la vitesse maximale que l'égaliseur peut accommoder. La courbe de l'EQM en sortie de l'égaliseur FBMMA3M implémenté est presque indissociable de la courbe résultante du simulateur en point flottant. Ceci confirme que la quantification a été adéquatement déterminée.

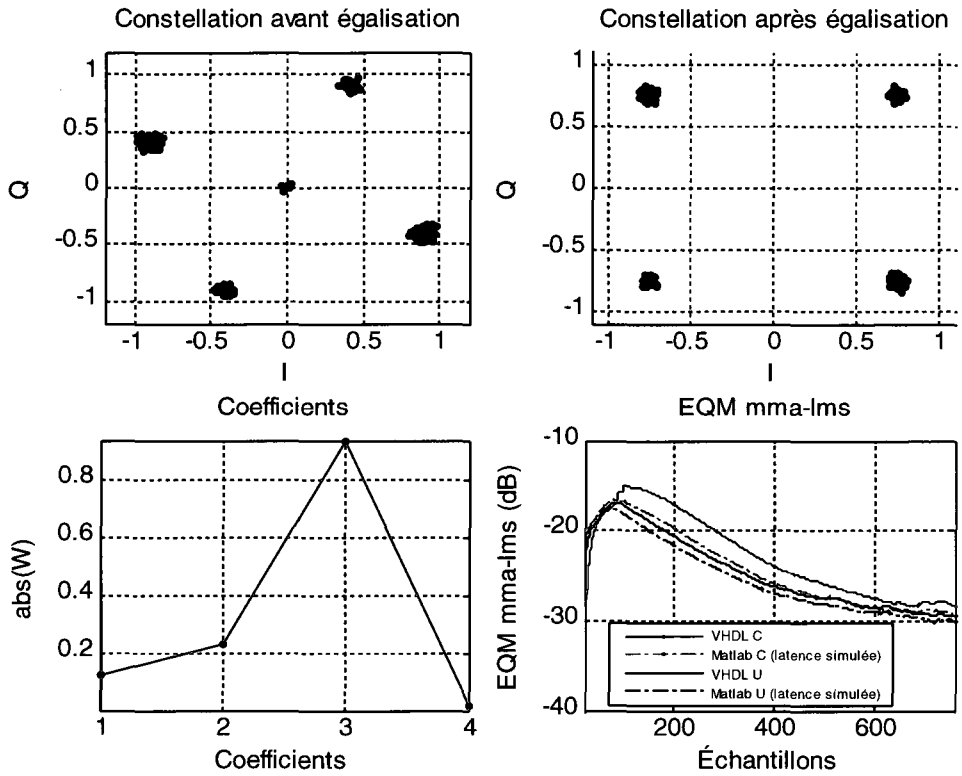


Figure 65 Résultats en sortie du module FBMMA3M avec une constellation de 4-QAM et $N = 4$

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

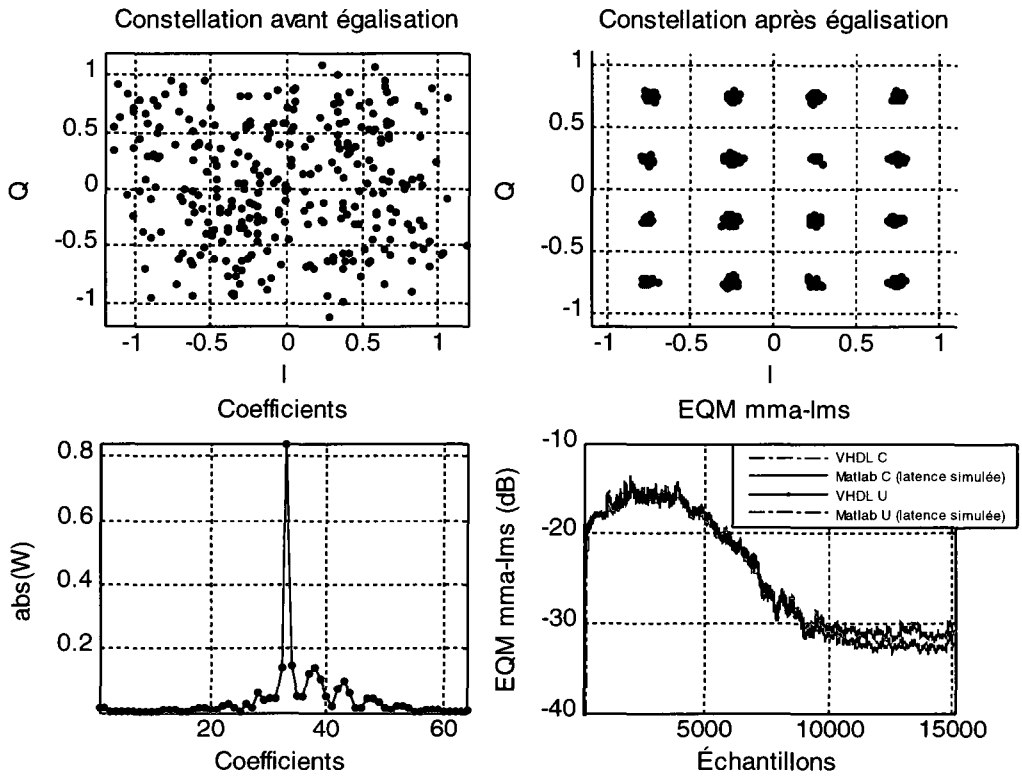


Figure 66 Résultats en sortie du module FBMMA3M avec une constellation de 16-QAM et $N = 64$

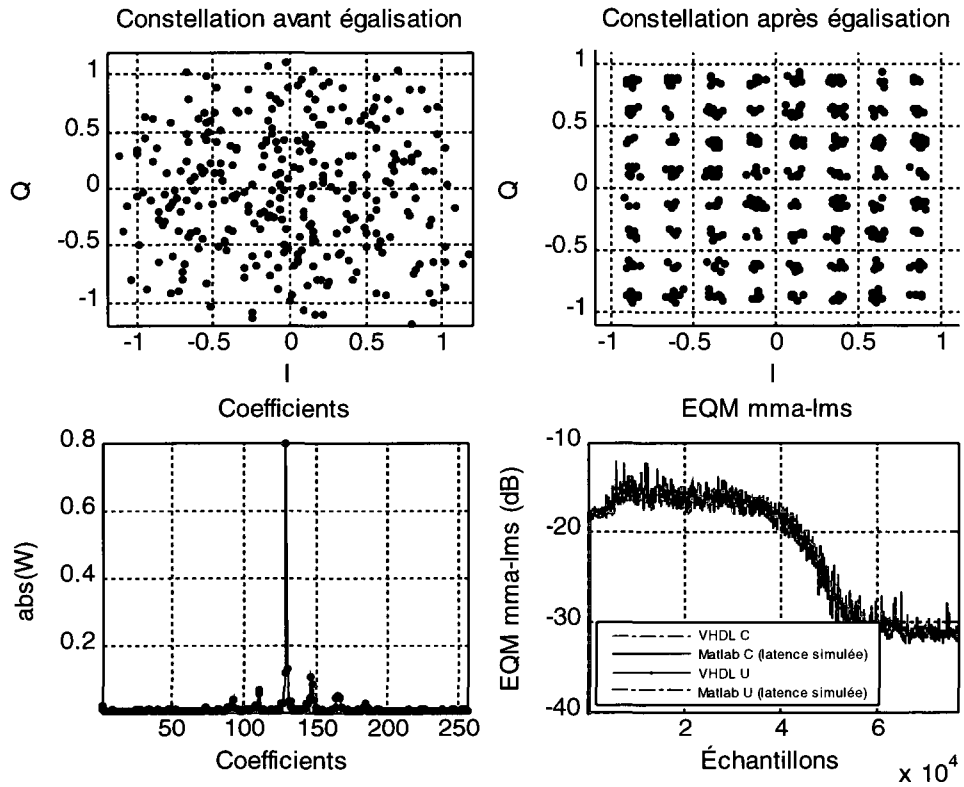


Figure 67 Résultats en sortie du module FBMMA3M avec une constellation de 64-QAM et $N = 256$

5.7 Conclusion

Dans ce chapitre, nous avons présenté les stratégies d'implémentation matérielle de l'égaliseur FBMMA3M. Nous avons également exposé les détails de conception de chaque sous modules de l'égaliseur fréquentiel.

Les résultats après placement et routage démontrent que les objectifs cités en début de chapitres ont été atteints. En effet, l'égaliseur FBMMA3M est configurable de 4 à 256 coefficients temporels. On peut le programmer pour qu'il accommode les modulations 4, 16 et 64 QAM. Puis considérant la lourdeur de l'algorithme FBLMS, nous avons réussi à implémenter un noyau capable de fonctionner en temps réel à plus de 150 MHz, soit 37.5 Msym/s. La stratégie de traitement entièrement sérielle nous a également permis de réduire les exigences en terme de ressources matérielles. Surtout en terme de multiplicateurs embarqués.

Finalement, nous avons présenté des résultats de convergence qui prouvent la fonctionnalité de l'égaliseur FBMMA3M tout en respectant une précision de calcul très élevée par rapport à notre modèle de référence en point flottant.

CONCLUSION

Afin de réduire les interférences entre symboles causées par les multiples chemins empruntés par les signaux de communications entre deux radios, on a souvent recourt à des filtres adaptatifs que l'on nomme égaliseurs. Pour un canal donné, le taux de transmission affecte directement la configuration nécessaire de l'égaliseur. En effet, plus le nombre de symboles par seconde est élevé, plus grande doit être la quantité de coefficients d'égaliseur pour couvrir toute la plage de symboles dispersés. Par surcroît, plus une constellation est dense, plus l'IES doit être réduite pour rencontrer les standards de taux d'erreur. Pour réduire davantage les interférences du canal, il faut augmenter le nombre de coefficients d'égaliseur. Dans ce travail, nous avons vu que l'égalisation fréquentielle basée sur l'algorithme FBLMS constitue une excellente alternative pour répondre à ces besoins. Ainsi, l'objectif de ce projet de maîtrise est de concevoir, simuler et réaliser un égaliseur fréquentiel configurable de 4 à 256 coefficients et capable d'accommoder les modulations 4, 16 et 64 QAM. L'égaliseur FBMMA3M que nous avons implémenté dans le cadre de cette recherche répond à tous ces critères.

Dans un premier temps, on a développé un environnement de simulation en point flottant pour placer l'égaliseur FBMMA3M dans un contexte de télécommunication le plus réaliste possible. Pour émuler la transmission à multiples chemins, nous avons employé un canal SUI. La première étape de conception consistait à déterminer une architecture d'égaliseur implémentable en langage VHDL avec le moins de ressources et le plus haut taux de transmission possible. Plusieurs structures ont été évaluées. Celle que nous avons conservée peut être employé dans trois différents modes de calcul de l'erreur. Le premier mode est dit aveugle puisqu'il ne nécessite aucune séquence d'entraînement. Ce dernier est très populaire dans la littérature puisqu'il permet de maximiser l'efficacité spectrale. Le second mode, décision dirigée, peut être employé conjointement avec le premier mode pour accélérer la convergence des coefficients. L'approche classique basée sur une séquence d'entraînement représente le troisième mode de calcul de l'erreur.

La seconde étape de conception consistait à convertir les points de quantification en point fixe avec une représentation en complément à deux. Pour ce faire, nous avons implémenté un modèle de simulation encore plus près ce qui sera réalisé en circuits logiques. En effet, sachant qu'un des objectifs principaux était d'atteindre le taux de transmission le plus élevé possible, nous avons choisi de modifier l'algorithme d'adaptation des coefficients en insérant un délai au niveau du calcul du gradient. Ce délai permettait ainsi d'envisager une architecture dont le traitement peut être accompli de manière sérielle.

L'échantillonnage des données à l'entrée de l'égaliseur est accompli à deux fois le taux de symboles. Les fonctions de FFT/IFFT comprises dans l'égaliseur FBMMA3M sont réalisées avec le module configurable $R2^2PC$, (Grandmaison, 2005). Les paramètres configurables et programmables de l'égaliseur FBMMA3M sont nombreux. Avant compilation, il est possible de sélectionner l'algorithme avec ou sans contrainte du gradient. Les analyses effectuées nous ont démontrées que l'action de contraindre le gradient n'a pas beaucoup d'impact sur la convergence des coefficients avec un canal SUI. De plus, l'algorithme UFBLMS requiert deux modules de FFT/IFFT de moins. Toujours avant compilation, il est possible de choisir un égaliseur de 4 à 256 coefficients. La quantification des tables de références pour le module de décision ainsi que le module de calcul de l'erreur MMA sont également à la portée de l'usager avant la synthèse. Pendant le traitement, l'usager peut charger le contenu de ces tables de référence pour accommoder des modulations M-QAM. Dans le cadre de notre travail, nous avons vérifié le comportement de l'égaliseur avec des modulations 4, 16 et 64 QAM. Toutefois, il est possible de charger des constellations plus denses. Le *pas d'adaptation* peut être modifié en tout temps pendant le traitement. Quatre niveaux sont disponibles. Le processus d'adaptation peut être contrôlé et les coefficients peuvent être chargés et rechargés en tout temps en cours de traitement. Finalement, trois différents modes de calcul de l'erreur peuvent être choisi pendant le traitement. Soit le mode aveugle, le mode auto dirigé et le mode entraîné. À noter que le mode aveugle est utilisé

conjointement avec le mode auto dirigé pour augmenter le taux de convergence des coefficients.

Ce mémoire contribue à l'avancement des technologies de télécommunications sans fil. Tout d'abord parce que nous avons modifié l'algorithme FBLMS pour créer un égaliseur fréquentiel que l'on peut programmer dans un mode de fonctionnement aveugle, dirigé ou entraîné. Au moment où nous avons accompli ce projet, nous n'avons trouvé aucun article à ce sujet dans la littérature. Par surcroît, l'architecture sérielle de l'égaliseur FBMMA3M fait en sorte que la consommation de multiplicateur est remarquablement inférieure à ce que les équations théoriques proposent. De ce fait, l'égaliseur FBMMA3M offre une complexité matérielle très faible relativement à la quantité de coefficient requis par l'usager. De plus, accompagné à ce module, nous incluons un modèle en point flottant de communication numérique en bande de base soumis à un canal SUI. Ce modèle permet de caractériser l'égaliseur FBMMA3M face à divers contextes de télécommunications.

Pour compléter ce travail, il serait intéressant d'intégrer le noyau d'égaliseur FBMMA3M dans une radio pour vérifier et tester son comportement dans un contexte de télécommunication réel.

Le haut niveau de programmabilité de l'égaliseur FBMMA3M pourrait s'avéré attrayant dans un contexte de radio logicielle. Sa structure portable ouvre la porte à de nombreuses applications implémentées sous différentes plateformes. Parmi les quelques innovations possibles, on pourrait envisager la modulation adaptative pour diminuer les probabilités d'erreurs. Il serait également intéressant de combiner l'égalisation fréquentielle avec la technologie OFDM. En effet, une des problématiques rencontrées avec les modems OFDM est la très grande variation de puissance à l'émission. Les amplificateurs de puissance doivent être très performants. Afin de réduire les coûts inhérents à cette réalité, on pourrait installer un récepteur à simple porteuse muni d'un égaliseur FBMMA3M (SC-FDE) ainsi qu'un émetteur OFDM dans une station de base et

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

munir les stations clients d'émetteurs SC-FDE et de récepteurs OFDM, (Falconeer, 2002). Le préfixe cyclique (CPI) utilisé pour palier aux IES avec les systèmes OFDM est semblable à la méthode de chevauchement et sauvegarde que nous utilisons avec l'égaliseur FBMMA3M. On pourrait donc en tirer profit pour accomplir la combinaison. La figure 1 illustre le principe d'un système de communication numérique conjoint OFDM/SC-FDE.

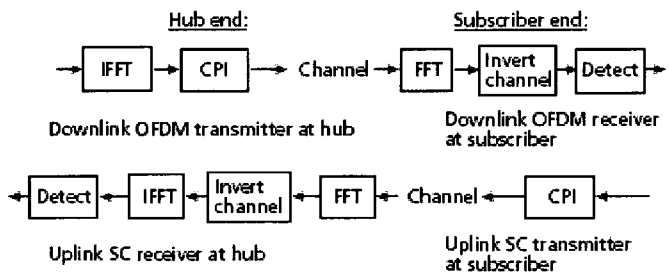


Figure 1 Coexistence OFDM-SC-FDE, tiré de (Falconeer, 2002).

BIBLIOGRAPHIE

Altera Corporation.[En ligne]. www.altera.com (Consulté le 2 oct. 2005).

Bose, T. (2004). *Digital signal and image processing.* (1st ed). John Wiley & Sons, Inc.

Clark, G.; Mitra, S.; Parker, S. (1981). Block implementation of adaptive digital filters. *IEEE transaction on circuits and systems*. 28(6), 584-592.

Dmochowski, P.A.; McLane, P.J. (2001). Frequency domain equalization for high data rate multipath channels. *IEEE transaction on communications, Computers and signal processing.* Vol. 2, 534-537.

Dumais, P., Ammari, M.L., Gagnon, F. & Thibeault, C. (2004). Multi-equalization a powerful adaptive filtering for time varying wireless channels. *IEEE transaction on Communication,* Vol. 6, 1744-1747.

Falconer, D.; Ariyavisitakul, S.L,; Benyamin-Seeyar, A.; Eidson, B. (2002). Frequency domain equalization for single-carrier broadband wireless systems. *IEEE communication magazine.* 40(4), 58-66.

Farhang-Boroujeny, B. (1992). Quantization effects in transform domain adaptive filters. *Singapore ICCS/ISITA 92. 'Communications on the move'.* 1. 25-29.

Farhang-Boroujeny, B. (1998). *Adaptive filters: Theory and application*. John Toronto, ON. Wiley & Sons Ltd.

Farhang-Boroujeny, B. (2000). Analysis of the frequency domain block lms algorithm. *IEEE transaction on speech and signal processing.* 48(8), 2332-2342.

Ferrera, E. (1980). Fast implementation of LMS adaptive filters. Transaction on *Acoustics, Speech, and Signal processing, IEEE.* 28(4), 474-475.

Ghorbani & Sheikhan, (1991). The effect of Solid State Power Amplifiers (SSPAs) Nonlinearities on MPSK and M-QAM Signal Transmission, *Conference on Digital Processing of Signals in Communication*, 12(9), 193-197.

Godard, D. (1980). Self-recovering equalization and carrier traking in two dimensional data communication. *IEEE Transaction on communications*. 28(11), 1867-1875.

Grandmaison, M-È, (2005). Conception d'un module reconfigurable de FFT. École de technologie supérieure, Montréal.

Rapport-Gratuit.com

Hogenauer, E. (1981). An economical class of digital filters for decimation and interpolation, *IEEE Transactions on Acoustics, Speech, and Signal Processing, 29(2),* 155-162

IEEE 802.16 (2003). *[en ligne]* Broadband Wireless Access Working Group, http://ieee802.org/16/, *Channel Models for Fixed Wireless Applications.*

Joshi, R. B.; Daneshrad, B.; Samueli, H. (1992); A VLSI architecture for a single chip 5-Mbaud QAM receiver. *Global telecommunications conference, IEEE*, 13, 1265-1268.

Lee, J. & Un, C. (1989). Performance analysis of frequency domain block LMS adaptive digital filters. *IEEE transaction on circuits and systems.* 36(2),173-189.

Long, G.; Ling, F.; Proakis, J.G. (1989). The LMS algorithm with delayed coefficient adaptation. *IEEE transaction on signal processing.* 37(9),1397-1405.

Mansour, D.& Gray, A. (1982). Unconstrained frequency domain adaptive filter. *IEEE transaction on signal processing.* 30(5), 726-734.

Nyquist, H. (2002). Certain topics in telegraph transmission theory. *Proceedings of the IEEE*, 90(2), 280-305.

Noguchi, T.;Daido, Y.;Nossek, J., (1986). Modulation techniques for microwave digital radio. *Communications Magazine, IEEE*, 24(10)

Oppenheim & Schafer, (1975). *Digital signal processing.* Englewood Cliffs, NJ. Prentice Hall, 1975.

Proakis, J. G. (2001). *Digital communications* (4th ed.). New York, N.Y.: McGraw-Hill.

Proakis, J. G., & Manolakis, D. G. (1996). *Digital signal processing : principles, algorithms, and applications* (3rd ed.). Upper Saddle River, N.J.: Prentice-Hall.

Qureshi, S.U.H. (1985). Adaptive equalization. *Proceedings of the IEEE,* 73(9), 1349-1387

Rappaport, S. Theodore (2002). *Wireless communications* (2d ed.). Upper Saddle River, N.J.: Prentice Hall PTR.

Shynk, J. J. (1992). Frequency-domain and multirate adaptive filtering. *IEEE Signal Processing Magazine,* 9(1), 14-37.

Stremler, Ferrel G. (1992). *Introduction to communication systems* (3th ed.). New York, N.Y.: Addison-Wesley Publishing Company, Inc.

VHDL tutorial.[En ligne]. http://www.vhdl-online.de/tutorial/ (Consulté le 15 sept. 2005).

Weerackody, V.; Kassam, S.A. (1994). Dual-mode type algorithms for blind equalization. *IEEE transactions on communications.* 42(1), 22-28.

Widrow & Sterns, (1985). *Adaptive signal processing.* Prentice-Hall. Englewood Cliffs. NJ.

Xilinx, Inc. [En ligne]. www.xilinx.com (Consulté le 5 oct. 2005).

Yang Yoon Gi; Cho Nam Ik; Lee Sang Uk. (1995). Fast blind equalization by using frequency domain block constant modulus algorithm. *IEEE transaction on circuits and systems.* 2, 1003-1006.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES