HAL
open science

Conception et réalisation de la partie numérique d'un simulateur matériel pour les canaux de propagation MIMO

Sylvie Picol

▶ **To cite this version:**

Sylvie Picol. Conception et réalisation de la partie numérique d'un simulateur matériel pour les canaux de propagation MIMO. Sciences de l'ingénieur [physics]. INSA de Rennes, 2007. Français. NNT : . tel-00339423

HAL Id: tel-00339423

https://theses.hal.science/tel-00339423v1

Submitted on 17 Nov 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N° d'ordre : D 07 - 28

Thèse

présentée devant

l'**Institut National des Sciences Appliquées de Rennes**

pour obtenir le titre de

Docteur

spécialité : *Électronique*

Conception et réalisation de la partie numérique d'un simulateur matériel pour les canaux de propagation MIMO

par

Sylvie Picol

Soutenue le 5 décembre 2007 devant la commission d'Examen

Composition du jury

Rapporteurs

Emmanuel BOUTILLON	Professeur à l'Université de Bretagne Sud
Rodolphe VAUZELLE	Professeur à l'Université de Poitiers

Examinateurs

Pierre DEGAUQUE	Professeur à l'Université de Lille
Samuel GUILLOUARD	Activity Manager, THOMSON Rennes
Patrice PAJUSCO	Ingénieur, France Télécom Orange Labs/RESA/NET/WEP
Ghaïs EL ZEIN	Professeur des Universités à l'INSA de Rennes
Dominique HOUZET	Professeur des Universités à l'INPG/ENSERG (Grenoble) - GIPSA-Lab
Gheorghe ZAHARIA	Maître de Conférences à l'INSA de Rennes

Institut d'Electronique et de Télécommunications de Rennes
Institut National des Sciences Appliquées de Rennes

à mon Papa

à Thomas

Le savant n'est pas l'homme qui fournit les vraies réponses ; c'est celui qui pose les vraies questions

Claude Lévi-Strauss (1908)

Ne me dites pas que ce problème est difficile. S'il n'était pas difficile, ce ne serait pas un problème

Ferdinand Foch (1851-1929)

En science, la phrase la plus excitante à entendre, celle qui annonce de nouvelles découvertes, n'est pas « Eureka », mais plutôt « Tiens, c'est marrant... »

Isaac Asimov(1920-1992)

Des chercheurs qui cherchent, on en trouve. Des chercheurs qui trouvent, on en cherche !

Charles de Gaulle(1890-1969)

Remerciements

Le travail de thèse, qui donne lieu à ce mémoire, s'est déroulé de la meilleure des façons au sein du groupe Communication-Propagation-Radar du laboratoire IETR à l'INSA de Rennes. Il est donc naturel de commencer par remercier notamment le directeur-adjoint du laboratoire, qui fut également co-directeur de cette thèse, Ghaïs el Zein. Merci de m'avoir accueillie dans ce laboratoire, tout d'abord sous la coupe de Dominique Houzet pour mon stage de fin d'étude de DEA, puis de m'avoir confié ce sujet de thèse avec une grande liberté dans la conduite de mes recherches et des commentaires toujours enrichissants. Un grand merci aussi à Dominique Houzet pour avoir co-dirigé cette thèse, malgré notre éloignement dû à sa fonction de Professeur à l'INPG de Grenoble. Toujours à l'écoute, il m'a fait profiter de sa grande expérience et de ses connaissances dans le domaine des composants programmables.

Je voudrais ensuite remercier tout ceux qui ont accepté de faire partie de mon jury : Monsieur Pierre Degauque, Professeur à l'Université de Lille, qui m'a fait l'honneur de le présider, Messieurs Emmanuel Boutillon, Professeur à l'Université de Bretagne Sud, et Rodolphe Vauzelle, Professeur à l'Université de Poitiers, qui ont assuré la lourde tâche de rapporteur de mon travail en m'ayant fourni de nombreuses remarques contructives, Messieurs Patrice Pajusco, Ingénieur France Télécom Orange et Samuel Guillouard, Activity Manager à THOMSON Rennes, qui ont su apporter une dimension industrielle à ce sujet de thèse et pour avoir accepté de juger ce travail en tant qu'examinateur. Merci à tous pour l'attention que vous avez accordé à la lecture de ce mémoire.

Lors de la première année de cette thèse, j'ai eu la chance de commencer à travailler avec Thomas Quiniou. Il a élaboré le sujet avec Ghaïs et m'a donné les premières orientations de mon travail. Ayant eu l'occasion de partir travailler à Nouméa par la suite, il a alors quitté l'INSA mais je le remercie ici jusqu'aux antipodes.

La suite de mon travail a alors été également suivi par Gheorghe Zaharia, Maître de Conférences à l'INSA de Rennes, et auquel j'adresse ma profonde reconnaissance pour m'avoir aidé tout au long du déroulement de cette thèse. Je ne saurai dire à quel point tes conseils et ton expertise sur de nombreux sujets m'ont apporté pour le bon déroulement

de ce travail. Pour ta constante disponibilité, ta patience et ta bonne humeur, « mulţu mesc » Gheorghe ! !

Par ailleurs, j'ai eu l'occasion pendant mon travail de solliciter l'aide de nombreux stagiaires qui m'ont consacré beaucoup de leur savoir et de leur temps. Je remercie Jose Manuel Aguilar Bruno, Antonio Hernandez Hernandez, Pepe Muñoz Bello dont les stages de fin d'études d'ingénieur font partie intégrante de cette thèse. J'ai eu ensuite le privilège de travailler avec Yvan Kokar dont le travail d'ingénierie a permis de réaliser nombre de mes travaux. Merci encore pour ton expertise, ta patience et ton aide jusqu'aux dernières minutes !

Parce qu'un travail de thèse ne peut être que le résultat de l'engagement de toute une équipe, je voudrais remercier collectivement tous les membres du groupe CPR que j'ai vu se renouveler au fil des années : tout d'abord, les plus anciens qui m'ont permis un bonne intégration à l'équipe : Philippe, Ronan, les Stéphane, Arnaud, Jean-Michel, Matthieu qui reste fidèle au poste, sans oublié Fabrice ! Puis mes acolytes : Florent, Wilfried, Julien, David et tout « ceux d'en haut »... Merci aussi aux suivants, à votre tour de présenter vos travaux : Minh, Antoine, Christophe, bon courage et bonne continuation. Merci aussi à tout le personnel d'enseignement et de recherche ainsi qu'au personnel administratif et technique, notamment pour leur aide dans la gestion de nos déplacements.

De plus, je souhaite adresser un vif remerciement à mon professeur d'électronique de l'IUT de Rennes, Hélène Larzul, qui aimait à répéter : « On peut tout faire avec un DUT GEII ! ! » Et bien oui, la preuve...

Je n'oublie pas non plus tout mes amis en région parisienne, rennaise, brestoise, quimperloise, en Nouvelle-Calédonie, Guadeloupe et aux Etats-Unis. Merci pour votre soutien et vos encouragements !

Enfin, je remercie mes parents qui m'ont toujours encourager dans mes études et ma grande sœur Sophie qui m'a montré le chemin. Plus généralement, merci à toute ma famille, merci également à la famille de Thomas, pour m'avoir soutenu tout au long de ces années d'études dans le supérieur. Je remercie aussi tout particulièrement Thomas, mon chéri et futur époux, pour avoir su me soutenir, me rassurer, me donner confiance en moi et m'encourager durant ces trois années et demi. Cette thèse doublerait de volume si je le remerciais comme il se doit. Merci Tom d'avoir tout simplement su m'attendre...

Table des matières

Remerciements **v**

Table des matières **vii**

Introduction **1**

1 Canal de propagation radioélectrique **5**

1.1 Caractérisation du canal de propagation 6

1.1.1 Phénomènes physiques de la propagation radioélectrique 7

1.1.1.1 Le bruit radioélectrique 7

1.1.1.2 Les mécanismes de la propagation 7

1.1.1.3 Les distorsions du signal 10

1.1.2 Canal de propagation 15

1.2 Modèles de canaux de propagation 18

1.2.1 Modèles déterministes 19

1.2.2 Modèles statistiques 20

1.2.3 Modèles hybrides 20

1.3 Etat de l'art des simulateurs de canaux MIMO 21

1.4 Conclusion 27

2 Simulateur matériel de propagation **29**

2.1 Cahier des charges du simulateur matériel 30

2.1.1 Description du bloc radiofréquence 32

2.1.2 Description du bloc « Modèles de canal » 34

2.1.3 Description du bloc « Partie numérique » 38

2.1.3.1 Deux architectures envisagées 38

2.1.3.2 Choix des paramètres 40

2.2 Plate-forme de prototypage 42

2.2.1 Choix du composant programmable 42

2.2.2 Choix des cartes électroniques 44

2.2.3 Génération et distribution de l'horloge 47
2.3 Conclusion 48

3 Approche fréquentielle 51
3.1 Description des différents blocs de la chaîne fréquentielle 52
3.1.1 Etude de la transformée de Fourier rapide, directe et inverse 52
3.1.1.1 Description de l'algorithme de la transformée de Fourier discrète 53
3.1.1.2 Etude de la FFT de Xilinx 56
3.1.2 Etude du bloc Mémoire 59
3.1.2.1 Volume de données et choix du bus de transfert 59
3.1.2.2 Chargement du bloc Mémoire 60
3.1.3 Etude du multiplieur complexe 61
3.1.4 Etude de la troncature 62
3.2 Mise en œuvre de la chaîne dans le domaine fréquentiel 64
3.2.1 Justification du signal de test 65
3.2.2 Etude de la chaîne fréquentielle 67
3.2.3 Résultats et précision de calcul 75
3.3 Conclusion 86

4 Approche temporelle 89
4.1 Description des différents blocs de la chaîne temporelle 90
4.1.1 Etude de la convolution 91
4.1.1.1 Description de l'algorithme 91
4.1.1.2 Etude du filtre FIR programmé 92
4.1.2 Etude du bloc Mémoire 93
4.1.2.1 Volume de données 93
4.1.2.2 Chargement du bloc Mémoire 94
4.2 Mise en œuvre de la chaîne dans le domaine temporel 95
4.2.1 Quantification du signal d'entrée et de la réponse impulsionnelle . . 96
4.2.2 Filtre FIR 64 96
4.2.3 Troncature et convertisseur numérique-analogique 98
4.2.4 Résultats et précision de calcul 99
4.2.4.1 Simulation du filtre FIR 64 99
4.2.4.2 Expérimentation du filtre FIR 64 sur carte Nallatech 103
4.3 Solution de substitution de l'approche fréquentielle 104
4.3.1 Architecture du filtre FIR 512 à 32 coefficients 105
4.3.2 Chargement des coefficients 106
4.3.3 Résultats et précision de calcul 108

4.3.3.1 Simulation du filtre FIR 512 à 32 coefficients 108
4.3.3.2 Expérimentation du filtre FIR 512 à 32 coefficients sur carte Nallatech 112
4.4 Conclusion 113

Conclusion **115**

Communications et rapports **121**

Table des figures **123**

Liste des tableaux **127**

Bibliographie **129**

Résumé **135**

Introduction

Contexte général de l'étude

Depuis l'apparition des premiers réseaux radiomobiles cellulaires analogiques au début des années 70, le nombre des systèmes de communications numériques a littéralement explosé pour satisfaire le marché des télécommunications. Si hier, les systèmes se cantonnaient essentiellement à la transmission de la parole, le développement des systèmes de communications numériques aidant, les services se sont depuis étendus à la transmission de données et aux applications de type multimédia. Par rapport aux transmissions filaires, les télécommunications sans-fil permettent de simplifier l'infrastructure d'un accès permanent aux réseaux locaux. Cependant, le succès de ces nouveaux systèmes conduit à une grande surcharge du spectre radioélectrique et pose un réel problème de cohabitation. De plus, ces nouveaux services nécessitent des débits de plus en plus importants, tout en garantissant une certaine qualité de service. Il est donc nécessaire d'envisager de nouvelles techniques de transmission.

Le besoin d'améliorer la capacité des réseaux sans-fil a alors amené à s'intéresser aux techniques de communication MIMO (*Multiple Input Multiple Output*) [1, 2, 3] qui répondent à ce besoin d'augmentation du débit et de la qualité des transmissions. Les systèmes MIMO se servent de réseaux d'antennes, à la fois à l'émission et à la réception d'une communication radio et ainsi exploitent la dimension spatiale pour la transmission de l'information. Différentes techniques MIMO existent, nécessitant plus ou moins la connaissance du canal (formation de voies, multiplexage, codage temps-espace) pour privilégier la robustesse et le débit. Comme dans toutes les études des systèmes, une analyse approfondie du canal de transmission et des mécanismes de propagation associés s'avère indispensable. Si, dans un contexte classique, la caractérisation et la modélisation du canal peuvent se limiter aux domaines temporel et fréquentiel, il est nécessaire pour les systèmes MIMO de considérer la dimension spatiale également.

Ces techniques MIMO révolutionnent aujourd'hui le monde des radiocommunications et constituent une forte activité au sein des différentes équipes de recherche nationales et internationales.

Par ailleurs, les performances expérimentales d'un système de communication peuvent être évaluées en utilisant un simulateur matériel du canal de propagation. La simulation logicielle des systèmes multi-antennes et à grande largeur de bande est complexe, surtout d'un point de vue temps de calcul. La simulation matérielle du canal radio fournit la vitesse de traitement nécessaire. D'autres avantages de la simulation matérielle sont l'évaluation des performances en temps réel et la répétabilité des tests pour n'importe quel système MIMO à l'essai. Un simulateur matériel peut également être utilisé pour comparer les performances de divers systèmes de radiocommunication dans un canal de propagation variant dans le temps dans les mêmes conditions de test souhaitées. De grandes compagnies comme Elektrobit, Spirent Communications ou Seibersdorf Research GmbH, qui est une entreprise d'ARC (*Austrian Research Centers*) ont développé des simulateurs matériels de canaux MIMO disponibles dans le commerce. Mais ces simulateurs sont souvent trop chers et donc prohibitifs pour un laboratoire de communications. Pourtant, pour satisfaire la demande des équipementiers, il est nécessaire d'avoir des moyens de simulation afin de reproduire des conditions de test réalistes.

C'est dans ce contexte que s'inscrit le projet SIMPAA 2 (SImulateur Matériel de Propagation pour Antennes Adaptatives 2) qui sert de cadre à notre étude. Il s'agit d'un PRIR (Projet de Recherche d'Intérêt Régional) supporté par la Région Bretagne.

Objectifs du projet SIMPAA 2

Le projet SIMPAA 2 constitue la suite du projet de recherche RNRT (Réseau National de Recherche en Télécommmunications) appelé SIMPAA [4]. Le projet SIMPAA avait pour but l'étude et la réalisation d'un simulateur matériel de canaux à réseaux multiantennes bidirectionnels adapté aux futurs systèmes UMTS (*Universal Mobile Telecommunications System*). Ce type de matériel permet de reproduire en laboratoire le comportement du canal de propagation radio-mobile. Pour les concepteurs de systèmes de communications, cet appareil se révèle très utile pour tester et valider les différents prototypes réalisés. L'objectif du projet SIMPAA 2 est de lever certains verrous identifiés dans le cadre du précédent projet RNRT, en particulier, d'une part, le développement de modèles de canaux, qui fait l'objet d'un second travail de thèse mené par H. Farhat, et, d'autre part, l'étude de la faisabilité de toute la partie numérique du simulateur en utilisant une plate-forme de prototypage à base de circuits programmables. De plus, en exploitant les performances technologiques des derniers circuits sur le marché, il est possible de pousser les limites des caractéristiques du simulateur et d'étendre son utilisation à de nouvelles applications telles que les réseaux locaux sans-fil à haut débit (WLAN - *Wireless Local Area Network*) ou les systèmes de la 4ième génération (4G). Ce projet SIMPAA 2 constitue, par ailleurs,

un complément essentiel au projet CPER (Contrat Plan Etat-Région) PALMYRE (PlAteforme de déveLoppeMent et d'évaluation de sYstèmes RadioElectriques) [5].

Les travaux présentés dans ce document et qui ont été menés dans le cadre du projet SIMPAA 2 concernent donc l'étude de la faisabilité et la réalisation de l'architecture de la partie numérique du simulateur matériel de canaux de propagation MIMO bidirectionnels.

Aperçu du document

Les réseaux UMTS et WLAN constituent les systèmes de télécommunications mobiles de la troisième génération capables d'offrir au grand public les services multimédias à haut débit. Ces services ont été améliorés par l'utilisation de la technologie MIMO. Par ailleurs, les ondes électromagnétiques transmises interagissent avec l'environnement de propagation. Il est donc nécessaire de tenir compte des principaux phénomènes de propagation pendant la conception des futurs systèmes de communications. De plus, les performances attendues dépendent étroitement des propriétés du canal de propagation qu'il est nécessaire de caractériser, modéliser et simuler.

Les phénomènes physiques de la propagation radioélectrique et la caractérisation du canal feront l'objet du chapitre 1 de ce document. Dans un second temps, la présentation des modèles de canaux et l'état de l'art des simulateurs existants complèteront ce premier chapitre.

Après avoir énoncé le cahier des charges du simulateur matériel de propagation MIMO, le chapitre 2 visera à expliciter les modules qui composent le simulateur. Notamment, on justifiera le choix des architectures selon les deux domaines proposés, quelques paramètres utiles, ainsi que le choix du matériel qui permettra d'implémenter les architectures selon les approches choisies.

Par la suite, ces deux approches envisagées feront l'objet des chapitres 3 et 4, avec les tests des architectures, leur validation et leurs résultats de mesures.

Enfin, la conclusion générale clôturera ce document en présentant un bilan du travail et des perspectives de travaux à venir.

Chapitre 1

Canal de propagation radioélectrique

Sommaire

1.1 Caractérisation du canal de propagation 6

1.1.1 Phénomènes physiques de la propagation radioélectrique 7

1.1.2 Canal de propagation 15

1.2 Modèles de canaux de propagation 18

1.2.1 Modèles déterministes 19

1.2.2 Modèles statistiques 20

1.2.3 Modèles hybrides 20

1.3 Etat de l'art des simulateurs de canaux MIMO 21

1.4 Conclusion 27

Que ce soit pour une liaison montante (du mobile à la station de base) ou une liaison descendante (de la station de base au mobile), le signal émis traverse un milieu de propagation et est affecté, d'une part, par le type d'environnement, et d'autre part, par la structure physique de l'environnement de propagation. Le signal peut emprunter plusieurs trajets, on parle alors de propagation par trajets multiples qui a pour conséquence l'apparition des échos du signal émis. Les paramètres de ces trajets multiples varient dans le temps puisque l'émetteur, le récepteur ou les objets faisant partie de l'environnement de propagation peuvent se déplacer. Les performances des systèmes de communication à réseaux d'antennes sont donc dépendantes du milieu de propagation et c'est pourquoi nous avons besoin de modèles permettant de reproduire le canal d'une manière réaliste. Il est donc nécessaire de modéliser les canaux de propagation radioélectrique décrivant les différents types d'environnements qui affectent le signal.

Dans un premier temps, l'objectif est de décrire et de caractériser le canal de propagation, en précisant tout d'abord la nature des différentes interactions avec l'environnement subies par le signal émis. Ensuite, la caractérisation du canal de propagation est présentée

à travers les caractéristiques de la réponse impulsionnelle à simuler, de façon à connaître les paramètres utiles à la simulation du canal.

Dans un second temps, ce chapitre se consacre à la simulation du canal de propagation, d'une part en décrivant les différentes classes de modèles disponibles, afin de connaître la meilleure façon de modéliser le canal, d'autre part en dressant un état de l'art des simulateurs matériels de canaux de propagation MIMO existants.

1.1 Caractérisation du canal de propagation

Le développement d'un système de communication radio-mobile exige la connaissance des principales caractéristiques du canal de propagation. Dans le contexte des communications sans-fil dans le réseau cellulaire UMTS, les environnements de propagation peuvent être classés en plusieurs catégories de cellules [6] différenciées par leur taille, leur trafic, le nombre et les dimensions des obstacles susceptibles de perturber la propagation des ondes électromagnétiques. Plus la taille de la cellule est grande, plus les retards des différents trajets sont significatifs et la puissance reçue réduite. Par conséquent, il est plus difficile d'atteindre de bonnes performances avec un débit élevé.

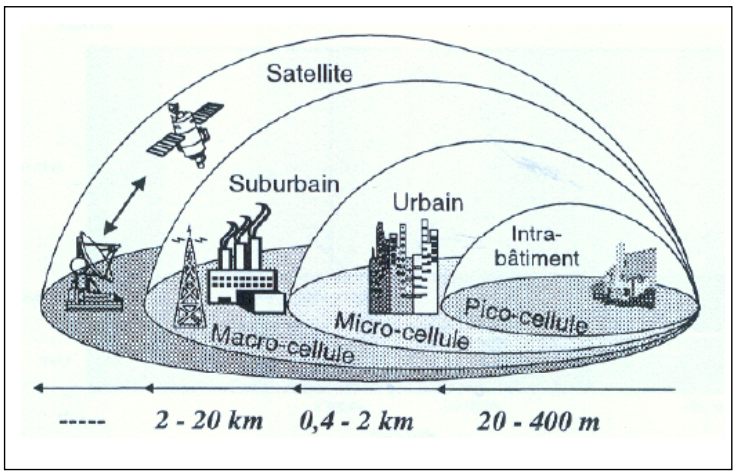

FIG. 1.1 – Cellules radioélectriques

Par exemple, les macro-cellules sont utilisées pour les environnements ruraux plats, montagneux et suburbains. Elles peuvent contenir une station de base située au moins au-dessus du niveau (moyen) des toits et sont caractérisées par des distances allant jusqu'à 20 km en milieu rural. La plupart des études a porté sur les milieux urbain et suburbain, à cause de la relative complexité pour les caractériser. Les micro-cellules sont utilisées pour les environnements urbains (rues, places, parking,...), la station de base étant souvent située au-dessous du niveau des toits. Les pico-cellules sont utilisées à l'intérieur des bâtiments.

Du fait du placement des antennes au-dessous du niveau des toits dans le contexte micro-cellulaire, les caractéristiques du canal de propagation sont très différentes de celles rencontrées dans le cas macro-cellulaire. Si les structures présentes à l'extérieur des bâtiments sont d'un autre ordre de grandeur que celles présentes à l'intérieur des bâtiments, les phénomènes de propagation dans le cas micro-cellulaire peuvent se rapprocher qualitativement de ceux concernant l'intérieur des bâtiments. Citons, par exemple, l'effet de guidage pouvant être présent dans une rue ou dans un couloir.

Voyons maintenant les effets de ces environnements sur la propagation.

1.1.1 Phénomènes physiques de la propagation radioélectrique

Les phénomènes radioélectriques sont de deux ordres : les distorsions de l'onde électromagnétique et la superposition des signaux indésirables, désignés indifféremment comme bruit [7]. Les quatre principaux mécanismes régissant l'interaction de l'onde avec l'environnement sont expliqués dans les paragraphes suivants. D'autre part, les distorsions dues à l'interaction de l'onde avec le milieu sont observées à deux niveaux distincts, selon que leur impact est visible à grande ou à petite échelle.

1.1.1.1 Le bruit radioélectrique

Le bruit regroupe l'ensemble des signaux ne transportant pas d'information utile et venant perturber le signal désiré. Il s'agit d'une perturbation dont les origines sont le canal de propagation (bruit externe) et les dispositifs électroniques utilisés dans le récepteur (bruit interne) [8, 9, 10].

Les bruits externes peuvent être d'origine extra-terrestre ou terrestre. La première catégorie ne rentrant en compte que dans les liaisons spatiales ou dans les voies montantes vers des satellites, seules restent les sources de bruit terrestres. Elles regroupent les bruits dûs aux parasites atmosphériques, ceux provenant des rayonnements divers de l'environnement, les interférences éventuelles entre les utilisateurs du canal de propagation ou encore les bruits d'origine industrielle et dûs à l'activité humaine en général. Quant au bruit interne, il a pour origine le mouvement brownien des électrons présents dans les composants électroniques du récepteur.

1.1.1.2 Les mécanismes de la propagation

Dans le cas idéal de la propagation en espace libre, il n'y a rien en dehors des antennes émettrices et réceptrices, il n'existe donc que le trajet direct. Dans un environnement où des obstacles sont présents, on obtient le trajet direct entre l'émetteur et le récepteur en vérifiant que la zone appelée « premier ellipsoïde de Fresnel », dont l'étendue varie avec la longueur d'onde et la distance entre l'émetteur et le récepteur [6], est dégagée de

tout obstacle. En réalité, à part le trajet direct entre émetteur et récepteur, chaque trajet qui apporte une contribution au niveau du récepteur a rencontré au moins un obstacle. On parle alors de propagation à trajets multiples. Les quatre phénomènes régissant l'interaction de l'onde avec l'environnement sont les phénomènes bien connus de réflexion, réfraction, diffraction et diffusion [6, 11, 12]. Ils sont considérés comme les mécanismes physiques de base de la propagation dans le contexte des communications sans-fil et sont illustrés par la figure 1.2.

Réflexion/Réfraction : La réflexion et la réfraction interviennent lorsque l'onde interagit avec un obstacle dont les dimensions sont très grandes et les irrégularités très petites devant la longueur d'onde. Lorsque l'objet est parfaitement conducteur, toute l'énergie incidente est réfléchie. Dans le cas contraire, une partie de l'énergie pénètre dans l'objet, selon le phénomène de réfraction. La quantité d'énergie transportée par l'onde réfractée dépend de la capacité d'absorption des matériaux. Pour une surface plane, les angles d'incidence, de réflexion et de réfraction sont reliés par la loi de Snell-Descartes et, en particulier, les angles d'incidence et de réflexion sont égaux. Dès lors que la surface présente une certaine rugosité par rapport à la longueur d'onde, l'onde incidente est réfléchie dans plusieurs directions et l'on parle de réflexion diffuse.

Diffraction : La diffraction se produit lorsque le chemin de propagation est obstrué par un obstacle présentant des dimensions faibles devant la longueur d'onde ou possédant des arêtes vives. Selon le principe de Huyghens, chaque point du front d'onde se comporte comme une source secondaire. L'énergie transmise par ces sources permet au signal de se propager dans les zones qui seraient considérées comme zones d'ombre par l'optique géométrique, expliquant ainsi que les ondes radio arrivent au niveau du récepteur en l'absence de visibilité directe ou de l'intervention d'autres types d'interaction.

Diffusion : La diffusion d'une onde apparaît s'il existe sur le trajet de l'onde un paquet très dense d'objets de dimensions du même ordre de grandeur ou inférieures à la longueur d'onde. Dans ce cas, l'onde est redirigée dans toutes les directions avec différentes atténuations. Le même phénomène est observé avec une surface rugueuse présentant des aspérités de dimensions proches de la longueur d'onde. L'effet sur un obstacle pris séparément relève de la diffraction, mais lorsqu'on l'étudie dans une zone à forte densité d'obstacles, on préférera utiliser une approche statistique débouchant sur le mécanisme de diffusion. Nous retrouvons ce type de phénomènes lors de la propagation des ondes en présence d'arbres, par exemple.

Ces mécanismes de propagation expliquent ainsi la possibilité de communication pour des liaisons radiomobiles où l'émetteur et le récepteur ne sont pas nécessairement en visibilité directe.

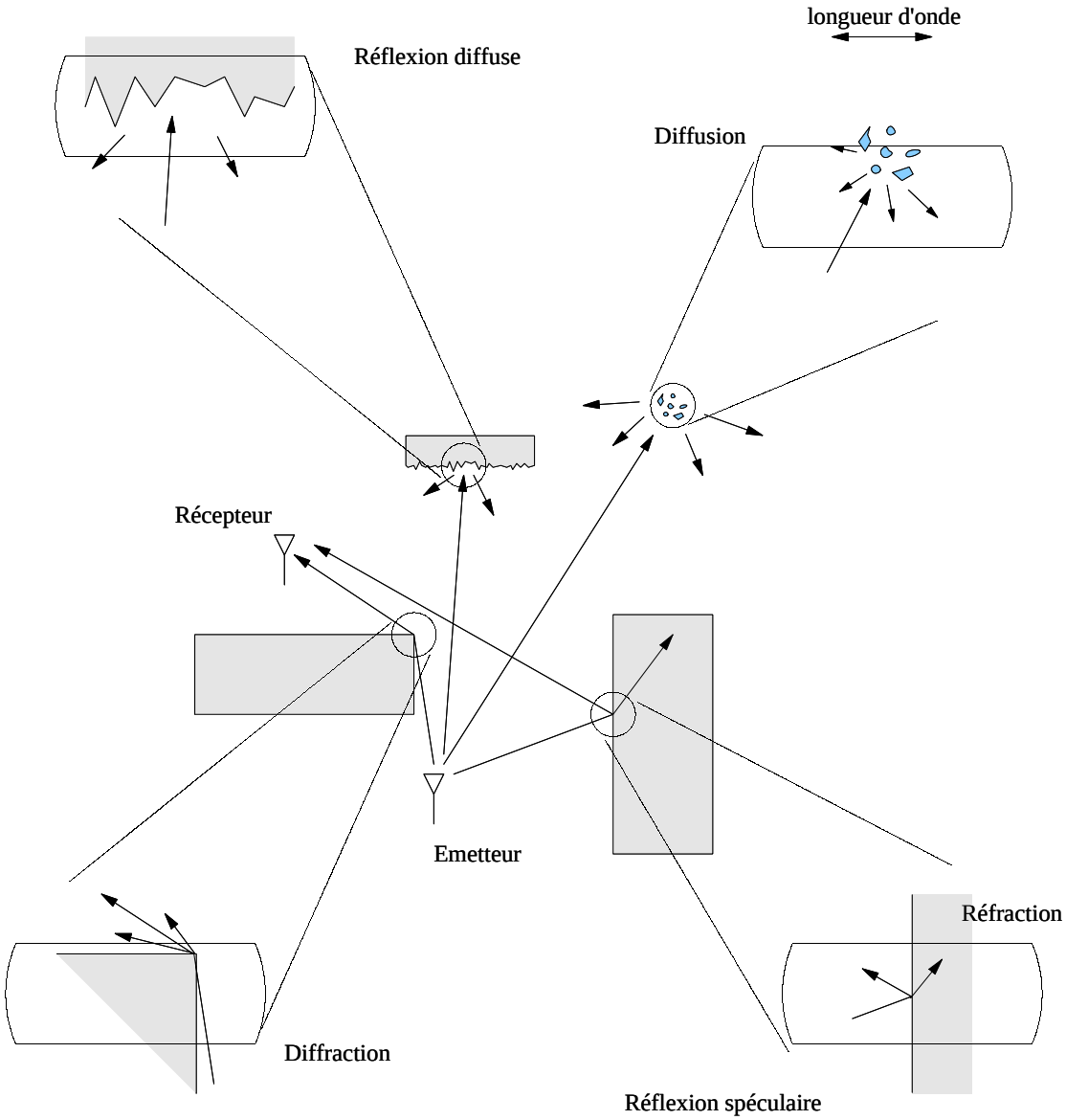

FIG. 1.2 – Propagation radioélectrique - Nature des interactions de l'onde électromagnétique avec le milieu

1.1.1.3 Les distorsions du signal

Les phénomènes à l'origine des variations à grande et moyenne échelle sont respectivement les pertes dues à la distance et les effets de masquage.

Les pertes de puissance en fonction de la distance sont les seules pertes observables dans le cas d'une propagation en espace libre où les antennes d'émission et de réception sont en visibilité directe et où aucun obstacle ne perturbe la propagation. Elles sont observables par la décroissance notée en pointillée sur la figure 1.3.

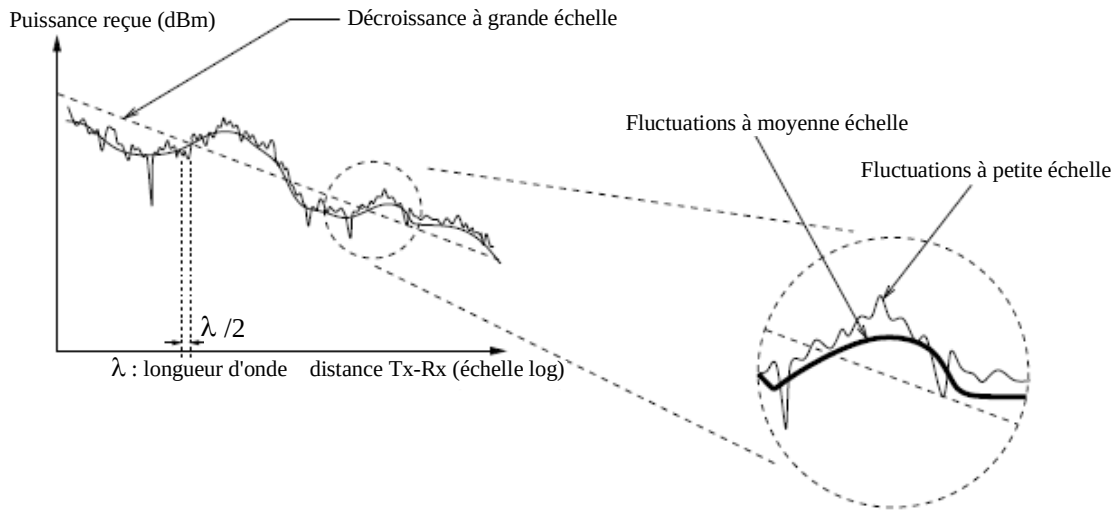

FIG. 1.3 – Fluctuations de la puissance reçue en fonction de la position du récepteur par rapport à l'émetteur

Ces pertes moyennes de puissance en espace libre sont proportionnelles au carré de la distance d séparant les antennes de l'émetteur de celles du récepteur. En revanche, dans un environnement de propagation réel où la présence de nombreux obstacles conduit à une propagation qui n'a plus lieu en espace libre, la puissance P du signal décroît proportionnellement à la distance entre antennes comme le montre la relation $P \propto \frac{1}{d^{\alpha}}$. Le paramètre réel positif α, fonction du type d'environnement, est compris entre les valeurs 2 et 5 associées respectivement à la propagation en espace libre et en milieu obstrué.

Par ailleurs, la présence d'obstacles dans l'ellipsoïde de Fresnel provoque un effet de masquage qui se traduit par une variation lente de la puissance du signal reçu, en plus des pertes de puissance en fonction de la distance. Cette variation lente à moyenne échelle de la puissance est fonction de la nature, des dimensions et de l'emplacement des matériaux traversés par l'onde électromagnétique.

Les fluctuations à grande échelle correspondent aux variations de la puissance moyenne mesurées sur un déplacement de plusieurs longueurs d'onde ou sur un intervalle de temps suffisamment long (cf. figure 1.3).

Les fluctuations à petite échelle sont observées sur un court intervalle de temps et un déplacement spatial suffisamment petit par rapport à la longueur d'onde pour négliger les variations à grande échelle (cf. figure 1.3). A l'origine de ces phénomènes, un certain nombre d'obstacles et l'irrégularité du terrain engendrent, au niveau du récepteur, l'apparition de plusieurs répliques du signal transmis qui interfèrent de manière constructive ou destructive [13].

La sélectivité fréquentielle : Un canal est sélectif en fréquence lorsque sa réponse fréquentielle présente des variations dans la bande de fréquences considérée. Dans le domaine temporel, cela se traduit par une série d'échos plus ou moins retardés.

A partir de la réponse impulsionnelle $h(x,\tau)$, à une position x donnée, le retard τ peut être considéré comme une variable aléatoire ayant une densité de probabilité de la forme [6] :

$$p(x,\tau) = \frac{|h(x,\tau)|^2}{\int_0^{+\infty} |h(x,\tau)|^2 d\tau} \tag{1.1}$$

On définit ensuite la valeur moyenne des retards $\overline{\tau}$ et le retard quadratique moyen $\overline{\tau^2}$ par :

$$\overline{\tau} = \int_0^{+\infty} \tau.p(\tau)d\tau \tag{1.2}$$

$$\overline{\tau^2} = \int_0^{+\infty} \tau^2.p(\tau)d\tau \tag{1.3}$$

La sélectivité fréquentielle est caractérisée par deux paramètres : l'écart type des retards et la bande de cohérence.

Ecart type des retards : L'écart type des retards τ_{rms} (*root mean square delay spread*) traduit l'étalement des retards des trajets formant la réponse impulsionnelle et est défini par :

$$\tau_{rms} = \sqrt{\overline{\tau^2} - \overline{\tau}^2} \tag{1.4}$$

Selon la modulation choisie, un symbole émis peut correspondre à un bit ou à un nombre de bits. Si τ_{rms} est largement inférieur (< 10 %) à la durée d'un symbole transmis alors le canal est non-sélectif (ou plat) en fréquence. En revanche, si τ_{rms} est supérieur à la durée d'un symbole transmis, alors le canal est sélectif en fréquence et il existe des évanouissements fréquentiels du signal. L'interférence entre symboles est alors plus forte, il y a donc plus d'erreurs et la qualité de la liaison est alors moins bonne.

Bande de cohérence : Dans le domaine fréquentiel, la bande de cohérence B_c d'un canal donne une mesure de la similarité de la réponse fréquentielle $H(f)$ du canal de

propagation, c'est-à-dire qu'elle indique le degré de corrélation entre deux fréquences distinctes de la réponse fréquentielle. La bande de cohérence est définie pour à un seuil qui représente un certain pourcentage de la valeur maximale du module de la fonction d'autocorrélation $R_H(\Delta f)$ de la réponse fréquentielle du canal [14] définie par :

$$R_H(\Delta f) = \int_{-\infty}^{+\infty} H(f).H^*(f + \Delta f)df \tag{1.5}$$

Sur le tracé du module de cette fonction, à la figure 1.4, on place un seuil $S = \eta.|R_H(0)|$ pour une valeur η comprise entre 0 et 1, $|R_H(0)|$ étant la valeur maximale du module de la fonction d'autocorrélation. On obtient ainsi la valeur de la bande de cohérence sur l'axe Δf. Par exemple, la bande de cohérence pour un seuil à environ 70% correspond à la bande de cohérence à -3 dB de la valeur maximale $|R_H(0)|$ du module de la fonction d'autocorrélation [14, 15].

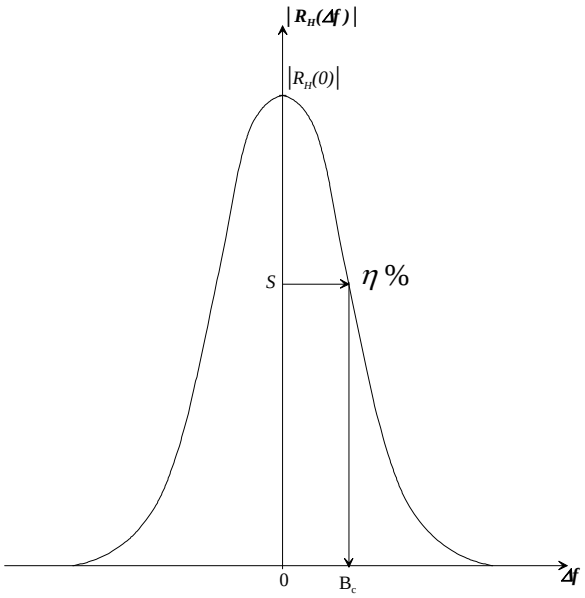

FIG. 1.4 – Module de la fonction d'autocorrélation de la réponse fréquentielle du canal

D'autre part, si l'écart entre deux fréquences est inférieur à la bande de cohérence, les évanouissements subis par l'une des deux conditionnent ceux subis par l'autre. Si maintenant l'écart est supérieur à la bande de cohérence, les deux fréquences subissent des variations faiblement corrélées, c'est-à-dire ayant un coefficient de corrélation inférieur au pourcentage choisi.

L'effet Doppler : Dans le cas d'un système radio-mobile, le mouvement de la source, du récepteur ou de l'environnement entraîne ce que l'on appelle l'effet Doppler, qui se traduit

par un décalage entre la fréquence de l'onde émise et celle de l'onde reçue lorsque l'émetteur et le récepteur sont en mouvement l'un par rapport à l'autre. Les phénomènes qui provoquent le décalage Doppler sont très répandus dans le contexte radio-mobile puisque l'émetteur, le récepteur ou les obstacles alentours sont susceptibles d'être en mouvement, ce qui implique l'évolution du canal de propagation dans le temps. Ces variations temporelles, induisant des fréquences de décalage Doppler f_d dans le domaine fréquentiel, sont caractérisées également en fonction de deux paramètres : la bande Doppler et le temps de cohérence.

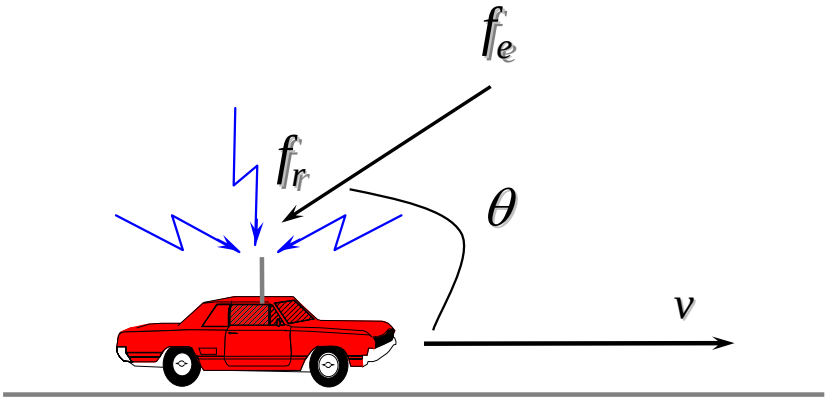

FIG. 1.5 – Illustration du phénomène Doppler

Bande Doppler : Lorsqu'on considère une propagation à trajet unique avec f_e la fréquence émise, f_r la fréquence reçue, v est la vitesse du récepteur que l'on considère mobile et θ l'angle d'arrivée de l'onde émise, comme illustré à la figure 1.5, l'expression de la fréquence Doppler f_d est :

$$f_d = f_e.\frac{v}{c}.\cos\theta \quad \text{et on a :} \quad f_r = f_e + f_d \tag{1.6}$$

La fréquence Doppler maximale est obtenue pour $\theta = 0$, le mobile récepteur se déplaçant vers l'émetteur.

Cependant, nous sommes dans un cas de propagation à trajets multiples. Ainsi, pour une fréquence émise en continue, on reçoit plusieurs fréquences de décalage Doppler selon les différents angles d'arrivée θ_n des n trajets considérés. Chaque fréquence Doppler f_{d_n}, associée au n^{ième} trajet est calculée par :

$$f_{d_n} = f_e.\frac{v}{c}.\cos\theta_n \tag{1.7}$$

De plus, lorsqu'on émet une bande de fréquences, plusieurs fréquences de décalage Doppler sont reçues pour chacune des fréquences émises. On reçoit donc une bande

de fréquences Doppler, dont la largeur B_d est égale à deux fois la fréquence Doppler maximale.

Comme illustré sur la figure 1.5, si la source et le récepteur se rapprochent l'un de l'autre, la bande de fréquences du signal reçu est supérieure à celle du signal émis. Si, par contre, ils s'éloignent l'un de l'autre, la bande de fréquences du signal reçu est inférieure à celle du signal émis. En déplacement perpendiculaire à la ligne émetteur-récepteur, il n'y a pas de décalage de fréquence car la composante radiale du vecteur vitesse est nulle.

Temps de cohérence : Le temps de cohérence T_c est l'équivalent dans le domaine temporel de la bande Doppler. C'est l'écart de temps au-delà duquel les évanouissements temporels sont suffisamment décorrélés. Le temps de cohérence est inversement lié à la bande Doppler. Il peut être défini pour un seuil qui représente un certain pourcentage de la valeur maximale du module de la fonction d'autocorrélation $R_h(\tau, \Delta t)$ de la réponse impulsionnelle du canal définie par :

$$R_h(\tau, \Delta t) = \int_{-\infty}^{+\infty} h(t, \tau).h^*(t + \Delta t, \tau)dt \tag{1.8}$$

De la même façon que pour la bande de cohérence, sur le tracé du module de la fonction d'autocorrélation à la figure 1.6, on place un seuil $S = \eta.|R_h(\tau, 0)|$ pour une valeur η comprise entre 0 et 1, $|R_h(\tau, 0)|$ étant la valeur maximale du module de la fonction d'autocorrélation. On obtient alors la valeur du temps de cohérence sur l'axe Δt, pour un retard τ donné.

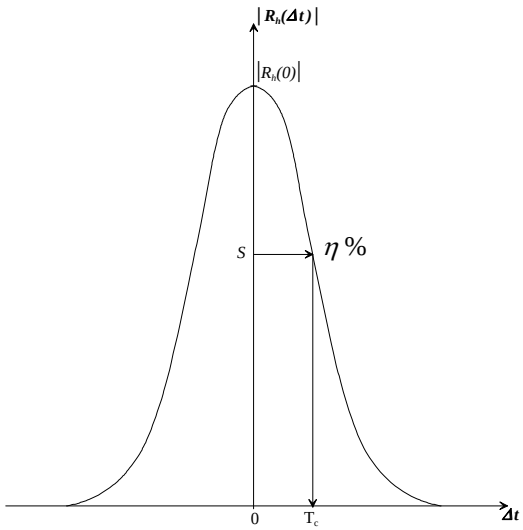

FIG. 1.6 – Module de la fonction d'autocorrélation de la réponse impulsionnelle du canal

On obtient le temps de cohérence $T_c(\tau)$, avec $0 < \tau < W_\tau$, W_τ étant la largeur de la réponse impulsionnelle. Ensuite, on détermine le temps de cohérence minimal qui est le minimmum des temps de cohérence pour un retard τ donné. Ce temps de cohérence minimal définit la fréquence de rafraîchissement des profils de réponse impulsionnelle.

1.1.2 Canal de propagation

Dans le cas général, le canal de propagation est variant dans le temps. De ce fait, la réponse impulsionnelle qui le caractérise est également variante dans le temps et permet d'évaluer l'influence des trajets multiples du canal sur un système de transmission. La réponse impulsionnelle est une fonction à la fois de la variable t, qui représente la variation du temps par rapport à une valeur de référence choisie, et de la variable τ qui représente le retard des trajets multiples du canal pour une valeur fixée de t.

Un canal de propagation MIMO se compose de plusieurs canaux SISO (*Single Input Single Output*) corrélés et variant dans le temps, comme le montre la figure 1.7 pour le cas particulier d'un canal MIMO 2×2. On considère que les variables temps t et retard τ sont *indépendantes*.

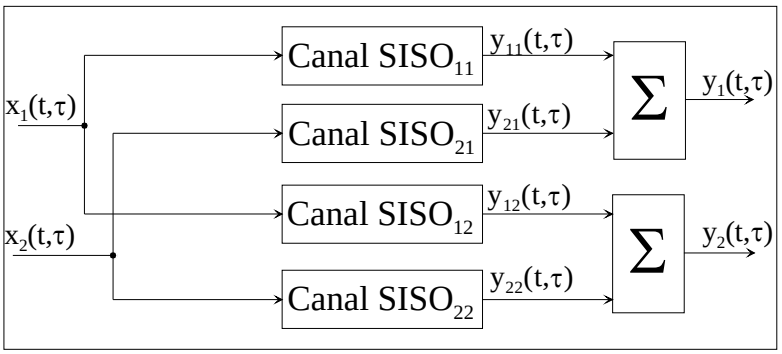

FIG. 1.7 – Représentation d'un canal MIMO dans une configuration 2×2.

D'autre part, chaque canal SISO variant aléatoirement dans le temps est décrit par les fonctions caractéristiques de Bello [16], comme le montre la figure 1.8.

Ces fonctions caractéristiques sont :

- $h(t, \tau)$ la réponse impulsionnelle du canal variant dans le temps ou fonction d'étalement des retards ;
- $H(t, f)$ la réponse fréquentielle du canal variant dans le temps ;
- $F(\nu, f)$ la fonction d'étalement Doppler où ν est la variable du décalage Doppler ;
- $S(\nu, \tau)$ la fonction d'étalement retard-Doppler.

La connaissance d'une de ces fonctions caractéristiques permet de calculer les autres fonctions en utilisant la transformée de Fourier directe ou inverse, comme illustré sur la figure 1.8. Ainsi, chaque canal SISO contient les quatre fonctions caractéristiques du canal.

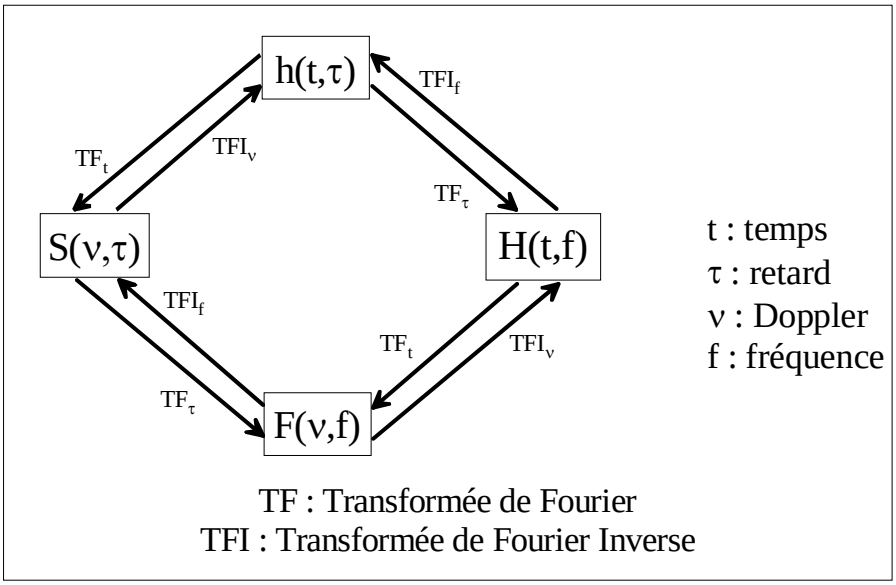

FIG. 1.8 – Les fonctions caractéristiques de Bello.

Comme chaque fonction caractéristique dépend de deux variables indépendantes, leur représentation graphique devrait être vue comme des surfaces. Cependant, un sondeur temporel de canal, comme celui conçu et réalisé à l'IETR [17], permet d'obtenir des profils de réponses impulsionnelles qui sont des « tranches » de ces surfaces comme illustrées à la figure 1.9.

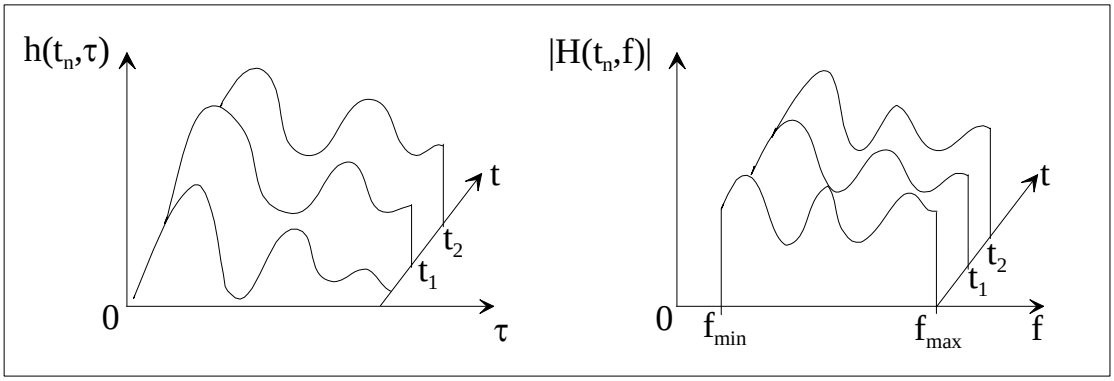

FIG. 1.9 – Profils de réponses impulsionnelles et fréquentielles.

D'autre part, ces profils seront amenés à être renouvelés avec une fréquence de rafraîchissement qui dépend du temps de cohérence du canal, déduit de l'autocorrélation de la réponse impulsionnelle, comme nous l'avons vu précédemment.

En pratique, on connaît des profils de réponses impulsionnelles $h(t_n, \tau)$, dont le rafraîchissement représente le passage de $h(t_n, \tau)$ à $h(t_{n+1}, \tau)$ ou de $H(t_n, f)$ à $H(t_{n+1}, f)$,

comme illustré à la figure 1.9. Si la différence entre t_{n+1} et t_n est égale à une constante T_{raf}, alors $\frac{1}{T_{raf}}$ est la fréquence de rafraîchissement f_{raf}.

En résumé, le canal variant dans le temps peut être décrit soit, dans le domaine temporel, par sa réponse impulsionnelle $h(t,\tau)$ soit, dans le domaine fréquentiel, par sa réponse fréquentielle $H(t,f)$.

Prenons l'exemple d'un canal MIMO variant dans le temps où les signaux émis sont des signaux radiofréquences (RF). On considère $N_T = 2$ antennes d'émission et $N_R = 2$ antennes de réception, comme illustré à la figure 1.10.

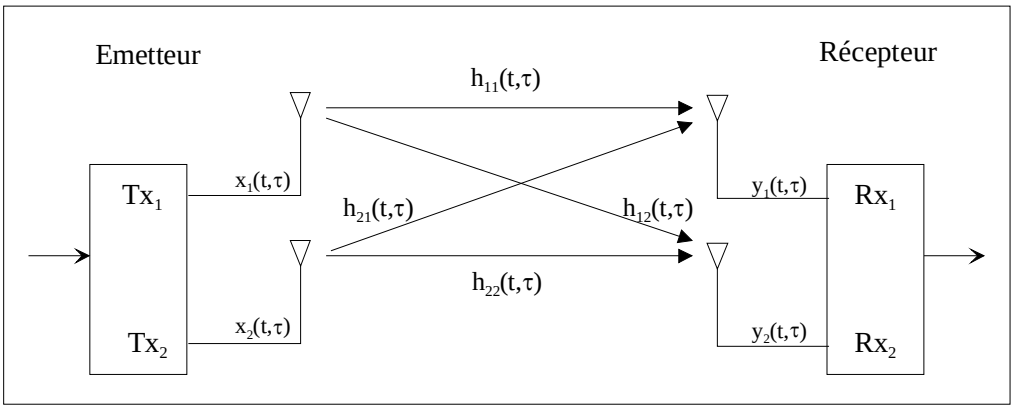

FIG. 1.10 – Canal MIMO (2 x 2 canaux SISO unilatéraux).

Si i et k représentent respectivement les indices des antennes d'émission et des antennes de réception, j étant l'unité imaginaire réservée aux calculs avec des nombres complexes et si on néglige le bruit du récepteur, le signal $y_k(t,\tau)$ reçu par la $k^{ième}$ antenne de réception peut être exprimé par :

$$y_k(t,\tau) = x_1(t,\tau) * h_{1k}(t,\tau) + x_2(t,\tau) * h_{2k}(t,\tau), \quad k = 1,2 \tag{1.9}$$

Cependant, pour une implémentation matérielle, il est plus facile d'utiliser la transformée de Fourier pour remplacer la convolution par rapport à un retard τ par un produit algébrique. Ainsi on a :

$$Y_k(t,f) = X_1(t,f)H_{1k}(t,f) + X_2(t,f)H_{2k}(t,f), \quad k = 1,2 \tag{1.10}$$

Après avoir identifié les différents phénomènes physiques mis en jeu lors de la propagation des ondes, cette partie a présenté succintement les préliminaires théoriques à toute étude du canal de propagation. L'étude théorique poussée des canaux de propagation n'étant pas l'objet central de cette thèse, nous ne présentons que le cas général de la caractérisation du canal de propagation. Les travaux de J. Guillet [18] présentent une description spatio-temporelle en prenant en compte les phénomènes physiques, les

ondes planes transverses et les propriétés de polarisation et de rayonnement des antennes d'émission et de réception pour la caractérisation du canal de propagation.

Ainsi, l'ensemble des notions de base pour la compréhension des mécanismes de la propagation a été présenté.

Par ailleurs, la conception d'un système de communication performant dépend en grande partie de la précision de la représentation du canal réel. C'est pourquoi, on souhaite développer un simulateur matériel devant reproduire des réponses impulsionnelles réalistes du canal de propagation. Pour permettre des tests les plus proches de la réalité, il est dans l'intérêt des ingénieurs et techniciens de systèmes de développer des procédures de simulation basées sur des modèles de canaux. Ces modèles alimenteront ensuite le simulateur matériel afin de reproduire une large gamme de réponses impulsionnelles ou fréquentielles de canal et donc simuler des situations diverses. Un état de l'art des modèles de canaux spatio-temporels sort évidemment du cadre de ce travail et peut être trouvé dans [19, 20, 13, 21, 22]. Ainsi, après la présentation succinte de différents modèles de canaux, un comparatif de divers simulateurs existants clôturera cette partie.

1.2 Modèles de canaux de propagation

Selon les différents types d'environnements indoor et outdoor, il est utile de modéliser le canal par différentes techniques. Plusieurs méthodes de classification des modèles de canaux sont proposées dans la littérature [18, 23, 24, 25]. Une classification possible est celle de la figure 1.11 où on distingue les modèles déterministes, les modèles statistiques (appelés aussi stochastiques) et les modèles hybrides créés par programmation et qui sont un mélange de modèle déterministe et de modèle statistique.

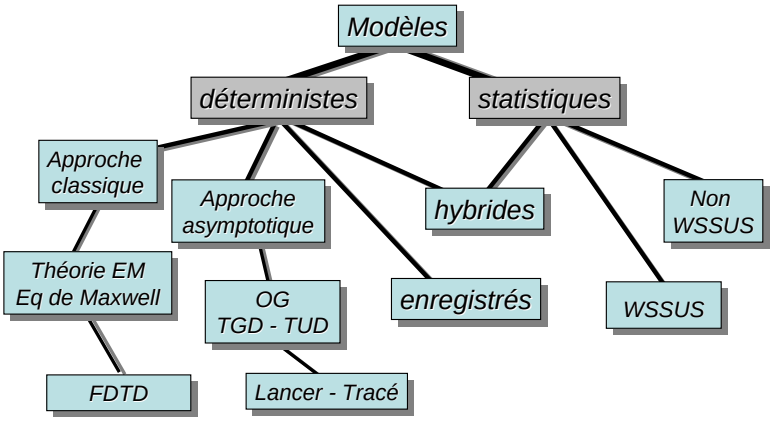

FIG. 1.11 – Modélisation théorique du canal de propagation.

1.2.1 Modèles déterministes

Les modèles déterministes sont basés sur une description précise de l'environnement de propagation, afin de prendre en compte l'influence des obstacles sur la propagation des ondes. Ces modèles ne sont valables que pour des sites où le canal est invariant dans le temps, sauf si le mouvement perturbateur est parfaitement connu, comme, par exemple, dans le cas d'un robot dans un site industriel qui effectue des mouvements parfaitement connus dans un environnement statique. Deux méthodes peuvent être proposées pour réaliser cette description : l'approche classique et l'approche asymptotique. On abordera aussi le cas des modèles dont les résultats sont enregistrés pour pouvoir être rejoués.

L'approche classique est développée par la théorie de l'électromagnétisme régie par les équations de Maxwell. Une des méthodes utilisées pour la résolution de ces équations est la méthode des différences finies appliquée dans le domaine temporel (FDTD - *Finite Difference Time Domain*). Par contre, cette méthode convient uniquement aux environnements de petites dimensions car la discrétisation de l'environnement peut devenir rapidement problématique pour de grandes distances. La comparaison entre la méthode FDTD et le lancer de rayons [26] ainsi que la combinaison de ces méthodes [27] sont développées pour permettre d'optimiser la méthode FDTD par l'approche hybride des deux méthodes. La solution exacte des équations de Maxwell peut être approchée par des méthodes asymptotiques.

Les approches asymptotiques retenues pour conduire à la représentation des champs par des rayons sont l'Optique Géométrique (OG) et la Théorie Géométrique de la Diffraction (TGD) ou la Théorie Uniforme de la Diffraction (TUD). L'OG permet de représenter les rayons directs et réfléchis. Son inconvénient est de prévoir des champs nuls dans les zones d'ombre géométrique où le trajet direct ne peut pas pénétrer. On peut pallier ce défaut à l'aide de la TUD en ajoutant aux rayons de l'OG des rayons diffractés, surtout par des arêtes, pénétrant dans les zones d'ombre. La technique du tracé de rayons est basée sur l'optique géométrique et permet d'estimer les caractéristiques du canal avec une bonne précision, si l'environnement modélisé n'est pas trop complexe, à partir de sa description géométrique et des propriétés électromagnétiques. Cette méthode permet d'estimer également leurs angles de départ (DoD - *Direction of Departure*) et d'arrivée (DoA - *Direction of Arrival*). Mais, le tracé de rayons est aussi l'approche la plus gourmande en ressources de calcul [28]. La technique du lancer de rayons consiste à tracer des rayons directs, réfléchis, transmis ou bien diffusés, de la position du réseau d'antennes d'émission vers la position du réseau d'antennes de réception, dans un espace à trois dimensions et en incrémentant, pas à pas et pour chaque rayon, l'angle d'émission à la fois en élévation et en azimut. Le lancer de rayons prévu pour la propagation en optique (OG, TUD) est assez précis pour les hautes fréquences.

Le modèle à rayons GRIMM [29, 30], développé par France Télécom R&D depuis 1992, est particulièrement intéressant car il conjugue efficacement les méthodes de lancer et de tracé de rayons et permet le calcul des rayons en 3D en un faible temps.

D'autre part, les études de propagation s'appuient souvent sur des campagnes de mesures effectuées par sondage du canal. Ainsi, des ensembles de paramètres caractérisant l'environnement étudié sont extraits à partir des réponses impulsionnelles ou fréquentielles mesurées. Ces profils enregistrés peuvent être rejoués par ordinateur. Il est également possible de rejouer les profils de réponses impulsionnelles issus des techniques du lancer ou du tracé de rayons. L'avantage de pouvoir rejouer des profils de réponses impulsionnelles est de pouvoir comparer des systèmes différents dans les mêmes conditions de test.

1.2.2 Modèles statistiques

Les modèles statistiques ou stochastiques sont obtenus à partir des mesures de réponses du canal. Les modèles statistiques visent à décrire l'évolution des paramètres du canal par des lois statistiques pour obtenir la représentation la plus réaliste possible du canal. L'approche déterministe de la propagation n'est pas toujours simple, notamment à l'intérieur des bâtiments où le nombre d'objets, devenant des réflecteurs ou des diffuseurs potentiels, est trop important pour les incorporer dans une technique de modélisation purement déterministe. De plus, les mouvements des personnes rendent la tâche encore plus difficile. On peut alors extraire par calcul quelques paramètres caractéristiques pour, ensuite, déterminer les lois statistiques qui caractérisent au mieux la variation de ces paramètres. Citons, par exemple, les lois de Rice ou de Rayleigh qui régissent les variations des amplitudes des trajets ou la loi de Poisson qui décrit la variation des retards.

On peut trouver deux catégories pour les modèles statistiques selon que l'on considère ou pas l'hypothèse du canal « stationnaire au sens large et à diffuseurs non corrélés ». L'hypothèse de stationnarité au sens large (WSS - *Wide Sens Stationarity*) signifie que les moments statistiques d'ordre 1 et 2 ne changent pas pendant un intervalle de temps. La corrélation entre deux profils de réponse impulsionnelle ne dépend plus que de l'écart de temps entre deux observations du signal reçu. Les canaux à diffuseurs non corrélés (US - *Uncorrelated Scattering*) supposent que les perturbations dues à des diffuseurs élémentaires, produisant des retards différents, sont décorrélés. Avec l'hypothèse WSSUS, les fonctions à étudier dépendent alors de moins de paramètres pour caractériser le canal [16], [31], [32].

1.2.3 Modèles hybrides

Un modèle hybride déterministe-statistique peut être soit déterministe avec un peu de modèle aléatoire, par exemple dans un site connu où on ajoute le mouvement aléatoire d'un obstacle, soit aléatoire avec un peu de déterminisme. Par exemple, la modélisation de

la réponse impulsionnelle proposée par Saleh et Valenzuela dans [33] repose sur une description à base d'un ensemble de rayons ayant des caractéristiques voisines. Ces ensembles de rayons sont caractérisés par une décroissance exponentielle de la puissance des trajets et des groupes de trajets, appelés « clusters » en anglais, tandis que les temps d'arrivés des rayons sont régis par un processus aléatoire de Poisson.

Il existe aussi des modèles normalisés qui sont des modèles définis par les organismes de normalisation européens comme ETSI (*European Telecommunications Standards Institute*) [34], notamment avec les modèles BRAN [35], et internationaux comme ITU-R (*International Telecommunication Union - Radiocommunication sector*) [36], jouant un rôle essentiel dans la gestion du spectre des radiofréquences. Par ailleurs, des modèles ont été définis par des organismes de partenariat entre l'Europe (ETSI) et l'Asie (Corée du Sud, Chine, Japon) dans le but de standardiser les systèmes de radiocommunication de troisième génération, notamment le 3GPP (3rd *Generation Partnership Project*) [37] et le 3GPP2 (3rd *Generation Partnership Project 2*) [38]. Ces deux organismes de normalisation ont unifié leurs efforts dans le domaine de la caractérisation et de la modélisation du canal de propagation et ont défini un modèle plus complet, le modèle SCM [39] qui fournit de façon statistique les paramètres nécessaires à la description des canaux MIMO, comme la puissance, l'amplitude, la phase, les retards, les DoD et DoA de chaque trajet, notamment pour les simulations indoor et outdoor.

De plus, les actions COST (*Cooperation in Science and Technology*) sont des actions de recherche non compétitive menées par des pays européens. Par exemple, le COST 273 offre un nouveau modèle de canal MIMO statistique [40].

Les modèles COST, 3GPP, 3GPP2 sont des modèles qui sont simples à mettre en œuvre et qui peuvent être pris comme référence.

1.3 Etat de l'art des simulateurs de canaux MIMO

De nombreux simulateurs de canaux de propagation existent pour aider les équipementiers dans la conception et le test de leurs systèmes. Certains simulateurs, ou plates-formes de simulation, intègrent à la fois le simulateur proprement dit et des modules complémentaires intégrant différentes fonctions comme par exemple la génération de signaux, la modulation, la création de modèles de canaux avec la notion d'évanouissements, ainsi que des fonctions de codage et autres techniques numériques.

Les simulateurs logiciels sont, le plus souvent, utilisés dans les laboratoires pour estimer les performances d'un système de radiocommunications sans avoir la contrainte de devoir effectuer des mesures réalistes coûteuses et difficiles à mettre en œuvre. Parmi les nombreux simulateurs logiciels existants [41, 42, 43], nous pouvons brièvement citer le simulateur logiciel Mascaraa, développé par France Télécom [44].

Les simulateurs matériels, par opposition aux simulateurs logiciels, sont capables de simuler le canal de propagation en étant branché physiquement au système que l'on souhaite soumettre à des tests. Un branchement physique permet aux équipementiers de tester le système dans des conditions réelles d'émission et de réception. Un autre intérêt est aussi de pouvoir réaliser les tests de différents systèmes en laboratoire, dans les conditions les plus réalistes possibles.

Les simulateurs matériels du marché sont développés par quelques sociétés, dont les plus connues sont Seibersdorf Research d'ARC [45], Spirent Communications [46], Azimuth [47] et bien sûr Elektrobit [48]. Récemment, un autre constructeur LitePoint [49] a mis sur le marché un simulateur multi-canal. Comme il serait difficile d'être exhaustif sur ce sujet, nous nous limiterons donc à la description et à la comparaison des cinq systèmes dont les marques viennent d'être citées.

ARC Seibersdorf Research : Le simulateur de canal temps réel Smartsim d'ARC pour les systèmes MIMO permet de tester ces systèmes en laboratoire dans de nombreuses situations et pour des environnements différents. Ce simulateur permet de reproduire le comportement des canaux RF. En raison de sa conception modulaire, le simulateur peut être adapté à tous les besoins des clients.

Fig. 1.12 – Simulateur matériel SmartSim d'ARC

ARC SmartSim est un simulateur matériel de canal radio qui effectue des simulations temps réel, dans un environnement de canaux RF, avec des évanouissements programmables. Ce simulateur multicanal est conçu pour simuler jusqu'à huit canaux SISO indépendants pour des systèmes MIMO dans une seule unité de simulation comme illustré à la figure 1.12. Si, pour d'autres applications, plus de huit canaux sont nécessaires, jusqu'à huit unités peuvent être reliées et synchronisées, développant ainsi un système MIMO jusqu'à 64 canaux. En revanche, la documentation constructeur [50] ne précise pas la possibilité d'une simulation bidirectionnelle des systèmes MIMO. Ce simulateur travaille pour une gamme des fréquences centrales pouvant aller de 450 MHz à 6 GHz et pour une largeur

de bande de 40 MHz. Il utilise le modèle COST 259 [51] et des séquences de test pour les normes GSM, UMTS, WLAN, WiMAX. Sinon, l'utilisateur peut y définir ses propres modèles.

Une IHM (Interface Homme Machine) intégrée permet de choisir différents paramètres, par exemple augmenter la puissance de l'émetteur ou réduire la distance de fonctionnement indiquée. Ces arrangements sont donc importants pour avoir de meilleures performances.

Les performances de ce simulateur sont basées sur un modèle de canal de propagation stochastique géométrique, lui-même basé sur les spécifications du COST 259 avec l'extension MIMO. Ce modèle permet une bonne précision des calculs pour les simulations d'évanouissements.

En plus de l'utilisation du modèle de canal stochastique géométrique intégré, l'utilisateur peut tirer profit de n'importe quel modèle de canal représenté par ses réponses impulsionnelles. Les réponses impulsionnelles du canal peuvent être soit générées par un environnement de simulation arbitraire, soit basées sur des mesures de la propagation dans l'environnement considéré.

Spirent Communications : Le simulateur de canal SR5500 de Spirent Communications, présenté à la figure 1.13, simule les caractéristiques du canal radio large bande telles que des trajets multiples variant dans le temps, l'évanouissement rapide, l'évanouissement lent et les pertes de puissance de canal.

FIG. 1.13 – Simulateur matériel SR5500 de Spirent Communications

Parmi ses caractéristiques principales, ce simulateur permet de simuler 24 trajets indépendants, la gamme des fréquences centrales aller de 400 à 2700 MHz pour une largeur de bande de 26 MHz afin de couvrir les applications WiMAX et 802.11a. Les séquences de test comprennent, en plus, les normes EDGE, GSM, GPRS. Il est également possible d'utiliser des modèles d'évanouissement tels que les modèles de Rice et de Rayleigh, entre autres. L'interface graphique permet l'affichage en temps réel des paramètres du canal.Le

SR5500 permet une extension évolutive de 2 à 8 canaux RF. Jusqu'à quatre appareils peuvent être synchronisés, tout en restant contrôlables indépendamment de sorte qu'ils puissent être configurés pour des conditions de test spécifiques, notamment en bidirectionnel. Cependant, les configurations possibles ne sont que des canaux SISO que l'on peut éventuellement multiplier pour simuler des canaux MIMO, au risque d'être moins précis pour les simulations de vrais canaux MIMO [52].

Azimuth : Le simulateur de canal Azimut Channel Emulator *ACE* 400NB est une plate-forme complète et autonome utilisée pour tester les performances des systèmes WiFi 802.11a/b/g et des appareils 802.11n. L'ACE 400NB, présenté à la figure 1.14, simule les caractéristiques des trajets multiples, permettant le test en temps réel des performances des dispositifs WiFi.

Fig. 1.14 – Simulateur matériel ACE 400 NB d'Azimuth

Ce simulateur combine les techniques de modélisation du canal, le contrôle RF avec une interface intuitive pour permettre l'automatisation des tests. Il réalise une simulation bidirectionnelle du canal de propagation MIMO 4×4 jusqu'à 32 canaux SISO unidirectionnels et comporte 6 modèles de canaux spécifiques de la norme 802.11n intégrés qui permettent à des utilisateurs de recréer les scénarios WiFi typiques pour le test de performances. Cette simulation temps réel dans un environnement de laboratoire remplace les essais pratiques traditionnels trop longs. C'est la première solution de test WiFi conçue pour simuler les environnements à trajets multiples.

Ce simulateur offre des capacités de modélisation de canal sophistiquées dans un système facile à utiliser, améliorant les tests temps réel des systèmes sans-fil. C'est un simulateur de canal entièrement bidirectionnel qui fournit de véritables modèles de canaux et supporte une largeur de bande de canal de 40 MHz [53].

Elektrobit : Elektrobit compte une dizaine d'années d'histoire en tant que concepteur et fabricant de simulateurs matériels de canaux radio. La série Propsim C offre deux

solutions : le Propsim C2 qui est un dispositif à deux canaux, ayant 35 MHz de largeur de bande, à plus faible coût, et le Propsim C8, auquel on s'intéresse, qui peut simuler jusqu'à 8 canaux, chacun ayant 70 MHz de largeur de bande. La gamme des fréquences centrales peut aller de 360 MHz à 6 GHz pour des tests incluant les normes GSM, EDGE, UMTS, WLAN et WiMAX.

FIG. 1.15 – Simulateur matériel Propsim C8 d'Elektrobit

La version de base de Propsim C8 permet toutes les configurations MIMO $N \times M$, où le produit $N \times M$ est inférieur ou égal à 8. La version plus élaborée disponible du C8 comprend 16 canaux permettant des simulations MIMO 4×4 avec un seul appareil. La documentation technique ne précise pas s'il s'agit de canaux bidirectionnels ou unidirectionnels [54].

Présenté à la figure 1.15, le simulateur multicanal Propsim C8 a été conçu pour répondre aux exigences de tests des systèmes sans-fil nécessitant des canaux MIMO et une grande largeur de bande. Le logiciel de simulation de canal, inclus dans Propsim C8, est conçu pour créer des conditions d'évanouissements multicanaux réalistes. Les paramètres réglables sont, par exemple, les retards, l'amplitude, la phase, les évanouissements, le Doppler, la simulation spatiale des canaux de propagation basée sur la dispersion des DoA... Ces paramètres peuvent être fixés indépendamment pour chaque canal.

En somme, Propsim C8 est un simulateur temps réel du canal radio qui permet la simulation simultanée de 8 canaux radio large bande. Multicanal et évolutif, il supporte la simulation de canal de toutes les normes de communications connues et des modèles de canaux depuis la 2G.

Ces modèles peuvent être définis librement par un utilisateur et mis en œuvre dans la configuration matérielle existante. Les capacités de Propsim C8 à mettre en œuvre des simulations à partir de fichiers fournissent la possibilité de rejouer des données de mesures du canal. Propsim C8 contient également l'outil de modélisation de canal pour utiliser des modèles de canal semi-déterministes. Propsim C8 est également capable de simuler les canaux mesurés comme tous les produits de la famille Propsim. La dernière version de Propsim C8 supporte des configurations 4×4, avec une seule unité de simulation, mais des configurations plus complexes, par exemple 8×8, peuvent être étudiées avec la synchronisation de plusieurs unités de simulation.

L'implémentation numérique complète fournit trois niveaux d'utilisation du simulateur : bande de base numérique, bande de base analogique et RF. Le simulateur peut être utilisé pour tester les systèmes pendant tout le cycle de développement, depuis la recherche et la conception d'algorithmes en bande de base, jusqu'à l'examen de la conformité, des performances et de la fonctionnalité des systèmes.

Litepoint : Le simulateur TrueChannel de LitePoint [55] est un simulateur de canal dynamique, extensible, bidirectionnel et à double bande conçu pour la validation et la caractérisation des normes 802.11 a/b/g et 802.11n MIMO. Il peut également servir aux tests de performances d'une part pour les concepteurs de systèmes qui les développent et, d'autre part, pour les réalisateurs des produits qui optimisent les performances des systèmes réalisés.

FIG. 1.16 – Simulateur matériel TrueChannel de LitePoint dans sa configuration 4×4.

Le simulateur comprend, entre autres, des modules d'interfaces RF et des modules d'acquisition émetteur/récepteur. Ces modules constituent un ensemble d'unités facilement accessible ce qui lui permet d'être extensible de la configuration 2×2 à la configuration 4×4. Il dispose également d'une interface graphique et d'une compatibilité avec MATLAB notamment pour pouvoir automatiser certaines applications de tests si l'utilisateur le souhaite. Il supporte les fréquences autour de 2,4 GHz et 5 GHz avec des largeurs de bande de 20 MHz et de 40 MHz. La simulation utilise les processeurs dédiés au traitement du signal numérique pour mettre en application, entre autres, les modèles de canaux définis par l'utilisateur. Les modèles de canaux peuvent être changés dynamiquement à partir d'une séquence prédéterminée par l'utilisateur. De plus, une fois une condition d'exploitation choisie, le simulateur fonctionne sans l'intervention de l'utilisateur, il peut configurer automatiquement et en temps réel le système. Par ailleurs, il simule les liens bidirectionnels.

De façon à mieux appréhender les différentes spécificités de chaque simulateur, nous avons dressé un récapitulatif de leurs principales caractéristiques dans le tableau 1.1.

Simulateurs / **Caractéritiques**	**ARC Seibersdorf Research ARC SmartSim**	**Spirent Communications SR5500**	**Azimuth ACE 400NB**	**Elektrobit Propsim C8**	**Litepoint TrueChannel**
Bandes de fréquences	450 - 6000 MHz	400 - 2700 MHz	802.11 a,b,g,h,i,n	360 MHz et 6 GHz	2,4 GHz et 5 GHz
Largeurs de bande	40 MHz	26 MHz	40 MHz	70 MHz (100 MHz)	20 MHz et 40 MHz
Nombre de canaux	1 à 8 (extensible à 64)	2 à 8	32	8	jusqu'à 32
Nombre de trajets	jusqu'à 36	24	18 par canal	non précisé	non précisé
Nombre de bits	14 bits en entrée 16 bits en sortie	non précisé	12 bits en entrée / 16 bits en sortie	non précisé	non précisé
Séquences de test	GSM, UMTS, WLAN, WiMAX	EDGE, GSM, GPRS, 802.11a/b/g, WiMAX, etc	802.11 a,b,g,h,i,n	GSM, EDGE, UMTS, WLAN, WiMAX, etc	802.11 a,b,g,n
Modèles de canal inclus	COST-259, avec extension MIMO	non précisé	802.11n	Standard 3GPP, GSM, ITU3G, JTC	non précisé
Bidirectionnel	non précisé	non précisé	oui	non précisé	oui

TAB. 1.1 – Tableau comparatif des principales caractéristiques de différents simulateurs matériels.

1.4 Conclusion

Avant d'atteindre les antennes réceptrices, via plusieurs trajets en général, le signal émis subit l'effet du canal qui se comporte comme un filtre linéaire. Le canal étant variant dans

le temps, le filtre équivalent sera également variant dans le temps. Les signaux à l'entrée et à la sortie de ce filtre peuvent être décrits dans le domaine temporel ou fréquentiel, ce qui conduit à une représentation selon quatre fonctions que nous avons détaillées en nous basant sur la démarche de Bello [16]. Si une seule des fonctions est connue, il est possible de calculer les autres pour bien caractériser et modéliser le canal.

Concernant les modèles, on peut dire qu'en général un modèle est dit déterministe lorsque l'environnement de propagation est parfaitement connu. Ce type de modèle n'est pas généralisable, puisqu'il est très spécifique à l'environnement à étudier, vu qu'il s'applique à une configuration donnée dans des conditions particulières, et qu'il nécessite une bonne maîtrise de tous les paramètres du canal.

A l'inverse, les modèles statistiques permettent de décrire l'évolution du canal par des lois statistiques. En effet, étant donné qu'il existe de nombreux paramètres propres à des phénomènes physiques et dont les variations ne sont pas toujours parfaitement connues, il devient alors plus très utile de les modéliser par des lois statistiques.

Ainsi, le simulateur matériel que nous allons présenter doit pouvoir accepter le plus grand nombre de modèles différents afin de pouvoir offrir de larges possibilités de simulation du canal de propagation, dans différents environnements.

Après avoir présenté le canal de propagation, la conception du simulateur matériel de propagation peut être décrite de façon générale. La conception implique des outils de prototypage et des composants programmables qui seront également détaillés dans le chapitre suivant.

Chapitre 2

Simulateur matériel de propagation

Sommaire

2.1 Cahier des charges du simulateur matériel 30

2.1.1 Description du bloc radiofréquence 32

2.1.2 Description du bloc « Modèles de canal » 34

2.1.3 Description du bloc « Partie numérique » 38

2.2 Plate-forme de prototypage 42

2.2.1 Choix du composant programmable 42

2.2.2 Choix des cartes électroniques 44

2.2.3 Génération et distribution de l'horloge 47

2.3 Conclusion 48

La demande de débits de plus en plus élevés pour les communications mobiles est l'un des vecteurs principaux de l'industrie des télécommunications. Il existe de nombreuses façons de gagner en performance en changeant quelques paramètres techniques, par exemple en augmentant la puissance d'émission, ou bien en utilisant les systèmes MIMO pour augmenter le débit et la qualité de la liaison. Mais, quelles que soient les améliorations réalisées, il faudra toujours examiner le système dans des conditions réelles. Pour cela, le simulateur matériel est un outil qui permet d'effectuer des tests dans des conditions réalistes, tout en restant en laboratoire.

Après avoir décrit les principaux phénomènes de propagation, quelques modèles pour alimenter le simulateur ainsi que plusieurs simulateurs matériels existants, nous allons détailler la conception du simulateur matériel de canaux de propagation MIMO. Le cahier des charges du simulateur est présenté comme un point de départ, dans lequel nous décrirons les blocs qui le composent, notamment les blocs de la partie numérique. Nous y présenterons succintement les architectures proposées, avant qu'elles ne soient détaillées dans les chapitres suivants, ainsi que le choix de plusieurs paramètres utiles pour concevoir le simulateur.

Dans une seconde partie, nous allons présenter le choix du composant programmable et de la plate-forme de prototypage qui servira à l'implémentation d'une partie de l'architecture du bloc numérique du simulateur sur composant programmable.

2.1 Cahier des charges du simulateur matériel

Le simulateur doit reproduire le comportement d'un canal de propagation MIMO pour un environnement spécifique, à l'extérieur ou à l'intérieur des bâtiments. Il permettra de vérifier le fonctionnement correct des nouveaux systèmes de communications et de comparer leurs performances dans les mêmes conditions de test.

Le simulateur doit fonctionner avec des signaux RF dans la bande des 2 GHz pour l'UMTS et des 5 GHz pour les systèmes WLAN. Afin de faire des tests d'interférences entre les canaux adjacents des systèmes UMTS, on a choisi de considérer trois canaux successifs, donc une largeur de bande de 3×5 MHz. La largeur des bandes de fréquences B est de 15 MHz pour l'UMTS et 20 MHz pour le WLAN.

De plus, l'UMTS utilise deux modes de fonctionnement : le mode TDD (*Time Division Duplex*), où la même bande de fréquences est utilisée pour la voie montante et pour la voie descendante alternativement, et le mode FDD (*Frequency Division Duplex*) ayant différentes bandes de fréquences pour ces deux voies. En outre, selon la puissance des signaux émis, le simulateur doit pouvoir accepter des signaux d'entrée sur une large plage de valeurs de puissance, ce qui implique un contrôle de puissance en entrée du simulateur.

D'autre part, le nombre de blocs RF est déterminé par le nombre d'antennes d'émission et de réception. Pour un émetteur avec N_T antennes et un récepteur utilisant N_R antennes, le simulateur MIMO doit avoir $N_T N_R$ canaux SISO bidirectionnels, soit $2N_T N_R$ canaux SISO unidirectionnels. Le choix de 16 canaux SISO duplex semblait être un bon compromis entre utilité et complexité, en admettant que N_T et N_R soient inférieurs ou égaux à 8 et le produit $N_T N_R$ inférieur ou égal à 16. Mais, le coût des cartes électroniques nécessaires à l'implémentation a imposé une diminution du nombre de canaux qu'il sera possible de simuler. En l'occurrence, nous verrons plus tard que nous avons 3 cartes électroniques, chacune avec deux convertisseurs analogiques-numériques et deux convertisseurs numériques-analogiques, ce qui fait un maximum de 6 antennes en émission et 6 antennes en réception. Le produit $N_T N_R$ devra donc être inférieur ou égal à 12 canaux SISO, sans dépasser 6 antennes en émission et 6 en réception.

La conception et la réalisation des blocs RF pour les systèmes UMTS ont été accomplies dans le précédent projet SIMPAA [56]. Dans son état actuel, le bloc RF permet, pour la partie « réception », de transposer le signal reçu en un signal de fréquence intermédiaire (FI), puis en un signal basse fréquence de largeur de bande B, et réciproquement pour la partie « émission ».

Les objectifs de l'actuel projet régional SIMPAA 2 sont principalement le bloc « Modèles du canal » traité dans le cadre de la thèse de H. Farhat [57, 58] et le bloc « Partie numérique » du simulateur MIMO qui nous concerne. Ces blocs sont grisés dans le schéma fonctionnel de la figure 2.1.

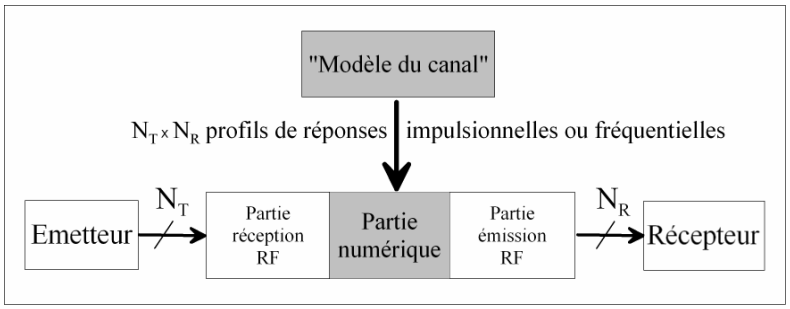

FIG. 2.1 – Schéma fonctionnel du simulateur de canal MIMO unilatéral.

Au début de l'étude, le cahier des charges du simulateur matériel reprend celui du projet RNRT SIMPAA, avec une extension au WLAN. En prenant en compte les besoins et en regardant ce qui est actuellement faisable, les spécifications sont élaborées et peuvent être résumées dans le tableau 2.1.

	SIMPAA	**SIMPAA 2**
Type d'environnement	urbain, suburbain, rural, intérieur, montagneux, pénétration	
Bandes de fréquences	1,9 à 2,2 GHz (UMTS*)	1,9 à 2,2 GHz (UMTS*) et 2.4 à 5 GHz (WLAN)
Largeurs de bande	15 MHz (UMTS)	15 MHz (UMTS) et 20 MHz (WLAN)
Nombre entrées/sorties RF	2 à 10	2 à 6
Nombre de trajets	> 100	
Modèles de canal	Tous les types de modèles	
Bidirectionnel	oui	
Nombre de bits	10	14
Retard maximal	20 µs	
Vitesse maximale relative	500 km/h (UMTS)	640 km/h (UMTS) et 25 km/h (WLAN)
Décalage Doppler maximal	1000 Hz (UMTS)	1200 Hz (UMTS) et 115 Hz (WLAN)
Contrôle de puissance	Dynamique CAG : 80 dB Dynamique instantanée : 40 dB	

* UMTS : modes duplex possibles TDD et FDD

TAB. 2.1 – Résumé des spécifications du simulateur matériel de canaux de propagation MIMO.

2.1.1 Description du bloc radiofréquence

Pour un canal SISO unidirectionnel, la partie « réception » du bloc RF réalise une conversion vers les basses fréquences. Sa fonction est d'obtenir un signal réel et continu. La partie « émission » du bloc RF réalise la conversion du signal de sortie du bloc numérique (sortie du CNA) vers les hautes fréquences. Le schéma bloc de la partie RF UMTS est donné à la figure 2.2.

Pour le récepteur et l'émetteur RF, les blocs « Atténuateur » et « Amplificateur » sont commandés par l'intermédiaire du bus de contrôle qui agit différemment selon qu'il doit ajuster la puissance de réception d'une part ou la puissance d'émission d'autre part.

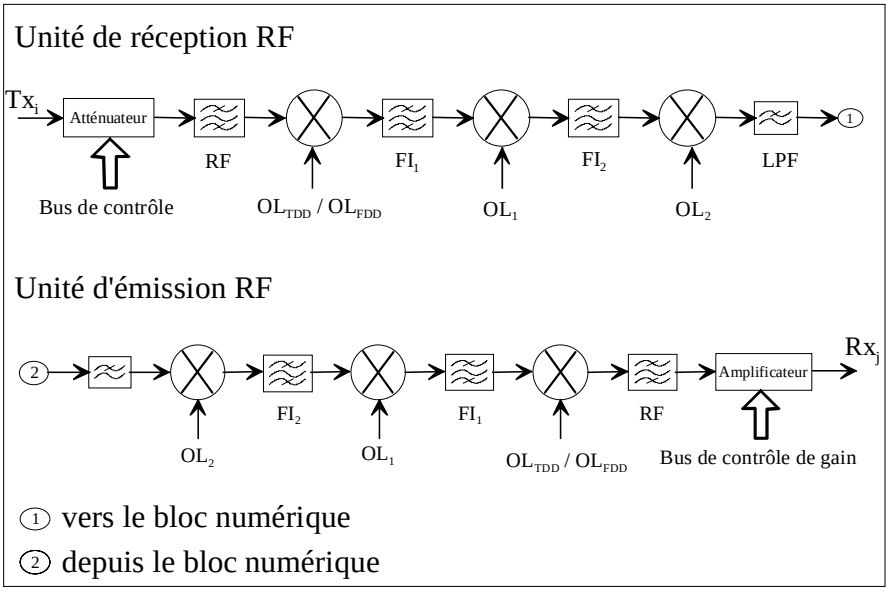

FIG. 2.2 – Schéma fonctionnel d'un bloc RF UMTS pour un canal SISO unidirectionnel.

Dans l'unité de réception RF, les oscillateurs locaux OL_{TDD} et OL_{FDD} permettent d'effectuer la conversion du signal RF vers le signal FI_1, en tenant compte des modes TDD et FDD qui utilisent différentes bandes de fréquences. Puis, le troisième oscillateur local OL_1 permet la conversion de FI_1 vers FI_2. Le filtre passe-bande FI_2 doit effectuer une bonne réjection des fréquences en dehors de la bande utile pour éviter des problèmes de recouvrement du spectre. L'oscillateur local OL_2 et le filtre passe-bas donnent un signal réel dans la bande $[0, B]$ qui est appliqué à l'entrée du bloc numérique.

En sortie du bloc numérique, la version passe-bas du signal analogique $y_k(t, \tau)$ est convertie en un signal RF par la partie « émission » du bloc RF. L'amplificateur RF final restitue le niveau de signal approprié en tenant compte de l'atténuation introduite par l'atténuateur contrôlé par le bloc de CAG (Contrôle Automatique de Gain).

La figure 2.3 montre le bloc RF qui correspond à un canal SISO pour la norme UMTS. On peut remarquer la partie de réception (en haut) et d'émission (en bas), ainsi que l'unité de raccordement (partie gauche) qui permet le lien entre l'antenne et les parties de réception et d'émission RF. Ce bloc RF a été réalisé en technologie imprimée pendant le projet SIMPAA précédent.

Pour un canal MIMO bidirectionnel, les parties RF doivent être réversibles pour le mode TDD uniquement. En effet, la partie réception devient alors la partie émission, et inversement, grâce à l'unité de raccordement que l'on peut voir sur la photo 2.3.

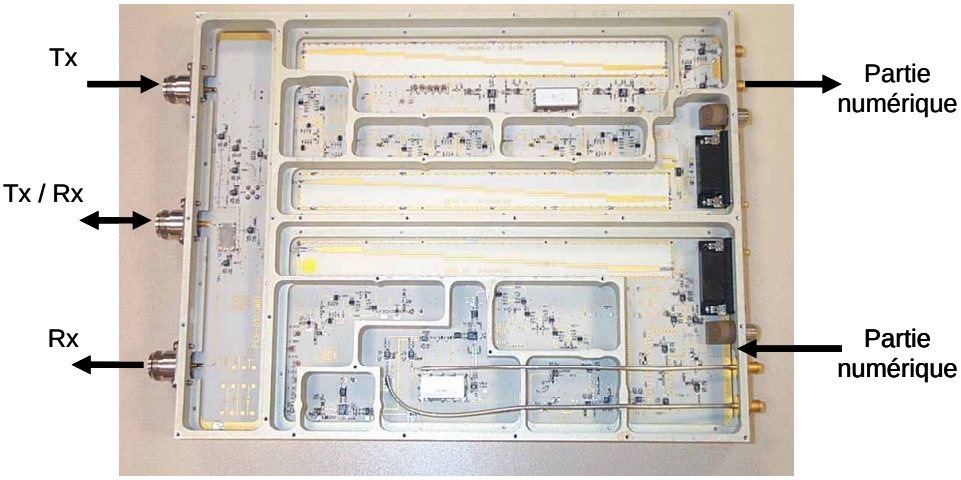

FIG. 2.3 – Bloc RF UMTS d'un canal SISO unidirectionnel (réception et émission).

Les parties RF présentées étaient prévues pour travailler en bande de base donc avec des signaux complexes. Or, nous verrons plus tard que les cartes de prototypages choisies contiennent deux convertisseurs analogiques-numériques et deux convertisseurs numériques-analogiques. Dans ce cas, en bande de base, il faudrait utiliser un convertisseur pour la partie réelle et un convertisseur pour la partie imaginaire du signal complexe. Une carte n'accepterait qu'un seul signal d'entrée et un seul signal de sortie, donc un seul canal SISO. Si nous considérons des signaux réels, on peut simuler quatre canaux SISO par carte.

Pour les applications UMTS de SIMPAA 2, il sera utile de modifier cette partie RF pour que la partie « réception » réalise une conversion afin d'obtenir un signal réel ayant un spectre dans la bande de fréquences $[0, B]$. Ainsi, le bloc numérique peut éviter le calcul du produit de convolution en complexe et donc utiliser moins de ressources, ce qui diminue le coût du matériel. Pour les applications WLAN, le bloc radiofréquence doit être créé en utilisant une architecture similaire.

2.1.2 Description du bloc « Modèles de canal »

Les modèles de canal fournissent au simulateur des profils $h(t_k, \tau)$ ou $H(t_k, f)$ calculés à partir d'un modèle théorique (déterministe ou statistique) ou d'un modèle standardisé. Les profils peuvent également être obtenus à partir de mesures en utilisant un sondeur de canal temporel MIMO, comme celui conçu et réalisé à l'IETR [17] et illustré à la figure 2.4.

FIG. 2.4 – Emetteur et récepteur du sondeur temporel réalisé à l'IETR.

De nombreuses études ont été réalisés sur les sondeurs de canal [59]. Il existe deux types de sondeurs de canal : les sondeurs temporels qui mesurent la réponse impulsionnelle du canal et les sondeurs fréquentiels qui donnent la réponse fréquentielle du canal. Pendant la mesure de tous les canaux SISO, le canal MIMO doit être pratiquement invariable. Ceci limite la vitesse du mobile et donc la valeur maximale du décalage Doppler.

En réalité, pour chaque canal SISO unilatéral, le sondeur temporel donne l'enveloppe complexe mesurée de sa réponse impulsionnelle.

Enveloppe complexe :

La notion d'enveloppe complexe permet de définir, pour un signal réel à bande limitée, un signal complexe équivalent en bande de base. Une réponse impulsionnelle $h(t)$ peut alors être exprimée par :

$$h(t) = a(t)\cos(2\pi f_0 t + \phi(t)) \tag{2.1}$$

mais il peut aussi s'écrire suivant ses deux composantes réelle (ou en phase) $h_p(t)$ et imaginaire (ou en quadrature) $h_q(t)$, telles que :

$$h(t) = h_p(t)\cos(2\pi f_0 t) - h_q(t)\sin(2\pi f_0 t) \tag{2.2}$$

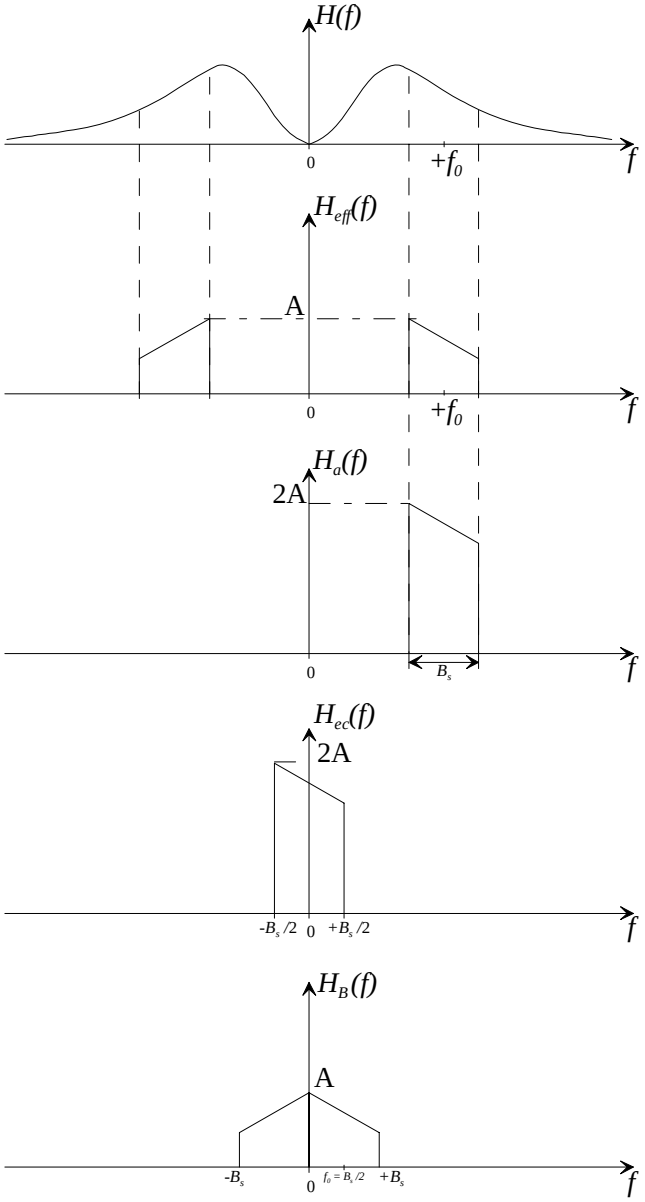

FIG. 2.5 – Enveloppe complexe de la réponse impulsionnelle du canal.

avec $h_p(t) = a(t)\cos(\phi(t))$ et $h_q(t) = a(t)\sin(\phi(t))$. Comme on observe la réponse fréquentielle $H(f)$ du canal à travers le filtre passe-bande de l'appareil de mesure, sa représentation fréquentielle est illustrée par la réponse fréquentielle effective $H_{eff}(f)$ sur la figure 2.5. Le signal $h(t)$ étant réel, son spectre vérifie la symétrie hermitienne donc :

$$H_{eff}(-f) = H^*_{eff}(f). \tag{2.3}$$

On obtient le signal dit analytique $h_a(t)$, associé au signal $h_{eff}(t)$ par le fait qu'il contient uniquement la partie du spectre de $H_{eff}(f)$ située du côté des fréquences positives et amplifiée par 2, cette partie étant suffisante pour retrouver toutes les informations sur le signal $h(t)$. Le signal analytique $h_a(t)$ du signal réel $h(t)$ peut alors être défini par :

$$h_a(t) = \mathcal{F}^{-1}\{(1 + sgn(f)).H_{eff}(f)\} \tag{2.4}$$

où sgn(f) est la fonction signe qui vaut -1 pour les fréquences négatives du signal et 1 pour les fréquences positives.

Le signal analytique est donc complexe et peut s'écrire :

$$h_a(t) = (h_p(t) + jh_q(t)).e^{j2\pi f_0 t} \tag{2.5}$$

Par glissement sur l'axe fréquentiel, la transformée de Fourier $H_a(f)$ du signal analytique $h_a(t)$ est centrée autour de la fréquence $f = 0$, ce qui correspond à la transformée de Fourier de l'enveloppe complexe du signal réel $h_{ec}(t)$. Son expression est alors donnée par :

$$h_{ec}(t) = h_a(t).e^{-j2\pi f_0 t}. \tag{2.6}$$

Compte tenu de l'expression de $h_a(t)$, on obtient :

$$h_{ec}(t) = h_p(t) + jh_q(t) = a(t).e^{j\phi(t)} \tag{2.7}$$

En considérant les équations (2.5) et (2.6), on peut retrouver le signal réel $h(t)$ comme étant la partie réelle du signal analytique mais aussi la partie réelle de l'enveloppe complexe décalée autour de la fréquence centrale, c'est-à-dire :

$$h(t) = \Re[h_a(t)] = \Re[h_{ec}(t).e^{j2\pi f_0 t}] \tag{2.8}$$

Ces étapes sont illustrées à la figure 2.5 dans le domaine fréquentiel.

Le sondeur de canal temporel fournit l'enveloppe complexe $h_{ec}(t)$ du signal en bande de base avec B_s la largeur de bande du sondeur, soit dans notre cas, 100 MHz. L'enveloppe complexe peut s'écrire :

$$h_{ec}(t) = h_p(t) + jh_q(t) \tag{2.9}$$

Son équivalent $H_{ec}(f)$ dans le domaine fréquentiel est illustré à la figure 2.5.

Pour retrouver le signal réel à partir de l'enveloppe complexe fournie par le sondeur, il suffit de filtrer la réponse mesurée par un filtre de bande B puis de reprendre la relation (2.8) pour $f_0 = \frac{B}{2}$, ce qui donne :

$$h_B(t) = h_p(t)\cos(2\pi\frac{B}{2}t) - h_q(t)\sin(2\pi\frac{B}{2}t) \tag{2.10}$$

$$= h_p(t)\cos(\pi Bt) - h_q(t)\sin(\pi Bt) \tag{2.11}$$

et on retrouve bien ainsi le signal réel de l'expression (2.2), dans notre cas particulier où nous souhaitons travailler avec une réponse impulsionnelle réelle dont le spectre occupe la bande [0, B].

Obtention des profils de canal :

Une possibilité pour obtenir des profils de réponses impulsionnelles est d'utiliser les résultats de mesures avec la relation (2.11) pour déterminer les profils réels correspondant à la bande [0, B]. Par la suite, il est possible soit d'utiliser ces réponses impulsionnelles soit de calculer, pour chaque instant t, les réponses fréquentielles $H_{i,k}(t, f)$, par transformée de Fourier des réponses impulsionnelles $h_{i,k}(t, \tau)$ du canal de propagation. Périodiquement, ces réponses impulsionnelles ou fréquentielles doivent être actualisées, afin de simuler l'évolution en temps du canal. La période de rafraîchissement des réponses impulsionnelles ou fréquentielles est déterminée par le temps de cohérence du canal, donc par la vitesse du mobile et l'environnement de propagation. Comme le nombre de fichiers enregistrés est limité, il peut être nécessaire de rejouer périodiquement l'ensemble de ces réponses impulsionnelles ou fréquentielles.

Une autre manière d'exploiter les résultats est d'utiliser un modèle de canal MIMO (statistique, déterministe ou hybride) pour chaque type d'environnement de propagation (intérieur, urbain, rural, montagneux, etc.) et de générer des profils de réponses impulsionnelles $h_{i,k}(t, \tau)$ ou fréquentielles $H_{i,k}(t, f)$ à l'extérieur du simulateur. Dans ce cas, les modules de transformée de Fourier rapide directe (FFT - *Fast Fourier Transform*) sont requis pour calculer les réponses fréquentielles $H_{i,k}(t, f)$ utilisées par l'expression (1.10). Finalement, pour chaque antenne de réception, un module de Transformée de Fourier rapide inverse (IFFT - *Inverse Fast Fourier Transform*) est mis en œuvre pour obtenir le signal de sortie $y_k(t, \tau)$.

En résumé, pour chaque environnement de propagation, les profils de réponses impulsionnelles du canal de propagation sont soit obtenus par des mesures effectuées avec le sondeur de canal MIMO opérant dans le domaine temporel et réalisé à l'IETR, soit calculés à partir d'un modèle théorique (déterministe ou statistique) ou d'un modèle standardisé. En utilisant la transformée de Fourier pour chaque environnement de propagation,

un grand ensemble de réponses fréquentielles successives peut ainsi être obtenu. Selon la gamme de fréquences (UMTS ou WLAN) et l'environnement de propagation, les réponses impulsionnelles ou fréquentielles de chaque canal SISO sont chargées dans la mémoire du bloc « Modèles de canal » de la figure 2.1 du simulateur matériel MIMO.

2.1.3 Description du bloc « Partie numérique »

2.1.3.1 Deux architectures envisagées

Comme nous l'avons vu avec les expressions (1.9) et (1.10), la partie numérique peut effectuer les calculs dans le domaine temporel par des opérations de convolution ou dans le domaine fréquentiel avec des modules de transformées de Fourier.

Dans un environnement « outdoor », les réponses impulsionnelles peuvent durer jusqu'à quelques dizaines de microsecondes, plus d'échantillons sont nécessaires pour les décrire. De plus, les réponses impulsionnelles doivent être plus souvent rafraîchies car l'environnement est susceptible d'évoluer plus rapidement. En effet, la vitesse de déplacement d'un mobile dépasse largement celle d'une personne qui se déplace à l'intérieur d'un bâtiment (milieu « indoor »). Le nombre de profils à stocker et à traiter est donc plus grand. Les opérations de convolution (1.9) seraient plus coûteuses à réaliser en terme de ressources matérielles, avec plusieurs filtres à réponses impulsionnelles finies (FIR - *Finite Impulse Response*), donc avec plusieurs multiplieurs et additionneurs, en fonction du nombre de coefficients à mettre en œuvre. Ainsi, pour les environnements outdoor, on peut choisir une architecture qui effectue les calculs dans le domaine fréquentiel. Dans ce cas, on utilise seulement un multiplieur complexe mais on a besoin d'un module FFT pour chaque canal SISO, puis d'un module IFFT. Cette architecture est présentée à la figure 2.6.

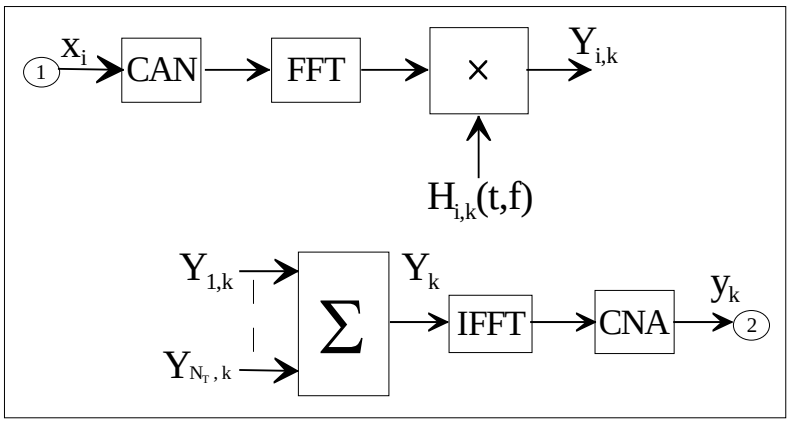

FIG. 2.6 – Schéma fonctionnel de la partie numérique dans le domaine fréquentiel.

La transformée de Fourier du signal d'entrée, obtenue en sortie du bloc FFT, est multipliée par la réponse fréquentielle $H_{i,k}(t, f)$ du canal SISO, provenant du bloc « Modèles de canal » décrit précédemment.

Le signal $Y_k(t, f)$ est la somme des contributions de chaque canal SISO. Ce signal est tronqué selon le nombre de bits de l'entrée du bloc IFFT qui donne les échantillons du signal de sortie basse fréquence $y_k(t, \tau)$. Après le convertisseur numérique-analogique, le signal analogique résultant représente l'entrée de la partie RF d'émission qui assure la conversion vers les hautes fréquences.

D'autre part, le bloc numérique, décrit à la figure 2.7, fonctionne dans le domaine temporel.

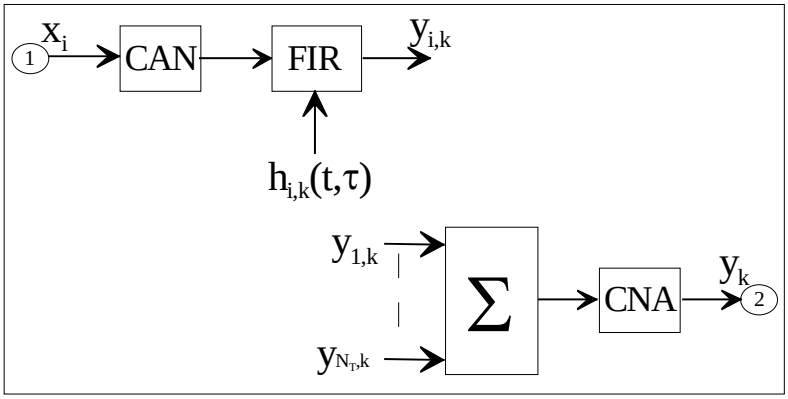

FIG. 2.7 – Schéma fonctionnel de la partie numérique dans le domaine temporel.

En effet, pour un environnement « indoor », les réponses impulsionnelles peuvent être jusqu'à 20 fois plus courtes (environ 1 μs) comme nous le verrons dans le tableau 2.2, ce qui est incompatible avec la grande latence des modules FFT et IFFT (quelques dizaines de μs). Il est donc plus judicieux d'utiliser des filtres FIR qui effectuent des produits de convolution dans un environnement « indoor ». Plusieurs multiplieurs réels sont utilisés, leur nombre dépend de la durée de la réponse impulsionnelle du canal et de la fréquence d'échantillonnage.

Le filtre FIR calcule le produit de convolution entre le signal d'entrée échantillonné et la version numérique de la réponse impulsionnelle de canal $h_{i,k}(t, \tau)$. Comme pour le domaine fréquentiel, le signal $y_k(t, \tau)$ de la $k^{ième}$ antenne de réception est la somme des signaux SISO $y_{i,k}(t, \tau)$. Puis, cette somme doit être tronquée selon le nombre de bits du convertisseur numérique-analogique.

Ainsi, afin d'avoir un bon compromis entre la complexité et la latence du système, deux solutions ont été envisagées : une approche dans le domaine temporel avec des filtres FIR pour les environnements intérieurs ou indoor et une approche dans le domaine fréquentiel

avec des modules de transformées de Fourier rapides directes (FFT) et inverses (IFFT) pour les environnements extérieurs ou outdoor.

2.1.3.2 Choix des paramètres

Dans le cahier des charges décrit précédemment, nous avons vu quelles fréquences et quelles bandes nous intéressent. On se situe dans une bande, soit de 15 MHz pour les systèmes UMTS, soit de 20 MHz pour les systèmes WLAN, entre 0 et la fréquence maximale qui est donc égale à la bande. Sachant que les filtres RF et FI utilisés ne sont pas parfaits et en utilisant le théorème de Shannon, la fréquence d'échantillonnage f_s doit être choisie supérieure ou égale au double de la fréquence maximale du spectre. On choisit alors une fréquence d'échantillonnage de 40 MHz pour les systèmes UMTS et de 50 MHz pour les systèmes WLAN. Ce choix permet d'avoir une fréquence d'échantillonnage raisonnablement basse et évite les problèmes de recouvrement du spectre.

A partir de la figure 1.9, si on considère un instant t fixé, on isole un profil de la réponse impulsionnelle. Comme indiqué sur la figure 2.8, on peut fixer un seuil à -30 dB par rapport à la valeur maximale du module de la réponse impulsionnelle en dB et on définit la fenêtre temporelle $W_{\tau eff}$ en dehors de laquelle le module de la réponse impulsionnelle est considéré nul. En réalité, on utilise la valeur minimale des τ_0 de l'ensemble des profils pour déterminer la fenêtre temporelle $W_{\tau eff}$. Le fait d'utiliser la valeur minimale des τ_0 a pour inconvénient

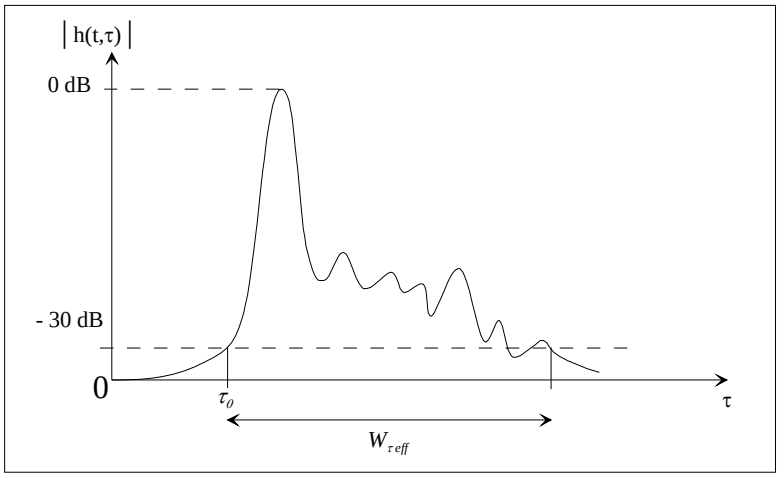

FIG. 2.8 – Représentation d'un profil de réponse impulsionnelle pour un instant t fixé.

de devoir analyser tout le paquet de profils utilisés pour la simulation, déterminer la valeur de τ_0 pour chaque profil $|h(t,\tau)|$ et enfin calculer $min\{\tau_0\}$. Mais ce calcul apporte de nombreux avantages. En effet, de cette façon, non seulement on utilise moins d'espace mémoire pour stocker les profils, mais aussi le nombre d'échantillons décrivant $|h(t,\tau)|$ est plus petit, la taille des filtres FIR ou des blocs FFT/IFFT diminue également, ainsi que

la latence de calcul. De plus, si la latence du bloc « Partie numérique » est inférieure au retard absolu τ_0, il n'y aura pas d'erreur due au calcul de la partie numérique.

Ainsi, la transformée de Fourier et les produits de convolution seront calculés seulement pour la fenêtre $W_{\tau eff}$, afin de réduire la taille du bloc numérique.

Pour ces divers environnements et connaissant la fréquence d'échantillonnage, la longueur du filtre FIR ou des blocs FFT/IFFT est calculée par :

$$N_{eff} = \frac{W_{\tau\ eff}}{T_s} = W_{\tau\ eff}.f_s \tag{2.12}$$

où $W_{\tau eff}$ est la largeur de la fenêtre temporelle effective de la réponse impulsionnelle de canal, c'est-à-dire la largeur de l'intervalle de temps où la réponse impulsionnelle peut être considérée non nulle [6], [40]. Pour l'environnement outdoor, en raison de l'utilisation des blocs FFT et IFFT, la valeur de N est choisie comme la valeur de puissance de 2 la plus proche de N_{eff}. La fenêtre résultante $W_\tau = N.T_s$ est également indiquée dans le tableau 2.2 qui récapitule quelques paramètres utiles, selon les environnements de propagation considérés.

	Environnements	Tailles de cellule	$W_{\tau\ eff}$ (µs)	N	W_τ (µs)
UMTS (B = 15 MHz) (f_s = 40 MHz)	Rural Urbain Indoor	2 - 20 km 0.4 - 2 km 20 - 400 m	20 3.7 0.7	512 128 28	12.8 3.2 0.7
WLAN (B = 20 MHz) (f_s = 50 MHz)	Bureau Indoor Outdoor	40 m 50 - 150 m 50 - 150 m	0.39 0.73 1.16	20 37 64	0.40 0.74 1.28

TAB. 2.2 – Tableau de paramètres.

Par ailleurs, afin de simuler une réponse impulsionnelle ou fréquentielle, il est nécessaire de rafraîchir les profils de ces réponses. Une méthode pour déterminer la fréquence de rafraîchissement est d'appliquer le théorème de Shannon. Alors, la fréquence de rafraîchissement f_{raf} doit être d'au moins deux fois la fréquence Doppler maximale f_{dmax} définie au chapitre précédent.

$$f_{raf} \geq 2.f_{dmax} = 2.f_{e_{max}}.\frac{v}{c} \tag{2.13}$$

Pour connaître la fréquence nécessaire au rafraîchissement des profils de réponses du canal, on doit procéder à une estimation pour chaque environnement considéré. Comme on utilise, en général, la fréquence centrale f_0 comme fréquence émise f_e pour calculer la fréquence Doppler maximale, on peut calculer la fréquence de rafraîchissement dans le cas UMTS puis dans le cas WLAN.

Dans le cas UMTS, si un récepteur mobile se trouve dans un TGV ayant une vitesse v_1, un deuxième TGV qui le croise avec une vitesse v_2 modifie l'environnement en coupant des trajets existants ou en créant des trajets supplémentaires par réflexion. Ainsi, l'environnement bouge très vite en prenant comme référentiel ce récepteur. On additionne alors les vitesses des TGV, ce qui peut donner une vitesse relative $v = v_1 + v_2$ allant jusqu'à 640 km/h. Avec une fréquence centrale f_0 pour l'UMTS de 2 GHz et connaissant la célérité de l'onde c, on obtient, selon la relation précédente, une fréquence de rafraîchissement $f_{raf_{UMTS}}$ minimale d'environ 2,37 kHz.

Si on considère l'environnements indoor, on a une vitesse relative pouvant atteindre 50 km/h, par exemple, pour des véhicules dans un hangar ou un hall d'aéroport. Pour le WLAN, la fréquence centrale étant de 5 GHz, on obtient alors une fréquence de rafraîchissement $f_{raf_{WLAN}}$ minimale d'environ 462,96 Hz.

Nous réutiliserons ces paramètres importants ultérieurement.

Une autre méthode pour connaître la fréquence de rafraîchissement est de calculer la fonction d'autocorrélation de la réponse impulsionnelle par rapport à la variable temps. Le temps de cohérence, comme défini dans le chapitre précédent, peut alors être la valeur prise à 90 % de la valeur maximale du module de cette fonction d'autocorrélation. Ce temps de cohérence peut être le temps de rafraîchissement T_r, inverse de la fréquence de rafraîchissement que l'on souhaite obtenir.

Cette première partie a présenté le cahier des charges, les architectures proposées et les paramètres du simulateur matériel MIMO, nous pouvons maintenant présenter les raisons qui ont motivé le choix du matériel utilisé pour la mise en œuvre du simulateur.

2.2 Plate-forme de prototypage

Après avoir détaillé le principe et l'architecture numérique du simulateur et avant de montrer les premières expérimentations, nous allons détaillé la plate-forme de prototypage choisie. Ce choix passe par la détermination du composant programmable adéquat et des cartes électroniques qui l'utilisent.

2.2.1 Choix du composant programmable

Tout d'abord, le choix s'est porté sur un composant programmable FPGA (*Field Programmable Gate Array*) de la société Xilinx [60] plutôt que sur des DSP (*Digital Signal Processor*, processeurs dédiés au traitement du signal. En effet, le FPGA opère au niveau logique binaire, le temps de calcul est alors très faible par rapport à un DSP, qui exécute des instructions sur des registres. Pour faire mieux, il faudrait développer un ASIC (*Application-Specific Integrated Circuit*) et on pourrait alors diminuer, d'un facteur 5 environ, le temps de calcul (latence), mais cela dépend aussi des circuits utilisés à l'intérieur du

FPGA. Ici, on utilise surtout des multiplieurs qui sont déjà bien optimisés dans le FPGA, du coup le gain serait faible avec un ASIC.

A titre indicatif, on montre à la figure 2.9 l'évolution des composants programmables de la famille Virtex de Xilinx en nombre de « slices » qu'ils contiennent. Les « slices » sont des cellules qui regroupent chacune deux LUT et deux bascules de type D, pour la famille Virtex de Xilinx. On montre ainsi que l'implémentation d'une architecture numérique relativement complexe, difficilement réalisable dans un FPGA Virtex-II lors du projet SIMPAA, devient possible quelques années plus tard lors du projet SIMPAA 2 grâce à l'évolution de nouvelles générations de circuits programmables.

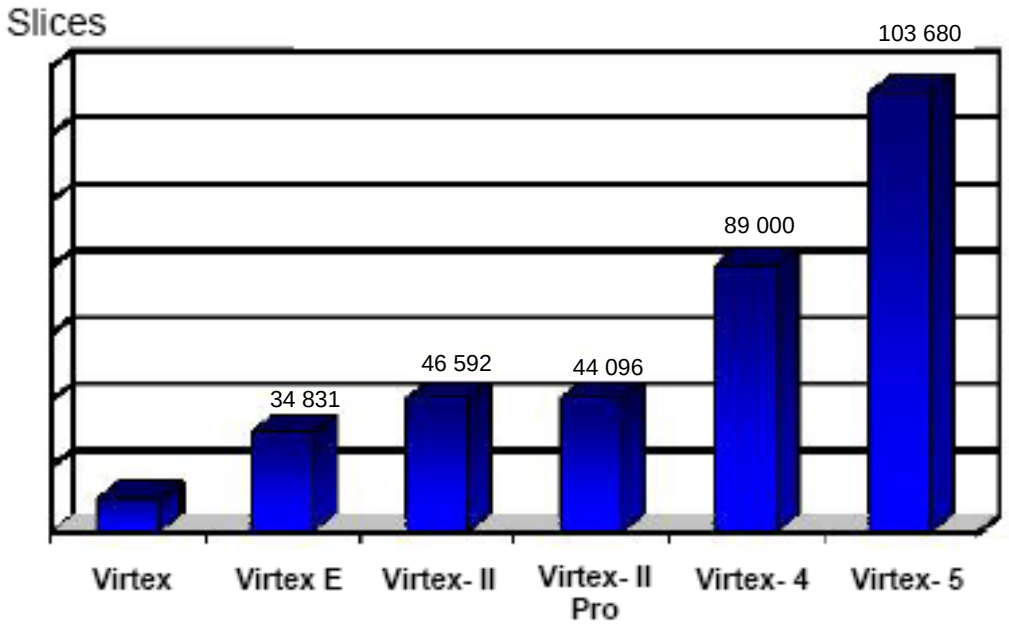

FIG. 2.9 – Evolution de la famille Virtex de Xilinx.

La famille des Virtex-5 n'étant pas encore disponible au moment du choix, nous avons opté pour la famille des Virtex-4 qui se compose de 3 membres [61] : les « LX » qui contiennent surtout de la logique, les « SX » qui contiennent de la mémoire et beaucoup de multiplieurs et les « FX » qui sont très complets et qui intègrent notamment un processeur PowerPC. Dans la mesure où l'architecture à filtres FIR demande beaucoup de multiplieurs, un des critères de choix du composant est le nombre de multiplieurs qu'il contient. De plus, le nombre de blocs « RAM », blocs de mémoire double-port nécessaires pour stocker les échantillons des profils de réponses impulsionnelles ou fréquentielles, est important à considérer également. Nous avons donc choisi le composant Virtex-4 « SX » qui est le mieux adapté aux applications de traitement numérique du signal. Cependant, selon la carte de prototypage choisie, le composant sera l'un des trois Virtex-4 « SX » : 25, 35 ou 55, qui diffèrent par le nombre de blocs qui les constituent.

Ces différents blocs sont principalement les blocs de mémoire double-port, cités plus haut et que l'on détaillera plus loin, les blocs de logique configurable (CLB - *Configurable Logic Block*) appelé simplement « logique », qui sont les éléments logiques de base d'un FPGA, et les blocs « XtremeDSP ». Le terme « XtremeDSP » peut être abusivement remplacé, dans différentes documentations, par le terme « DSP » ou « multiplieurs 18 × 18 ». Mais, les 192 blocs « XtremeDSP » contenus dans le composant Virtex-4 SX 35 sont le plus souvent simplement surnommés « multiplieurs », chacun contenant en fait un multiplieur 18 bits par 18 bits, un additionneur sur 48 bits et des registres [62]. Les blocs « RAM », ou blocs de mémoire de 18 kbits et à double-port, sont au nombre de 192 dans le composant Virtex-4 SX 35 mais ils peuvent être cascadés pour former de plus grands blocs de mémoire. Le fait que ces blocs de mémoire soient à double-port permet de lire une adresse pendant que l'on écrit sur une autre, ce qui peut s'avérer pratique pour certaines applications.

2.2.2 Choix des cartes électroniques

Parallèlement au choix du composant programmable et afin d'implémenter le simulateur matériel, plusieurs solutions de plate-forme de prototypage ont été étudiées. La solution retenue met en œuvre un ensemble d'outils de développement de la société Xilinx, nommé « XtremeDSP Development Kit-IV » pour Virtex-4 [63] et réalisé en collaboration avec la société Nallatech. Ce kit de développement fournit notamment une plate-forme complète pour des applications haute-performance de traitement du signal telles que la radio logicielle, la communication sans-fil, la gestion des réseaux...

La photo 2.10 montre qu'en fait ce kit contient un module Nallatech « BenADDA-IV » comprenant les convertisseurs, le quartz, deux banques de mémoire, le Virtex-4 et le Virtex-

Fig. 2.10 – Kit de développement XtremeDSP pour Virtex-4.

II (qui n'est pas visible sur cette photo). Ce module est intégré sur une carte Nallatech appelée « BenONE » qui, quant à elle, supporte le Spartan-II dédié aux interfaces JTAG, PCI et USB.

Cette carte intègre le FPGA Virtex-4 SX 35, suffisant pour notre application. En effet, même si le Virtex-4 SX 55 était préférable car il contient plus de multiplieurs (« XtremeDSP »), la carte de prototypage Nallatech n'existait pas avec ce composant programmable. Nous avons donc choisi la meilleure alternative du moment : la carte de prototypage Nallatech pour Virtex-4 [63] avec un FPGA Virtex-4 SX 35.

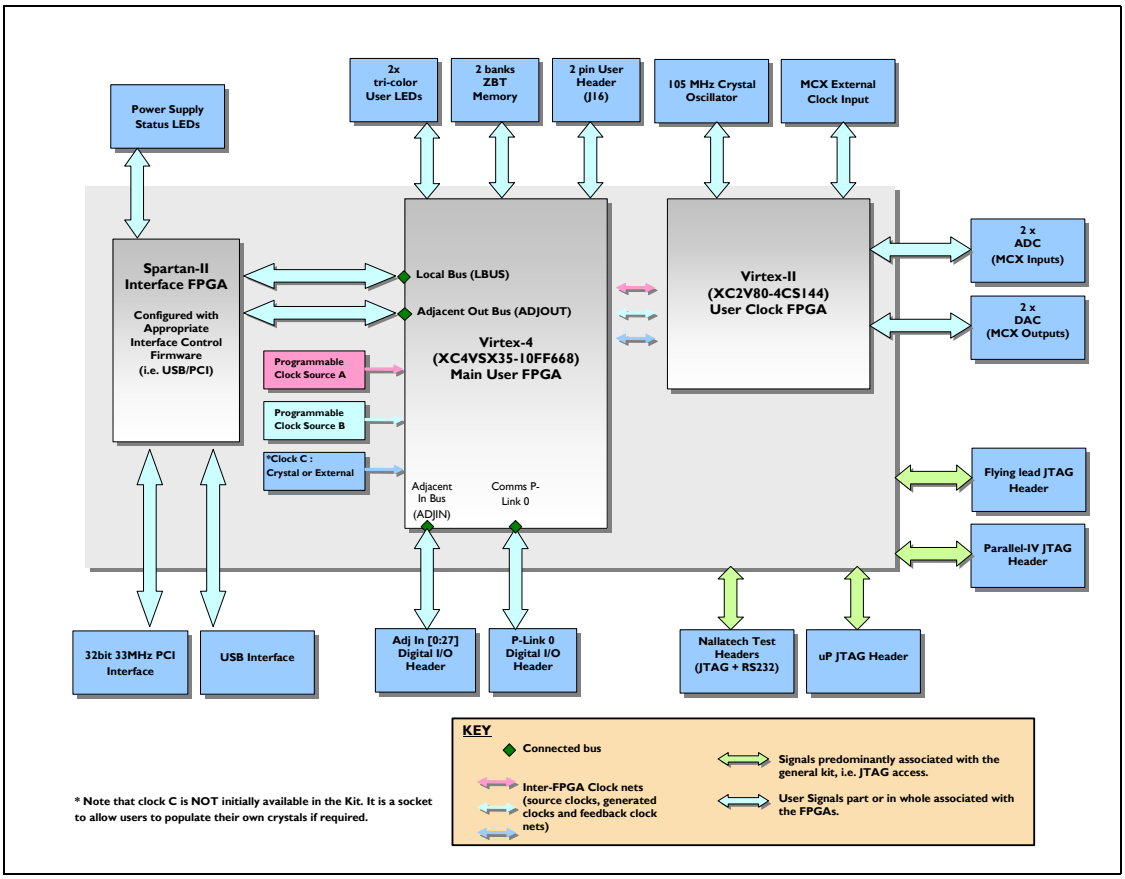

FIG. 2.11 – Diagramme fonctionnel du kit de développement XtremeDSP pour Virtex-4.

Cette carte s'est révélée très pratique car elle contient également d'autres composants qui peuvent servir à notre application tels que des mémoires, un quartz, deux convertisseurs analogiques-numériques (CAN) [64] et deux convertisseurs numériques-analogiques (CNA) [65] haute-performance, d'une résolution de 14 bits chacun. Concernant les convertisseurs, on a pu constater qu'ils peuvent fonctionner à la fréquence d'échantillonnage et dans la bande de fréquence qui nous intéressent. Nous verrons également, dans les chapitres suivants, que la résolution de 14 bits de ces convertisseurs permet d'obtenir une bonne

précision des résultats. D'autre part, cette résolution correspond à la meilleure résolution des convertisseurs du moment, ce qui nous a conforté dans le choix de la carte. Sur la figure 2.11, qui présente le diagramme de fonctionnement de cette carte de prototypage, on peut distinguer le FPGA Spartan-II de Xilinx, dédié aux interfaces JTAG, PCI et USB. Ce Spartan communique avec le Virtex-4 SX35 qui est le composant principal. Il nous permettra d'implémenter les architectures choisies et de reprogrammer le composant selon le type de canal choisi par l'utilisateur. Il est également relié notamment à deux banques de mémoire et au troisième FPGA de la carte, le Virtex-II [66], qui est spécialement dédié aux signaux d'horloge. Ce Virtex-II possède de ce fait une entrée pour une éventuelle horloge externe et une liaison vers un quartz de 105 MHz. Le Virtex-II est également indispensable à la communication avec les deux CAN et les deux CNA.

Pour simuler un canal de propagation MIMO, il est nécessaire de pouvoir simuler au moins 4 canaux SISO unidirectionnels. Mais, si on choisit de pouvoir simuler une configuration de plus de 4 canaux SISO unidirectionnels, nous verrons aux chapitres suivants que cela implique d'utiliser plusieurs cartes Nallatech comme celle présentée ci-dessus. En effet, le travail de la partie numérique en bande de base nécessiterait un CAN pour la partie réelle du signal d'entrée et un autre CAN pour la partie imaginaire. De même, pour un signal de sortie, on devrait utiliser un CNA pour sa partie réelle et un CNA pour sa partie imaginaire. L'utilisation des signaux en bande de base implique des calculs avec des signaux complexes, ce qui empêcherait de réaliser plus d'un canal SISO unilatéral par carte, ce qui est vraiment décourageant. On a alors choisi de travailler en réel afin de s'affranchir des calculs complexes propres aux signaux en bande de base pour ainsi réduire d'une façon significative les ressources. En travaillant en réel, on peut utiliser les deux CAN pour deux signaux reçus et les deux CNA pour deux signaux de sortie. On peut donc avoir en même temps une configuration minimum 2×2 soit quatre canaux SISO par carte. Si les ressources du FPGA permettent la réalisation d'autres canaux SISO, alors on pourra même simuler une configuration de plus de quatre canaux SISO dans un seul FPGA. Par contre, le résultat numérique devra être envoyé vers d'autres cartes similaires pour contribuer au calcul des autres signaux reçus. De plus, le calcul avec des signaux réels ne nécessite qu'un seul multiplieur réel à la place des quatre multiplieurs réels formant un multiplieur complexe, ce qui permet de réduire d'une façon significative les ressources et d'augmenter le nombre des canaux SISO simulés.

La solution retenue met alors en œuvre une plate-forme de prototypage constituée de trois cartes identiques « XtremeDSP Development Kit-IV », chacune pouvant être reliée à deux antennes d'émission et deux antennes de réception. Il sera ainsi possible de simuler des canaux MIMO allant jusqu'à la configuration 3×4, c'est-à-dire jusqu'à douze canaux SISO unidirectionnels, comme nous pouvons le voir sur la figure 2.12.

Afin de bien maîtriser chaque bloc des architectures proposées (partie numérique temporelle, partie numérique fréquentielle), nous avons souhaité les tester indépendamment. Pour cela, nous utilisons les outils MATLAB 7.1, Modelsim 6 et ISE 7.1. Ce dernier est développé par la société Xilinx [60] et c'est pourquoi nous l'avons choisi pour travailler avec le composant programmable choisi précédemment. Nous utilisons également des IP (*Intellectual Properties*) fournies par Xilinx. Ces IP sont des macro-cellules préconçues, configurables, générées par le logiciel ISE avec l'outil « CORE Generator ». Avec ces IP, il est possible de simuler l'architecture choisie de façon conviviale en utilisant leur interface graphique pour leur configuration.

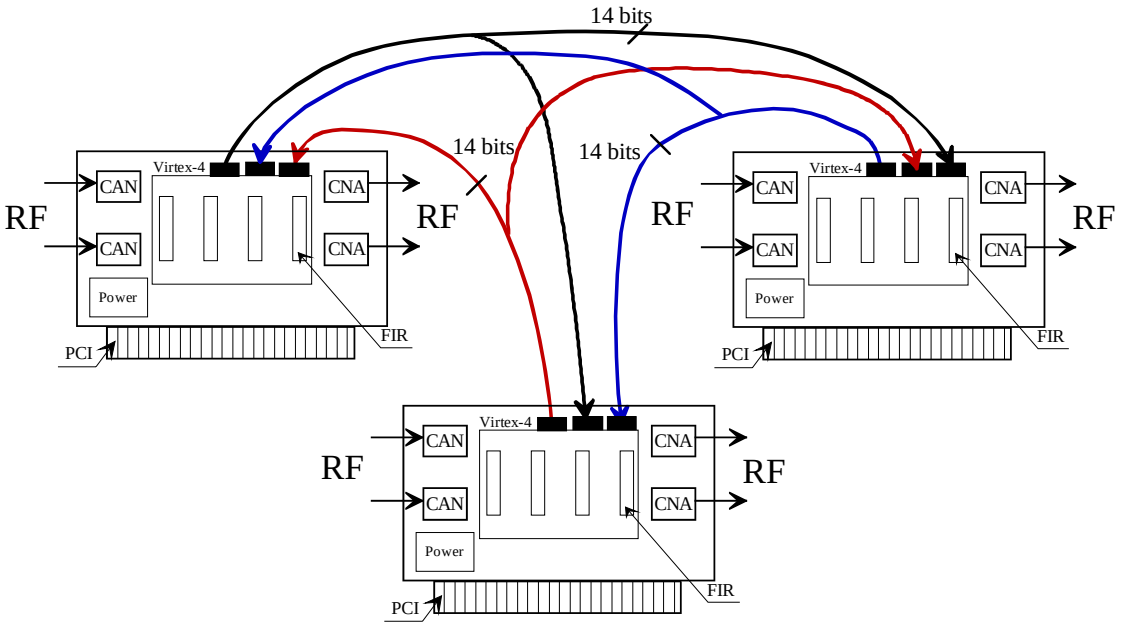

FIG. 2.12 – Connexions des cartes de prototypages pour une configuration de 12 canaux SISO unidirectionnels.

Pour effectuer les tests pratiques, on a besoin de connaître la distribution de l'horloge aux différents périphériques du Virtex-4.

2.2.3 Génération et distribution de l'horloge

Comme nous l'avons vu dans la description de la carte, le composant programmable FPGA Virtex-II XC2V80 est spécialement dédié au traitement de l'horloge. La carte contient quatre sources d'horloge disponibles : l'oscillateur à quartz de 105 MHz, qui pour nos besoins a été remplacé par un quartz à 100 MHz, les horloges programmables, via le logiciel FUSE fourni avec le kit de développement, un emplacement où l'on peut ajouter un quartz supplémentaire et une entrée permettant de connecter une horloge externe.

Selon la documentation de la carte [67], on observe que les deux horloges programmables ainsi que l'oscillateur à quartz optionnel sont directement connectés au Virtex-4 alors que l'oscillateur à quartz de 100 MHz et le connecteur pour l'horloge externe sont connectés au Virtex-II dédié à l'horloge. Cependant, on remarque que les signaux d'horloge, pilotant les convertisseurs, proviennent uniquement du Virtex-II. Pour notre application qui utilise ces convertisseurs, il faudra donc, de toute façon, passer par la programmation du Virtex-II dédié à l'horloge. On peut alors choisir d'utiliser l'oscillateur à quartz de 100 MHz comme source d'horloge à notre disposition, ce qui, non seulement, sera plus précis que de générer une horloge à l'aide d'un générateur de fréquence, mais aussi facilitera la synchronisation entre la fréquence d'échantillonnage utilisée par les convertisseurs et la fréquence d'horloge du bloc numérique.

Comme la fréquence d'échantillonnage requise la plus élevée est de 50 MHz, on peut diviser la fréquence de l'oscillateur à quartz pour obtenir la fréquence d'entrée des échantillons. Pour réaliser la division par deux de la fréquence du signal d'horloge provenant du quartz, on utilise l'IP DCM (*Digital Clock Manager*) fourni par Xilinx et intégré au Virtex-4. Ce composant, spécialement dédié au traitement de l'horloge, permet, entre autre, de diviser ou de multiplier la fréquence d'un signal d'horloge par un nombre prédéfini (à indiquer dans les paramètres génériques du programme VHDL). Dans notre cas, le signal d'horloge est simplement divisé par 2 afin de cadencer les convertisseurs à 50 MHz. Pour le cas de l'UMTS, pour lequel on a choisi une fréquence d'échantillonnage à 40 MHz, il suffit de diviser la fréquence du signal d'horloge provenant du quartz par 2,5 en utilisant l'IP DCM.

Concernant l'entrée des profils de réponses impulsionnelles, pour l'architecture temporelle, ou fréquentielles, pour l'architecture fréquentielle, les profils seront stockés sur le disque dur d'un ordinateur. Selon l'architecture choisie, soit les réponses impulsionnelles seront ensuite chargées en tant que coefficients des filtres FIR, soit les réponses fréquentielles seront multipliées par les échantillons de la transformée de Fourier du signal d'entrée, en sortie du bloc FFT. Cependant, pour des questions de ressources, nous ne pouvons utiliser que les 192 multiplieurs (ou « XtremeDSP ») présents sur le FPGA.

2.3 Conclusion

Cette première partie a donc permis de poser les premiers jalons de la conception de l'architecture du simulateur, à savoir les descriptions de la partie RF, de la partie « Modèles de canal » et de la partie numérique avec le choix de quelques paramètres. Après une étude comparative, afin de réduire la complexité du bloc numérique, les signaux de sortie du simulateur MIMO seront calculés dans le domaine fréquentiel pour un environnement extérieur et dans le domaine temporel pour un environnement intérieur aux bâtiments.

Par ailleurs, le simulateur doit pouvoir simuler le canal de propagation MIMO dans un type d'environnement choisi par l'utilisateur. La partie numérique doit donc être alimentée par des réponses impulsionnelles ou fréquentielles de canal, selon l'environnement choisi, en utilisant un modèle théorique, normalisé ou des résultats de mesure.

Le choix de la plate-forme de test ayant été justifié, nous allons maintenant nous intéresser à la partie simulation théorique et réalisation. Chaque domaine, temporel et fréquentiel, fait l'objet d'un chapitre contenant les descriptions et les simulations des architectures numériques considérées ainsi que l'implémentation sur cartes électroniques et les tests réels avec l'exploitation de leurs résultats.

Chapitre 3

Approche fréquentielle

Sommaire

3.1 Description des différents blocs de la chaîne fréquentielle . . 52

3.1.1 Etude de la transformée de Fourier rapide, directe et inverse . . . 52

3.1.2 Etude du bloc Mémoire 59

3.1.3 Etude du multiplieur complexe 61

3.1.4 Etude de la troncature 62

3.2 Mise en œuvre de la chaîne dans le domaine fréquentiel . . . 64

3.2.1 Justification du signal de test 65

3.2.2 Etude de la chaîne fréquentielle 67

3.2.3 Résultats et précision de calcul 75

3.3 Conclusion 86

Comme nous l'avons justifié dans le chapitre précédent, deux approches sont proposées pour réaliser la partie numérique du simulateur. Rappelons que, dans un environnement « outdoor », les réponses impulsionnelles peuvent avoir une durée de l'ordre de la dizaine de microsecondes et qu'elles doivent être plus souvent rafraîchies car l'environnement est susceptible d'évoluer plus rapidement. La contrainte de latence est donc plus faible par rapport au milieu indoor où les réponses impulsionnelles peuvent durer plusieurs centaines de nanosecondes. Les opérations de convolution étant plus coûteuses en terme de ressources, on choisit alors une architecture qui effectue les calculs dans le domaine fréquentiel. L'approche fréquentielle de la partie numérique du simulateur matériel de propagation est détaillée dans ce chapitre.

Dans une première partie, et avant de procéder à la simulation, puis à l'implémentation de cette architecture, chaque bloc qui la compose sera étudié d'un point de vue traitement du signal. Dans une seconde partie, nous verrons la chaîne fréquentielle en numérique dans sa globalité pour la tester, étudier sa précision, et commenter les résultats.

3.1 Description des différents blocs de la chaîne fréquentielle

Le canal MIMO étant composé de plusieurs canaux SISO, nous détaillerons l'approche fréquentielle de la partie numérique du simulateur matériel pour un seul canal SISO.

Dans le domaine fréquentiel, l'architecture proposée pour la partie numérique du simulateur matériel de propagation pour un canal SISO peut être représentée par la figure 3.1 qui montre l'évolution des signaux et de leur transformée de Fourier. Or, nous traiterons la chaîne fréquentielle en numérique, avec des signaux échantillonnés et quantifiés sur un certain nombre de bits que nous définirons. C'est pourquoi la chaîne fréquentielle peut également être décrite par la figure 3.2 qui utilise la représentation numérique de ces signaux.

Cette architecture met en œuvre un module effectuant la transformée de Fourier rapide directe (FFT) [68, 69] qui peut être configuré pour effectuer une transformée de Fourier rapide inverse (IFFT). Le multiplieur complexe, le bloc Mémoire et le module de troncature seront également détaillés.

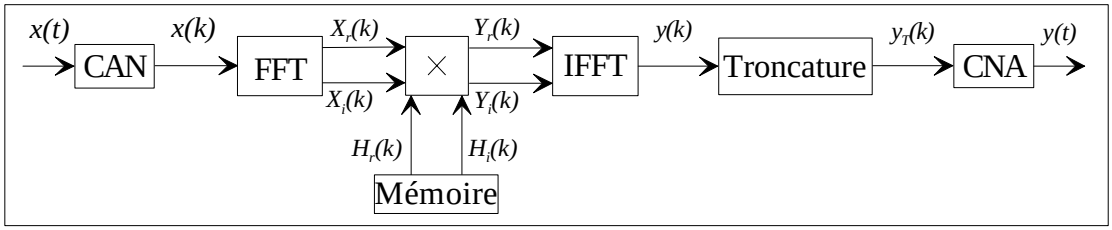

FIG. 3.1 – Diagramme de la chaîne dans le domaine fréquentiel pour un canal SISO unidirectionnel.

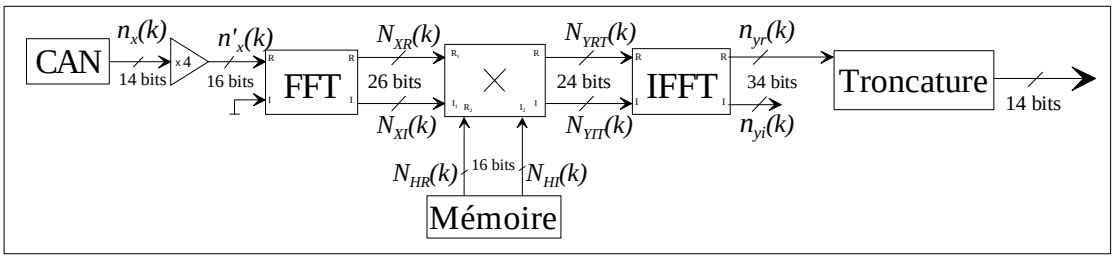

FIG. 3.2 – Diagramme de la chaîne dans le domaine fréquentiel en numérique pour un canal SISO unidirectionnel.

3.1.1 Etude de la transformée de Fourier rapide, directe et inverse

Afin de simplifier la description des signaux, nous nous plaçons dans le cas simple d'un système linéaire invariant dans le temps. Plus tard, nous pourrons appliquer cette des-

cription pour des systèmes évoluant dans le temps et ayant une réponse impulsionnelle de type $h(t, \tau)$.

3.1.1.1 Description de l'algorithme de la transformée de Fourier discrète

La transformée de Fourier est un outil mathématique largement utilisé et que l'on souhaite détailler ici pour montrer le lien entre les résultats d'une transformée de Fourier et les résultats de l'algorithme de la FFT. En effet, nous utilisons la transformée de Fourier pour un signal temporel continu, qui sera échantillonné et dont les échantillons seront traités tout au long de la chaîne fréquentielle. On devra être capable de retrouver le signal continu en sortie après les différentes étapes de cette chaîne.

L'expression générale de la transformée de Fourier d'un signal temporel continu $x(t)$ est donnée par :

$$X(f) = TF\{x(t)\} = \int_{-\infty}^{+\infty} x(t).e^{-j2\pi ft}dt. \tag{3.1}$$

Dans notre cas, on applique la transformée de Fourier pour un signal réel à durée et à bande finies. S'il existe $W_t > 0$ tel que $x(t) = 0$, $\forall\ t \notin [0,\ W_t]$, alors :

$$X(f) = TF\{x(t)\} = \int_{0}^{W_t} x(t).e^{-j2\pi ft}dt. \tag{3.2}$$

Si on note la période d'échantillonnage $\Delta t = \frac{1}{f_s}$ avec f_s choisie comme nous l'avons vu dans le chapitre précédent, alors on peut diviser l'intervalle $[0,\ W_t]$ en N sous-intervalles de longueur Δt, donc $\Delta t = \frac{W_t}{N}$. Mais comme, en général, $\frac{W_t}{\Delta t}$ n'est pas un entier, on choisit N comme étant le plus petit entier vérifiant $N\Delta t \geq W_t$, ce qui revient à calculer N avec la relation $N = \lceil \frac{W_t}{\Delta t} \rceil$. On obtient ainsi les points $t_k = k\Delta t$ avec $k = \overline{0, N}$. On a alors :

$$X(f) = TF\{x(t)\} = \sum_{k=1}^{N} \int_{(k-1)\Delta t}^{k\Delta t} x(t).e^{-j2\pi ft}dt. \tag{3.3}$$

De plus, si N est assez grand, Δt est assez petit pour considérer que $x(t)$, $\sin(2\pi ft)$ et $\cos(2\pi ft)$ ont de faibles variations sur un sous-intervalle de longueur Δt. On a alors :

$$X(f) \simeq \sum_{k=0}^{N-1} \Delta t.x(t_k).e^{-j2\pi ft_k} = \Delta t. \sum_{k=0}^{N-1} x(t_k).e^{-j2\pi ft_k}. \tag{3.4}$$

On peut ainsi construire le vecteur $\boldsymbol{x}$ constitué des échantillons $x_k = x(t_k)$ tel que :

$$\boldsymbol{x} = [x(0),\ x(\Delta t),\ x(2\Delta t),\ \ldots,\ x((N-1)\Delta t)]. \tag{3.5}$$

Par ailleurs, le signal $X(f)$ ne sera calculé par la suite que pour certaines valeurs de fréquence, notées $f_l = l\Delta f$, avec $l = 0,\ 1,\ \ldots,\ N-1$ et $\Delta f > 0$. Par conséquent, le signal $X(f)$ sera calculé sur une *bande finie*.

$$X(f_l) \simeq \sum_{k=0}^{N-1} \Delta t.x(t_k).e^{-j2\pi f_l t_k} \text{ avec } l = \overline{0,\ N-1} \tag{3.6}$$

$$X(f_l) \simeq \Delta t \sum_{k=0}^{N-1} x(k\Delta t).e^{-j2\pi lk\Delta f\Delta t} \text{ avec } l = \overline{0,\ N-1} \tag{3.7}$$

Avec les valeurs $X(f_l) = X(l\Delta f)$, on peut construire le vecteur $\boldsymbol{X}$ tel que :

$$\boldsymbol{X} = [X(0),\ X(\Delta f),\ X(2\Delta f),\ \ldots,\ X((N-1)\Delta f)] \tag{3.8}$$

On obtient donc :

$$\boldsymbol{X} = \Delta t.\boldsymbol{x}.\boldsymbol{W} \tag{3.9}$$

où $\boldsymbol{W}$ est une matrice carrée $N \times N$ formée par les éléments w^{lk} tels que :

$$w^{lk} = e^{-j.2\pi lk\Delta f\Delta t}. \tag{3.10}$$

Abordons le cas particulier de la transformée de Fourier rapide, ou FFT en anglais. Si l'on accepte $N\Delta t\Delta f = 1$ et $N = 2^n$, avec $n \in \mathbb{N}^*$ (conditions propres à la FFT), alors

$$X(f_l) \simeq \Delta t \sum_{k=0}^{N-1} x(k\Delta t).e^{-j\frac{2\pi lk}{N}}. \tag{3.11}$$

En général, l'algorithme de la FFT considère $\Delta t = 1$ (valeur normalisée) [70, 71]. Le calcul de la FFT donne donc :

$$\sum_{k=0}^{N-1} x_k.e^{-j\frac{2\pi lk}{N}} \quad \text{avec } l = \overline{0,\ N-1}. \tag{3.12}$$

On peut écrire aussi :

$$\sum_{k=0}^{N-1} x_k.w^{lk} \quad \text{avec } w = e^{-j\frac{2\pi}{N}} \in \mathbb{C} \text{ et } l = \overline{0,\ N-1}. \tag{3.13}$$

L'opérateur FFT effectue donc une transformée de Fourier au facteur Δt près. Si on appelle $\boldsymbol{X'}$ le résultat de l'opérateur FFT, la relation entre les vecteurs lignes $\boldsymbol{X'}$ et $\boldsymbol{x}$ est

alors donnée par :

$$\boldsymbol{X'} = \boldsymbol{x}.\boldsymbol{W} \tag{3.14}$$

avec $\boldsymbol{W}$ la matrice carrée $N \times N$ exprimée ci-après et dont les éléments w^{lk} sont calculés pour $l,\ k = \overline{0,\ N-1}$. De plus, le fait d'avoir $N = 2^n$, avec $n \in \mathbb{N}^*$, permet d'avoir des éléments w^{lk} qui se répètent dans la matrice.

$$\boldsymbol{W} = \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & w & w^2 & w^3 & \cdots & w^{N-1} \\ 1 & w^2 & w^4 & w^6 & \cdots & w^{2(N-1)} \\ 1 & w^3 & w^6 & \ddots & & \vdots \\ \vdots & \vdots & \vdots & & \ddots & \vdots \\ 1 & w^{N-1} & w^{2(N-1)} & \cdots & \cdots & w^{(N-1)^2} \end{bmatrix} \tag{3.15}$$

Si le vecteur $\boldsymbol{x} \in \mathbb{R}$, on constate que les éléments de sortie de la transformée de Fourier présentent une symétrie hermitienne par rapport à l'indice $\frac{N}{2}$ (correspondant à la fréquence de Nyquist). Cette symétrie est décrite par $X_{\frac{N}{2}-k} = X^*_{\frac{N}{2}+k}$ pour $0 \leq\ k \leq\ \frac{N}{2}-1$.

Le produit par Δt doit ensuite être effectué par l'utilisateur de la FFT pour retrouver, d'une façon approximative, le résultat théorique calculé par la formule intégrale propre à la transformée de Fourier.

Pour déterminer le lien entre la transformée de Fourier inverse et l'algorithme de la IFFT (*Inverse Fast Fourier Transform*), on part de la formule intégrale similaire ci-dessous :

$$x(t) = TF^{-1}\{X(f)\} = \int_{-\infty}^{+\infty} X(f).e^{j2\pi ft}df. \tag{3.16}$$

Après un cheminement équivalent, on obtient la relation de la IFFT :

$$\boldsymbol{x} = \Delta f.\boldsymbol{X}.\boldsymbol{W_I} \tag{3.17}$$

avec la matrice de la transformée de Fourier rapide inverse $\boldsymbol{W_I}$.

En général, l'algorithme de la IFFT considère $\Delta f = 1$ (valeur normalisée). Le calcul de la IFFT donne donc $\boldsymbol{X}.\boldsymbol{W_I}$. On obtient donc le résultat de la transformée de Fourier inverse au facteur Δf près.

Connaissant la relation 3.9, on peut écrire :

$$\boldsymbol{x} = \Delta f.\Delta t.\boldsymbol{x}.\boldsymbol{W}.\boldsymbol{W_I} \tag{3.18}$$

$$= \frac{1}{N}.\boldsymbol{x}.\boldsymbol{W}.\boldsymbol{W_I} \tag{3.19}$$

Ainsi, en effectuant le produit des deux matrices $\boldsymbol{W}$ et $\boldsymbol{W_I}$, on doit obtenir la matrice identité $\boldsymbol{I}$, au facteur $\frac{1}{N}$ près :

$$\frac{1}{N}.\boldsymbol{W_I}.\boldsymbol{W} = \boldsymbol{I}. \tag{3.20}$$

Par calcul direct, il est possible de vérifier l'égalité :

$$\frac{1}{N}.\boldsymbol{W^*}.\boldsymbol{W} = \boldsymbol{I} \tag{3.21}$$

et on en déduit :

$$\boldsymbol{W_I} = \boldsymbol{W^*}. \tag{3.22}$$

La matrice $\boldsymbol{W_I}$ de l'algorithme de la IFFT est donc la valeur complexe conjuguée de la matrice $\boldsymbol{W}$ de l'algorithme de la FFT.

3.1.1.2 Etude de la FFT de Xilinx

La chaîne fréquentielle complète sera simulée à partir des paramètres UMTS du tableau 2.2. Ainsi, dans le pire des cas de l'environnement outdoor du tableau 2.2, la longueur N des blocs FFT et IFFT sera de $2^9 = 512$ points. D'un point de vue matériel, ces blocs seront réalisés avec le même module, c'est-à-dire une IP (définie au chapitre précédent) fournie par le « CORE Generator » de Xilinx [72]. Cette IP permet de calculer des transformées de Fourier discrètes, directes ou inverses, de longueur N pouvant aller de 2^3 à 2^{16} points. Pour effectuer une FFT directe ou une FFT inverse, le seul paramètre qui différencie l'utilisation de cette IP est respectivement la mise à « 1 » ou à « 0 » du signal « fwd_inv ». Après avoir détaillé le calcul de la matrice de la FFT dans le cas général à la section précédente, nous nous intéressons à cette IP FFT fournie par Xilinx.

Afin d'alimenter cette IP, les données d'entrée peuvent être fournies sous la forme d'un vecteur de valeurs complexes représentées par des nombres binaires de 8 à 24 bits en complément à deux. De même, les éléments de la matrice, appelés par Xilinx « facteurs de phase », peuvent être représentés sur 8 à 24 bits. Trois types de calcul de précision sont disponibles mais nous n'utiliserons que celui appelé pleine précision « unscaled » qui garde tous les bits tout au long du calcul afin de garder la meilleure précision. Plusieurs paramètres sont configurables dont la taille du bloc FFT, c'est-à-dire le nombre de points, le choix de la transformée directe ou inverse, le type de précision. Une interface graphique permet de paramétrer le bloc FFT/IFFT afin de choisir le nombre de points, le nombre de bits du signal d'entrée, le type de calcul avec arrondi ou troncature en gardant la pleine échelle ou pas, etc.

Le bloc FFT/IFFT de Xilinx accepte des valeurs des échantillons complexes du signal d'entrée représentées uniquement sur 8, 12, 16, 20 ou 24 bits. Le bloc a une entrée pour la partie réelle et une entrée pour la partie imaginaire des échantillons. Compte tenu des

convertisseurs utilisés, les échantillons du signal d'entrée sont représentés sur 14 bits et, afin de garder toute la précision des convertisseurs, deux bits de poids faible et de valeur nulle sont rajoutés aux 14 bits des échantillons du signal d'entrée pour que ces derniers puissent être acceptés sur l'entrée 16 bits du bloc FFT/IFFT de Xilinx. Cette opération revient à multiplier par 4 la valeur des échantillons d'entrée.

Dans le bloc FFT/IFFT de Xilinx, les éléments de la matrice utilisée ($\boldsymbol{W}$ ou $\boldsymbol{W_I}$) sont quantifiés sur un certain nombre de bits. A la place de la matrice $\boldsymbol{W}$ indiquée par la relation (3.15), un bloc FFT/IFFT de Xilinx utilise une « matrice quantifiée » $\boldsymbol{W_q}$. Afin de mieux connaître cette matrice quantifiée $\boldsymbol{W_q}$ utilisée pour le calcul de la FFT de Xilinx, un signal d'entrée $\boldsymbol{x} = [x(0),\ x(1),\ \cdots,\ x(N-1)]$, où $N = 512$, est appliqué en entrée de la FFT de Xilinx tel que $x(1) = 01000000000000 = 2^{12}$, avant de rajouter deux bits les moins significatifs et de valeur nulle de façon à avoir une entrée de la FFT sur 16 bits telle que $x'(1) = 2^{14}$. Le reste des valeurs du vecteur ligne $\boldsymbol{x'}$ vaut 0. Le vecteur ligne $\boldsymbol{X}$ obtenu en sortie est alors égal à la deuxième ligne de la matrice $\boldsymbol{W_q}$ multipliée par 2^{14}, comme illustré ci-après.

$$\underbrace{\begin{bmatrix} 0 & 2^{14} & 0 & \cdots & 0 \end{bmatrix}}_{\boldsymbol{x'}} \times \underbrace{\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & w_q & w_q^2 & \cdots & w_q^{N-1} \\ 1 & w_q^2 & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ 1 & w_q^{N-1} & \cdots & \cdots & w_q^{(N-1)^2} \end{bmatrix}}_{\boldsymbol{W_q}} = \underbrace{2^{14}.\begin{bmatrix} 1 & w_q & w_q^2 & \cdots & w_q^{N-1} \end{bmatrix}}_{\boldsymbol{X}} \quad (3.23)$$

La connaissance des $N-1$ valeurs w_q^k $(1 \leq k \leq N-1)$ permet donc de connaître complètement la matrice quantifiée $\boldsymbol{W_q}$. Ainsi, nous pouvons en déduire la précision des éléments quantifiés de la matrice $\boldsymbol{W_q}$ utilisée par le module FFT de Xilinx par rapport à la matrice $\boldsymbol{W}$ « théorique », telle qu'elle est utilisée par exemple par MATLAB. Après plusieurs essais, le meilleur compromis entre précision, durée de calcul et ressources utilisées est obtenu pour des éléments de la matrice codés sur 16 bits également. Nous garderons donc ce choix pour paramétrer l'interface graphique de l'IP FFT/IFFT de Xilinx.

Comme nous le verrons dans la section de mise en œuvre de la chaîne fréquentielle, les résultats en sortie du bloc FFT sont des nombres *proportionnels* aux valeurs des échantillons quantifiés. Comme nous l'avons précisé précédemment par l'expression (3.9), si ces valeurs de sortie ne servaient pas par la suite pour d'autres calculs, elles devraient être multipliées non seulement par la période d'échantillonnage du signal d'entrée Δt afin de retrouver le signal dit « théorique », comme pour la matrice de la FFT de MATLAB, mais aussi par un facteur multiplicatif car la FFT de Xilinx n'accepte en entrée que des

nombres entiers écrits en complément à 2. Dans notre cas, ces valeurs de sortie entrent dans un multiplieur complexe pour continuer la chaîne fréquentielle. Ainsi, les multiplications des échantillons de sortie par les facteurs d'échelle cités, permettant de retrouver le signal théorique, ne se feront qu'à la fin de la chaîne fréquentielle complète, de façon à comparer le signal de sortie avec le signal théorique, comme nous le verrons plus tard.

D'autre part, différentes configurations ont été considérées en simulation pour connaître le taux d'occupation du composant programmable pour le bloc FFT et le bloc IFFT fournis par Xilinx. Le taux d'occupation du composant peut être connu après avoir effectué trois opérations principales à partir du programme écrit en VHDL : la synthèse, le mapping et le placement-routage. On appelle synthèse du programme VHDL la compilation d'une description fonctionnelle d'un circuit pour générer un schéma en portes logiques et en bascules. Puis, l'opération de mapping décrit le regroupement de ces portes logiques sous forme de LUT (*Look-Up Table*), sorte de table de correspondance en mémoire statique, qui permet d'associer des valeurs pré-calculées. Dans notre cas, les LUT peuvent, par exemple, contenir les $N-1$ valeurs pré-calculées des éléments de la matrice $\boldsymbol{W_q}$ de façon à accélérer le calcul de la FFT. Enfin, après le placement des composants, le routage concrétise l'arrangement des connexions entre les ressources logiques et les entrées-sorties matérielles du composant. A l'issue de ces étapes, réalisées avec l'outil ISE de Xilinx, on obtient le taux d'occupation du composant programmable, ainsi que la fréquence maximale de fonctionnement de l'IP FFT de 172 MHz.

Dans ce tableau, les termes, « XtremeDSP » et blocs RAM ont été expliqués dans le chapitre précédent, le terme LUT est évoqué ci-dessus, quant aux « slices » ce sont des cellules qui regroupent chacune deux LUT et deux bascules de type D, pour la famille Virtex de Xilinx.

Nombre de « Slices »	2569 sur 15360	16 %
Nombre de LUT	3962 sur 30720	12 %
Nombre de blocs RAM	13 sur 192	6 %
Nombre de bascules	4750 sur 30720	15 %
Nombre de « XtremeDSP »	22 sur 192	11 %

TAB. 3.1 – Taux d'occupation du composant pour un seul module FFT 512.

Ce taux d'occupation est utile pour connaître les ressources utilisées par le composant programmable et avec quel pourcentage, afin de l'optimiser au maximum. Dans le tableau 3.1, on ne montre que l'occupation du module FFT 512 pris à part et on peut remarquer que ce sont les circuits logiques qui occupent la plupart du composant suivis par les multiplieurs, qui y ont également une part importante.

3.1.2 Etude du bloc Mémoire

Dans le chapitre précédent, nous avons vu comment les profils de réponses fréquentielles, nécessaires à la simulation du canal de propagation, pouvaient être obtenus. Nous allons maintenant détailler le bloc Mémoire de la figure 3.1 qui sert à stocker ces profils avant de les utiliser. Cependant, il faut d'abord savoir combien de bits représentent les profils et comment charger ce bloc Mémoire.

3.1.2.1 Volume de données et choix du bus de transfert

Les parties réelles et imaginaires des échantillons des réponses fréquentielles utilisées sont codées sur 16 bits de façon à avoir une précision équivalente à la précision des valeurs d'entrée du bloc FFT.

Dans notre cas, une réponse fréquentielle complexe est représentée par 512 échantillons ou mots de 32 bits, dont 16 bits pour la partie réelle et 16 bits pour la partie imaginaire. De plus, on rajoute un mot de 32 bits à chaque profil de façon à indiquer le bloc de mémoire de destination du profil, car nous verrons plus loin que plusieurs blocs de mémoire sont utilisés. On doit transférer 513 mots de 32 bits par profil. En prenant, par exemple, un canal MIMO 2×2 unidirectionnel, comme celui de la figure 1.10, on a 4 canaux SISO unidirectionnels à simuler, donc on a besoin de 4 profils de réponses fréquentielles. On doit alors transmettre (513 × 4) mots de 32 bits, soit 2052 mots de 32 bits donc environ 65 kbits à transmettre ou 8208 octets.

Les profils contenant ces échantillons de 32 bits sont stockés dans un fichier texte sur le disque dur d'un ordinateur. Ce fichier est ensuite lu pour charger le bloc Mémoire qui alimentera le simulateur matériel. La lecture du fichier peut se faire soit à partir de l'interface USB 1.1, soit à partir de l'interface PCI, toutes les deux disponibles sur la carte de prototypage utilisée. Comme nous avons vu au chapitre 2 que, dans le pire cas UMTS rural, la fréquence de rafraîchissement $f_{raf_{UMTS}}$ est de 2,37 kHz, la période de rafraîchissement est donc d'environ 422 μs pendant laquelle on doit rafraîchir l'ensemble des 4 profils (pour notre exemple), c'est-à-dire 8208 octets, ce qui fait un débit de :

$$\frac{8208}{422.10^{-6}} = 18,5 \text{ Mo/s}. \tag{3.24}$$

Le bus USB ne permettrait pas de respecter ce débit. Le bus PCI a donc été choisi pour le chargement des profils de réponses fréquentielles, son débit pouvant atteindre 30 Mo/s en écriture. De plus, le bus PCI est un bus de 32 bits, donc à chaque coup d'horloge, on transmet un échantillon complexe de la réponse fréquentielle. D'ailleurs, comme un canal SISO correspond ici à un profil de (512 × 32) bits, soit 2048 octets, le débit de 30 Mo/s permettrait, pendant le temps de rafraîchissement de 422 μs, le chargement de 6 canaux SISO.

Nous utiliserons également le bus PCI, c'est-à-dire le bus le plus rapide, pour les applications WLAN (indoor) pour le rafraîchissement des réponses impulsionnelles.

3.1.2.2 Chargement du bloc Mémoire

Le schéma-bloc de la figure 3.3 montre le lien entre le PC qui contient le fichier des profils de réponses fréquentielles à rafraîchir et la carte de développpement XtremeDSP de Nallatech contenant le Virtex-4 où est implémentée la Partie numérique que l'on peut voir sur le schéma fonctionnel de la figure 2.1.

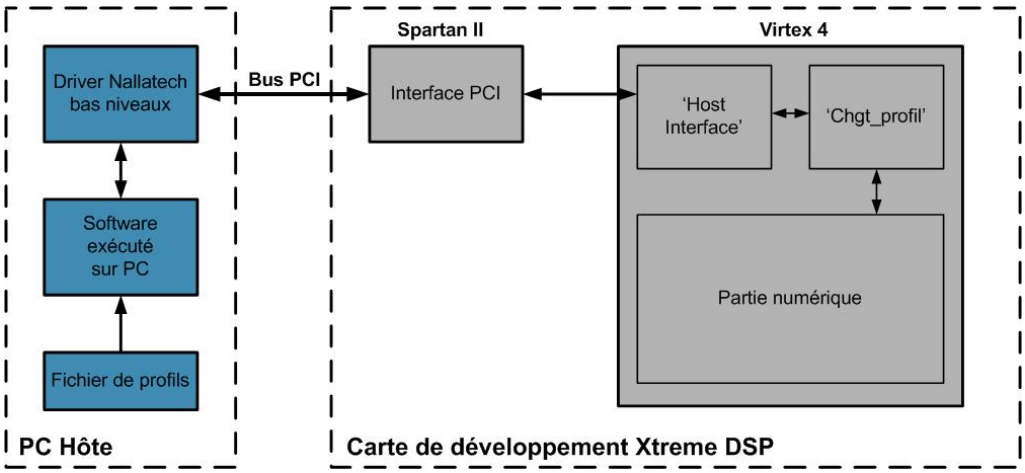

FIG. 3.3 – Lien entre le PC et la carte Nallatech.

Comme nous l'avons vu sur la gauche du schéma de la figure 2.11, le composant programmable Spartan-II est spécialement dédié au traitement des interfaces USB et PCI. Il a été programmé par Nallatech pour récupérer les données présentes sur les bus et pour les rediriger vers le Virtex-4 où sera implémentée notre architecture numérique.

Nallatech fournit également une IP nommée « Host Interface » que l'on programme pour faire le lien entre les deux FPGA. Elle permet de lire les données provenant du bus PCI choisi précédemment et de les stocker dans la FIFO de l'IP « Host Interface ». Puis, le module appelé « Chgt_profil », que nous avons développé, lit les valeurs des échantillons dans la FIFO et les distribue dans les deux blocs « RAM » ou bloc de mémoire double-port, appelés « DP_RAM_A » et « DP_RAM_B », comme nous pouvons le voir sur la figure 3.4.

Cette figure détaille le lien entre l'IP « Host Interface » et le chargement des profils dans le module appelé « BOX-RAM », constitué des deux blocs « RAM » cités et présents dans le composant programmable Virtex-4. Ce module « BOX-RAM » constitue le bloc Mémoire de l'architecture numérique dans le domaine fréquentiel.

Les deux blocs « RAM » permettent de lire un profil tout en en chargeant un autre. On remarque, en effet, le signal 'profil_sel' qui contrôle, d'une part, le démultiplexeur et, d'autre part, après avoir été complémenté, le multiplexeur. Ainsi, lorsque le multiplexeur sélectionne un bloc « RAM » pour lire les 512 valeurs d'un profil de réponses fréquentielles complexes, le démultiplexeur sélectionne l'autre bloc « RAM » pour y écrire les 512 valeurs du profil suivant. Ainsi, pendant qu'un profil est utilisé, le profil suivant est chargé et sera utilisé après le temps de rafraîchissement des profils définis au chapitre précédent. Le signal 'profil_sel' est donc périodique, de période deux fois le temps de rafraîchissement avec un rapport cyclique de 50 %.

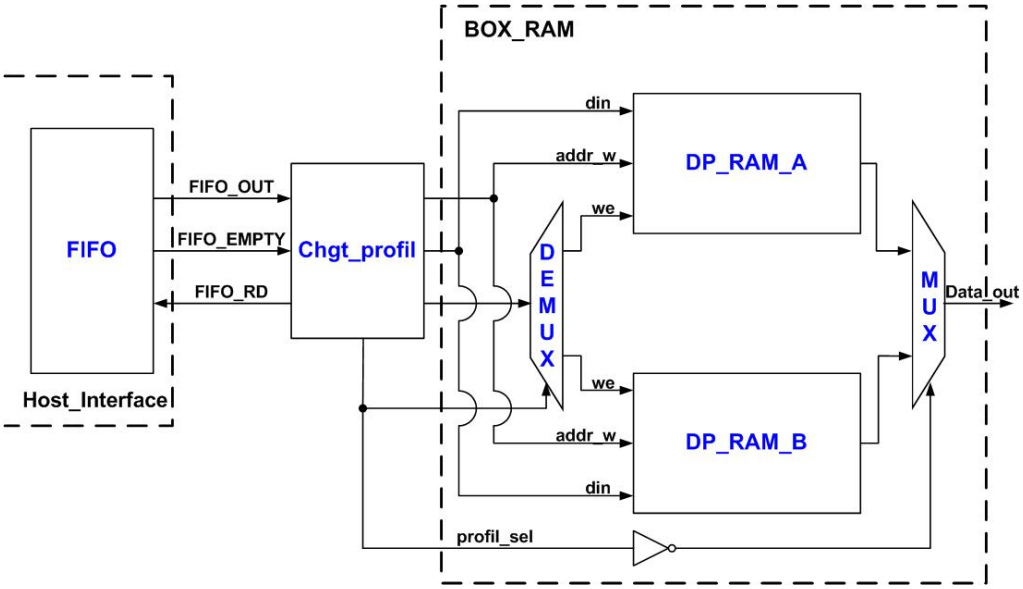

FIG. 3.4 – Chargement des profils de réponses fréquentielles pour un canal SISO

Il s'agit d'un fonctionnement en double buffer ou « ping-pong ». La sortie du multiplexeur 'Data_out' est un bus de 32 bits dont les 16 MSB sont dirigés vers l'entrée des échantillons réels et les 16 LSB vers l'entrée des échantillons imaginaires du multiplieur complexe.

3.1.3 Etude du multiplieur complexe

La multiplication complexe de la sortie de la FFT du signal d'entrée avec les profils de réponses fréquentielles du canal est réalisée par l'IP « Complex Multiplier » de Xilinx [73], qu'une interface graphique permet de paramétrer.

L'opération utilise les « XtremeDSP » présents sur le FPGA. Comme nous l'avons vu dans le chapitre précédent, chaque « XtremeDSP » contient un multiplieur 18 bits par 18 bits, un additionneur de 48 bits et des registres. Tous les opérandes, ainsi que les résultats, sont représentés au format complément à deux. Les longueurs des opérandes

et des résultats sont paramétrables. La longueur, pour les opérandes, va de 4 à 64 bits et peut aller jusqu'à 129 bits pour le produit. Dans le cas où la longueur du produit est choisie plus petite que la longueur maximale possible, le multiplieur peut réaliser soit l'arrondi soit la troncature des résultats, à la demande de l'utilisateur, par le biais de l'interface graphique. En outre, deux architectures de ce multiplieur sont proposées dans [73], selon que l'on souhaite optimiser la vitesse des opérations ou le nombre de multiplieurs réels utilisés à l'intérieur du multiplieur complexe. Par exemple, si l'on a deux nombres complexes $a = a_r + ja_i$ et $b = b_r + jb_i$ à multiplier, on obtient le produit p tel que :

$$p = ab = p_r + jp_i \tag{3.25}$$

L'architecture habituelle utilise quatre multiplieurs réels, deux additionneurs et un complément à deux, tels que :

$$p_r = a_r b_r - a_i b_i \quad \text{et} \quad p_i = a_r b_i + a_i b_r, \tag{3.26}$$

or, si on optimise le nombre de multiplieurs réels, on peut écrire p_r et p_i différemment :

$$p_r = a_r(b_r + b_i) - (a_r + a_i)b_i \quad \text{et} \quad p_i = a_r(b_r + b_i) + (a_i - a_r)b_r, \tag{3.27}$$

ce qui permet de n'utiliser que trois multiplieurs réels.

En effectuant les opérations de synthèse, de mapping et de placement-routage, on peut remarquer que si l'on configure le multiplieur complexe pour qu'il optimise la vitesse, la fréquence d'horloge est alors plus élevée. En revanche, si on choisit d'optimiser le nombre de multiplieurs « XtremeDSP », alors l'architecture qui utilise le moins de multiplieurs est préférée, avec un peu plus de ressources logiques utilisées. Dans les deux cas, la latence est nulle car les architectures de ce multiplieur complexe sont réalisées de façon combinatoire.

3.1.4 Etude de la troncature

Les valeurs calculées en sorties du bloc IFFT sont des valeurs $n_y(k)$ proportionnelles à $y(k)$ fournies sur $M_y = 34$ bits. Le bloc Troncature, situé après le bloc IFFT de Xilinx, est nécessaire afin de réduire le nombre de bits des échantillons de sortie du bloc IFFT sur $n_{CNA} = 14$ bits pour que ces échantillons puissent être acceptés par le CNA, tout en gardant la meilleure précision possible. Contrairement aux blocs présentés précédemment, ce bloc a été programmé par nos soins.

La solution immédiate la plus simple consiste à garder les 14 bits les plus significatifs, mais, pour des valeurs faibles des sorties du bloc IFFT sur 34 bits, ne garder que les bits de poids forts peut engendrer des valeurs nulles pour l'entrée des convertisseurs alors qu'elles

n'étaient pas nulles en sortie du bloc IFFT. L'exemple de la figure 3.5 montre un mot de 34 bits tronqué sur 14 bits en rouge.

Dans ce cas, la sortie du bloc numérique, sur 34 bits par exemple, est égale, en valeur décimale à 2726102. Après la suppression des bits de poids faible, nous obtenons le mot suivant : '00000000000010' qui est égal à 2, en décimal. Sachant que supprimer les n_{LSB} bits de poids faible d'un mot revient à prendre la partie entière de la division de ce mot par $2^{n_{LSB}}$, la sortie du CNA doit être multipliée par $2^{n_{LSB}}$ soit ici $2^{20} = 1048576$. Ainsi, après la multiplication de la sortie nous obtenons : $2 \times 1048576 = 2097152$ au lieu de 2726102, ce qui représente presque 25 % d'erreur.

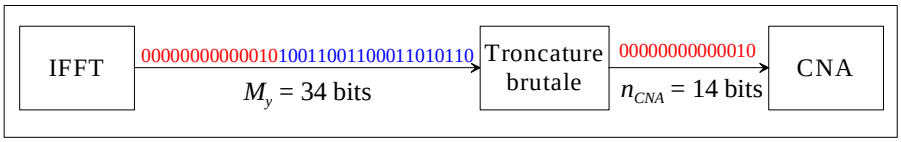

FIG. 3.5 – Exemple de troncature.

Après avoir remarqué que cette troncature « brutale » des valeurs des échantillons de sorties dégradait les valeurs des résultats, une meilleure façon de tronquer ces valeurs a été abordée.

A la place d'une simple troncature brutale, qui garde les 14 premiers bits en commençant avec le MSB, on a considéré une troncature qui utilise une fenêtre glissante sur 14 bits. Les notations utilisées par la suite sont les suivantes : T indique la troncature que l'on qualifiera de « brutale » sur $n_{CNA} = 14$ bits (valeur imposée par le CNA) et TG représente la troncature avec une fenêtre glissante sur $n_{CNA} = 14$ bits.

La troncature glissante consiste à scruter les bits de poids forts identiques, en commençant par le MSB, jusqu'à ce qu'on détecte une différence de valeur entre deux bits adjacents. Le dernier bit identique au MSB devient le nouveau bit de signe et est conservé, ainsi que les 13 bits suivants. Les autres bits sont éliminés. Cette troncature est illustrée à la figure 3.6.

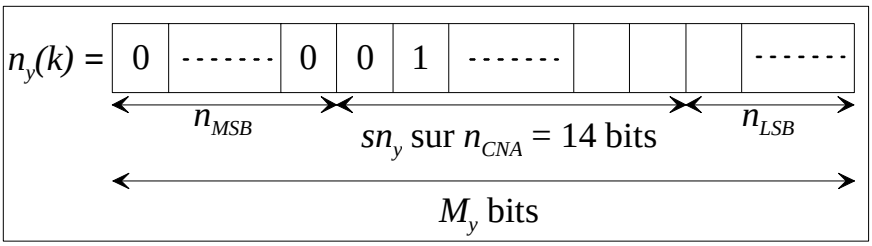

FIG. 3.6 – Schéma de la troncature glissante.

Les bits qui suivent ces 14 bits retenus sont alors les bits les moins significatifs et sont éliminés. Le glissement de la fenêtre des 14 bits retenus impose une correction de l'amplitude finale par la multiplication avec le facteur d'échelle $2^{n_{LSB}}$, où n_{LSB} est le nombre de bits les moins significatifs éliminés par cette troncature glissante. Ainsi, $sn_y(k)$ étant la valeur dénommée « short », c'est-à-dire la valeur calculée avec les 14 bits de la fenêtre glissante, on obtient :

$$n_{y_{TG}}(k) = 2^{n_{LSB}} \times sn_y(k). \tag{3.28}$$

En appliquant la troncature glissante à l'exemple précédent, nous observons en rouge, sur la figure 3.7, les bits à conserver. Nous obtenons ainsi, après troncature, le mot : '01010011001100' qui correspond à la valeur 5324. En sachant que 9 LSB ont été supprimés, il faut multiplier la sortie du CNA par : $2^9 = 512$. Nous obtenons finalement en sortie : $5324 \times 512 = 2725888$ au lieu de 2726102, ce qui représente une erreur relative inférieure à 0,01 %.

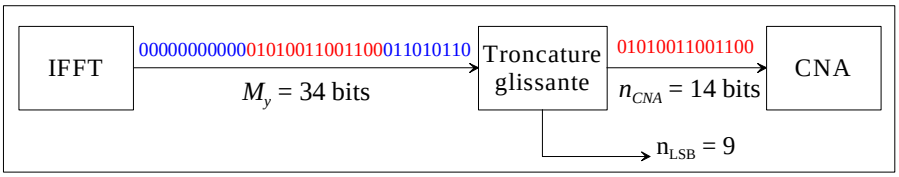

FIG. 3.7 – Exemple de troncature glissante.

Pour les nombres très faibles, il est possible de glisser la fenêtre de 14 bits jusqu'à l'extrémité droite, sans détecter de changement de signe. Dans ce cas, $n_{LSB} = 0$ et les 14 bits retenus sont les 14 LSB du nombre n_y initial.

Cet algorithme de troncature est mis en œuvre par un programme VHDL, le nombre n_{LSB} sortant sur un port d'entrée/sortie pour permettre la multiplication de la tension en sortie du CNA par le facteur d'échelle $2^{n_{LSB}}$. Cette multiplication permettra de corriger l'amplitude du signal de sortie. Elle concerne donc tous les échantillons fournis par le CNA. Comme chaque échantillon peut avoir une valeur n_{LSB} différente, le facteur d'échelle doit être modifié pour chaque intervalle Δt (20 ns pour WLAN ou 25 ns pour UMTS). Cette multiplication corrective doit être donc assez rapide pour améliorer la précision de calcul.

3.2 Mise en œuvre de la chaîne dans le domaine fréquentiel

Après avoir détaillé théoriquement chacun des blocs constituant la chaîne numérique dans le domaine fréquentiel, nous pouvons aborder la mise en œuvre pratique de cette même chaîne. Tout d'abord, nous présentons le choix des signaux de test puis, un par un, chaque élément constituant la chaîne. Les objectifs sont ici de comparer les résultats de la chaîne

fréquentielle avec les valeurs théoriques pour, d'une part, vérifier le bon fonctionnement de chaque bloc, sa sortie devant être proche de la valeur calculée de façon théorique, d'autre part vérifier la discrétisation sur 16 bits des réponses fréquentielles et des matrices $\boldsymbol{W}$ et $\boldsymbol{W_I}$ pour déterminer la précision de calcul et valider ainsi le choix des convertisseurs, mais aussi vérifier l'utilité des facteurs multiplicatifs à la fin de la chaîne.

3.2.1 Justification du signal de test

Afin de tester les différents blocs de l'architecture numérique du simulateur, nous devons utiliser des signaux de test, simple à traiter. Le résultat obtenu avec ces signaux de test doit pouvoir être obtenu par calcul théorique. Si on a un signal court dans le temps, proche d'un dirac, le spectre en fréquence est très étalé et inversement, si un signal est étalé dans le temps, on a un signal proche d'un « dirac » en fréquence. De plus, un module FFT utilise un nombre fini de points et donne des résultats sur un nombre fini de points, ce qui implique l'utilisation d'un signal à support fini aussi bien dans le domaine temporel que dans le domaine fréquentiel. Le signal gaussien répond à ces critères. La gaussienne est donc le bon compromis pour avoir un nombre fini de points à la fois en fréquentiel et en temporel.

Afin de tester le bloc numérique, on utilise donc, d'une part, une gaussienne notée $x(t)$ pour le signal d'entrée et, d'autre part, on crée également une gaussienne $h(t)$ qui, quantifiée, représentera les échantillons des profils de réponses impulsionnelles du canal. Pour le domaine fréquentiel qui nous intéresse ici, nous utiliserons la gaussienne $H(f)$, transformée de Fourier de la gaussienne $h(t)$, pour représenter un profil de réponse fréquentielle qui alimentera le simulateur. Comme le bloc numérique effectue la convolution de deux gaussiennes dans le domaine temporel ou la multiplication de deux gaussiennnes fréquentielles, la sortie $y(t)$ sera, elle aussi, une gaussienne.

Comme on doit obtenir le signal de sortie $y(t)$ donné par la relation :

$$y(t) = x(t) * h(t), \tag{3.29}$$

on exprime les signaux $x(t)$, $h(t)$ et $y(t)$ par :

$$x(t) = x_m.e^{\frac{-(t-m_x)^2}{2.\sigma_x^2}} \tag{3.30}$$

$$h(t) = h_m.e^{\frac{-(t-m_h)^2}{2.\sigma_h^2}} \tag{3.31}$$

$$y(t) = y_m.e^{\frac{-(t-m_y)^2}{2.\sigma_y^2}} \tag{3.32}$$

Leur transformée de Fourier donne les expressions des gaussiennes fréquentielles suivantes :

$$X(f) = x_m \sigma_x \sqrt{2\pi} e^{-(2\pi f)^2 \sigma_x^2} e^{-j2\pi f m_x} \tag{3.33}$$

$$H(f) = h_m \sigma_h \sqrt{2\pi} e^{-(2\pi f)^2 \sigma_h^2} e^{-j2\pi f m_h} \tag{3.34}$$

$$Y(f) = y_m \sigma_y \sqrt{2\pi} e^{-(2\pi f)^2 \sigma_y^2} e^{-j2\pi f m_y} \tag{3.35}$$

Comme la convolution dans le domaine temporel peut être remplacée par la multiplication dans le domaine fréquentiel, on obtient :

$$Y(f) = X(f).H(f) \tag{3.36}$$

$$= x_m h_m \sigma_x \sigma_h (\sqrt{2\pi})^2 e^{-(2\pi f)^2 (\sigma_x^2 + \sigma_h^2)} e^{-j2\pi f (m_x + m_h)}. \tag{3.37}$$

En identifiant les éléments de cette expression avec ceux de la relation (3.35), on obtient :

$$m_y = m_x + m_h \tag{3.38}$$

$$\sigma_y^2 = \sigma_x^2 + \sigma_h^2 \tag{3.39}$$

$$y_m \sigma_y \sqrt{2\pi} = x_m h_m \sigma_x \sigma_h (\sqrt{2\pi})^2 \tag{3.40}$$

Comme on souhaite obtenir le signal de sortie $y(t)$ de façon à ce qu'il soit dans la fenêtre temporelle W_t, on fixe sa moyenne $m_y = \frac{W_t}{2}$ et son écart type $\sigma_y = \frac{m_y}{4}$ car la gaussienne peut être considérée nulle au-delà de $4\sigma_y$. Ainsi, compte tenu des expressions (3.38) et (3.39), on choisit les valeurs de m_x, m_h, σ_x et σ_h. En situation réelle, on ne connaît pas le signal d'entrée, notamment sa puissance. La valeur de l'amplitude des gaussiennes $x(t)$ et $y(t)$ doit alors être choisie afin d'obtenir une bonne précision du signal numérisé. Si les convertisseurs ont une pleine échelle de [- V_m, V_m], le bloc de CAG sera réglé de façon à ce que, d'une part, les entrées radiofréquences acceptent la puissance du signal d'entrée et, d'autre part, les tensions analogiques à l'entrée du CAN se trouvent dans la plage [- $\frac{V_m}{2}$, $\frac{V_m}{2}$] pour éviter les problèmes d'écrêtage. De plus, le signal de sortie $y(t)$ ne doit pas quitter la plage de valeurs des convertisseurs, c'est pourquoi on choisira les valeurs maximales des signaux $x(t)$ et $y(t)$ telles que $x_m = y_m = \frac{V_m}{2}$. Pour déterminer h_m, on peut alors se servir de l'expression (3.40) qui devient :

$$h_m = \frac{\sigma_y}{\sqrt{2\pi} \sigma_x \sigma_h} \tag{3.41}$$

On peut remarquer que les moyennes m_x, m_h et m_y ainsi que les écarts types σ_x, σ_h et σ_y se mesurent en secondes. La relation (3.41) montre que h_m, donc $h(t)$, se mesure en Hz. Ainsi, chaque paramètre des gaussiennes est déterminé et on peut procéder aux tests des différents blocs composant la chaîne dans le domaine fréquentiel.

Les échantillons de la gaussienne temporelle d'entrée $x(t)$ quantifiés, contenus dans le programme VHDL, généré par l'exécution d'un programme MATLAB pour ce test, sont utilisés en entrée du bloc FFT. La gaussienne fréquentielle $H(l\Delta f)$ quantifiée représente, quant à elle, le profil de réponse fréquentielle stocké dans le bloc Mémoire.

3.2.2 Etude de la chaîne fréquentielle

Dans le domaine fréquentiel, la chaîne est réalisée à l'aide des outils de Xilinx. Cette section explique l'évolution des signaux utilisés par chaque bloc qui compose cette chaîne. Les signaux de test utilisés sont donc les gaussiennes décrites précédemment et dont les échantillons sont calculés avec MATLAB. On utilise ces valeurs théoriques des signaux pour vérifier les valeurs fournies par la chaîne implémentée dans le FPGA Virtex-4 avec les outils de Xilinx. Les paragraphes qui suivent font référence au schéma de la figure 3.2.

Signal d'entrée $x(t)$

A l'entrée, comme expliqué à la section précédente, on considère le signal réel gaussien

$$x(t) = x_m.e^{-\frac{(t-m_x)^2}{2\sigma_x^2}}, \ \forall\ t \in [0,\ W_t] \tag{3.42}$$

avec $x_m = 0,5$ V $\in [-V_m, V_m]$, où $V_m = 1$ V.

En raison du CAN, le signal d'entrée réel $x(t)$ est numérisé sur $n_{CNA} = 14$ bits. La quantification du CNA doit assurer la tension $-V_m$ pour le plus petit nombre négatif écrit sur 14 bits, c'est-à-dire $-2^{n_{CNA}-1}$. La quantification du signal $x(t)$ vaut alors :

$$\Delta x = \frac{-V_m}{-2^{n_{CNA}-1}} = \frac{1}{2^{13}} = 0,122 \text{ ms.} \tag{3.43}$$

Ainsi, à la place du vecteur ligne $\boldsymbol{x} = [x(0),\ x(\Delta t),\ \ldots,\ x((N-1)\Delta t)]$ où $N = 512$, le CAN fournit, après conversion, le vecteur ligne $\boldsymbol{n_x}$ tel que :

$$\boldsymbol{n_x} = \text{round}\left\{\frac{\boldsymbol{x}}{\Delta x}\right\} = [n_x(0),\ n_x(1),\ \ldots,\ n_x(N-1)], \tag{3.44}$$

la fonction « round{paramètre} » permettant d'arrondir la valeur du paramètre à l'entier le plus proche (supérieur ou inférieur) et de renvoyer cet entier. Si le paramètre est équidistant de ses deux entiers les plus proches (si le nombre se termine par « ,5 »), la valeur est arrondie à l'entier immédiatement supérieur. On a alors les échantillons :

$$n_x(k) = \text{round}\left\{\frac{x(k\Delta t)}{\Delta x}\right\},\ k\ =\ 0,\ 1,\ \ldots,\ N-1 \tag{3.45}$$

et un vecteur quantifié $\boldsymbol{x_q}$ qui lui correspond :

$$\boldsymbol{x_q} = \Delta x.\boldsymbol{n_x} \approx \boldsymbol{x}. \tag{3.46}$$

Comme expliqué précédemment, le module FFT de Xilinx n'accepte, en entrée, que des échantillons sur un nombre fini de bits égal à 8, 12, 16, 20 ou 24 bits. On choisit alors d'appliquer à l'entrée du module FFT de Xilinx des valeurs sur 16 bits afin de garder la précision du convertisseur. Le passage de 14 bits à 16 bits multiplie donc par 4 le vecteur $\boldsymbol{n_x}$. Cette amplification est décrite par la figure 3.8.

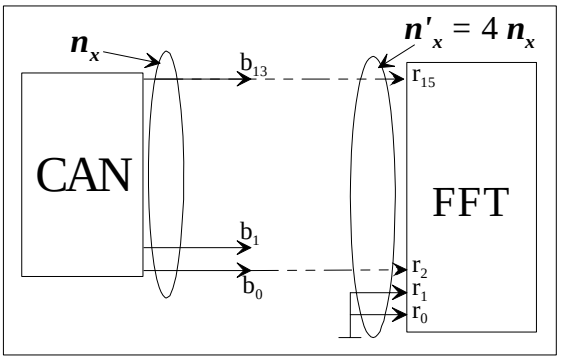

FIG. 3.8 – Amplification du signal d'entrée.

On obtient alors le vecteur ligne $\boldsymbol{n'_x} = 4.\boldsymbol{n_x}$ et le signal d'entrée quantifié $\boldsymbol{x_q}$ est donc également multiplié par 4 :

$$\boldsymbol{x'_q} = \Delta x.\boldsymbol{n'_x} = 4.\Delta x.\boldsymbol{n_x} = 4.\boldsymbol{x_q}. \tag{3.47}$$

Par conséquent, à la place du vecteur $\boldsymbol{x}$ formé par N échantillons, le module FFT de Xilinx reçoit à son entrée réelle le vecteur $\boldsymbol{n'_x}$ à peu près proportionnel au vecteur $\boldsymbol{x}$ formé avec N échantillons du signal d'entrée $x(t)$:

$$\boldsymbol{n'_x} \approx \frac{4}{\Delta x}\boldsymbol{x}. \tag{3.48}$$

Module FFT

Le module FFT fait ensuite le produit entre le vecteur $\boldsymbol{n'_x}$ des échantillons du signal d'entrée, représentés sur 16 bits, et la matrice carrée $\boldsymbol{W_q}$ déterminée précédemment par (3.23). Or, avec l'interface graphique du module FFT de Xilinx [(1)], on choisit le type de

[(1)] La documentation du module FFT de Xilinx utilisé lors de la mise en œuvre de cette chaîne complète n'est plus disponible en ligne. En revanche, on peut se reporter à [74] car il s'agit de la mise à jour de cette documentation, à peu de chose près identique à celle utilisée.

calcul des produits internes avec arrondi pour garder un maximum de précision. Ainsi, chaque produit intermédiaire $n'_x(k).w_q^{lk}$ est arrondi.

Par ailleurs, pour une FFT calculée sur 512 points, le nombre de points est $N = 2^n$, donc $n = 9$. Les valeurs réelles $N_{XR}(l)$ et imaginaires $N_{XI}(l)$, en sortie du bloc FFT de Xilinx, sont alors représentées sur $16 + n + 1$ bits, soit 26 bits, et leur module vaut :

$$N_X(l) = \sum_{k=0}^{N-1} \text{round}\left\{n'_x(k).w_q^{lk}\right\}, \text{ pour } l = 0,\ 1,\ \ldots,\ N-1. \tag{3.49}$$

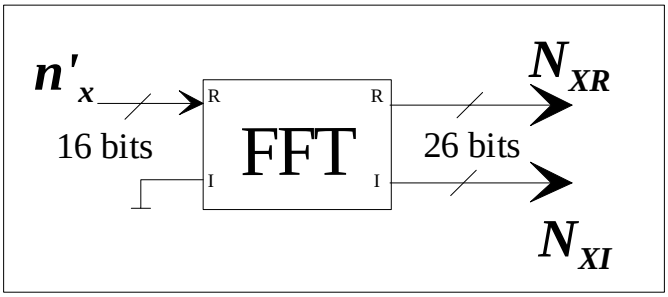

FIG. 3.9 – Utilisation du bloc FFT 512 de Xilinx.

Un bloc FFT, sans quantification, comme par exemple avec MATLAB, effectue le produit $\boldsymbol{x}.\boldsymbol{W}$, tandis que, pour obtenir des échantillons $X(l\Delta f)$ de la transformée de Fourier du signal $x(t)$, on devrait utiliser la relation $\boldsymbol{X} = \Delta t.\boldsymbol{x}.\boldsymbol{W}$. Comme $\boldsymbol{x_q} = \Delta x.\boldsymbol{n_x}$ et que à $\boldsymbol{W}$ on associe $\boldsymbol{W_q}$, on devrait avoir :

$$\boldsymbol{X} \approx \boldsymbol{X_q} = \Delta t.\boldsymbol{x_q}.\boldsymbol{W_q} = \Delta t.\Delta x.\boldsymbol{n_x}.\boldsymbol{W_q} = \frac{1}{4}\Delta t.\Delta x.\boldsymbol{n'_x}.\boldsymbol{W_q}. \tag{3.50}$$

En utilisant les considérations ci-dessus pour le bloc FFT de Xilinx, on obtient :

$$\boldsymbol{X_q} = \frac{1}{4}\Delta t.\Delta x.\boldsymbol{N_X} = \frac{1}{4}\Delta t.\Delta x.(\boldsymbol{N_{XR}} + j\boldsymbol{N_{XI}}). \tag{3.51}$$

Dans la relation (3.51), $\boldsymbol{X_q}$ est le vecteur ligne qui représente une approximation du vecteur $\boldsymbol{X}$ contenant N échantillons complexes de la transformée de Fourier du signal d'entrée. $\boldsymbol{X_q}$ prend alors les valeurs :

$$\boldsymbol{X_q} = \underbrace{[X_q(0),\ X_q(\Delta f), \ldots,\ X_q\left(\left(\frac{N}{2}-1\right)\Delta f\right),\ \underbrace{X_q\left(\frac{N}{2}\Delta f\right)}_{f_{NY}=\frac{N}{2}\Delta f}}_{\text{spectre réel pour } f \in \left[0,\ f_{NY}\right]}, \ldots,\ X_q\left((N-1)\,\Delta f\right)], \tag{3.52}$$

Sachant que la fréquence de Nyquist vaut :

$$f_{NY} = \frac{N}{2}\Delta f = \frac{N\Delta f\Delta t}{2\Delta t} = \frac{1}{2\Delta t} = \frac{f_s}{2} \tag{3.53}$$

et comme $f_s \geq 2B$, on en déduit $f_{NY} \geq B$. Selon l'hypothèse de départ, pour des fréquences $f \geq f_{NY} \geq B$, on a $X(f) = 0$. Autrement dit, les valeurs de $\boldsymbol{X_q}$, calculées pour des fréquences supérieures à la fréquence de Nyquist, ne correspondent pas à un spectre physique réel. Afin de déterminer leur signification, il faut tenir compte de la symétrie hermitienne vérifiée par les valeurs de $\boldsymbol{X_q}$:

$$X_q\left(\left(\frac{N}{2}+k\right)\Delta f\right) = X_q^*\left(\left(\frac{N}{2}-k\right)\Delta f\right), \text{ pour } k = 0,\ 1,\ \ldots,\ \frac{N}{2}-1. \tag{3.54}$$

On rappelle que cette symétrie hermitienne est valable seulement pour des signaux réels appliqués à l'entrée du bloc FFT.

Quantification de la réponse fréquentielle $\boldsymbol{H(f)}$

Pour représenter le canal, si on utilise le signal réel gaussien $h(t)$ tel que :

$$h(t) = h_m e^{-\frac{(t-m_h)^2}{2\sigma_h^2}},\ \forall\ t \in \mathbb{R} \tag{3.55}$$

alors sa transformée de Fourier théorique est :

$$H(f) = \underbrace{h_m\sigma_h\sqrt{2\pi}}_{=\ H_m}.e^{-2\pi^2 f^2\sigma_h^2}.e^{-j.2\pi f m_h} \tag{3.56}$$

ce qui implique, en prenant pour f les valeurs discrètes $l\Delta f$:

$$H(l\Delta f) = H_m.e^{-2\pi^2(l\Delta f)^2\sigma_h^2}.e^{-j.2\pi(l\Delta f)m_h},\ l = 0,\ 1,\ \ldots,\ N-1, \tag{3.57}$$

avec $N\Delta t\Delta f = 1$ et $N = 512$.

Pour la quantification des valeurs $H(l\Delta f)$ sur $M_H = 16$ bits, on utilise :

$$\Delta H = \frac{H_m}{2^{M_H-1}}, \tag{3.58}$$

ce qui conduit au vecteur ligne quantifié $\boldsymbol{H_q}$:

$$\boldsymbol{H_q} = \Delta H.\text{round}\left\{\frac{\boldsymbol{H}}{\Delta H}\right\} = \Delta H.(\boldsymbol{N_{HR}} + j\boldsymbol{N_{HI}}) = \Delta H.\boldsymbol{N_H}. \tag{3.59}$$

La réponse fréquentielle $H(f)$ est alors associée au vecteur ligne $\boldsymbol{H_q} = \Delta H.\boldsymbol{N_H} = \Delta H.(\boldsymbol{N_{HR}} + j\boldsymbol{N_{HI}})$, $\boldsymbol{N_{HR}}$ et $\boldsymbol{N_{HI}}$ étant des vecteurs lignes d'entiers signés, écrits sur $M_H = 16$ bits.

Multiplieur complexe

La sortie complexe $X(f)$ du module FFT est multipliée par la réponse fréquentielle $H(f)$ du canal correspondant, stockée dans la mémoire. On a alors $Y(f) = X(f).H(f)$, ce qui équivaut, pour les vecteurs lignes des échantillons, à $\boldsymbol{N_Y} = \boldsymbol{N_X}.\boldsymbol{N_H^T}$, c'est-à-dire :

$$\boldsymbol{N_{YR}} + j\boldsymbol{N_{YI}} = (\boldsymbol{N_{XR}} + j\boldsymbol{N_{XI}}).(\boldsymbol{N_{HR}} + j\boldsymbol{N_{HI}})^{\boldsymbol{T}} \tag{3.60}$$

$$= \underbrace{\underbrace{(\underbrace{\boldsymbol{N_{XR}}}_{26bits}.\underbrace{\boldsymbol{N_{HR}^T}}_{16bits}}_{42bits} - \underbrace{\underbrace{\boldsymbol{N_{XI}}}_{26bits}.\underbrace{\boldsymbol{N_{HI}^T}}_{16bits}}_{42bits})}_{43bits} + j(\boldsymbol{N_{XR}}.\boldsymbol{N_{HI}^T} + \boldsymbol{N_{XI}}.\boldsymbol{N_{HR}^T}). \tag{3.61}$$

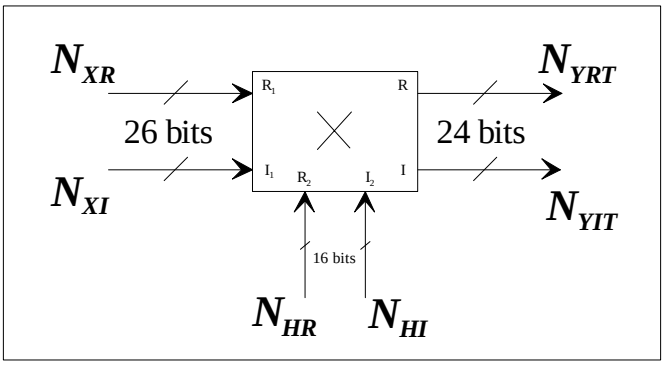

FIG. 3.10 – Utilisation du multiplieur complexe de Xilinx.

Or $X(f)$ et $H(f)$ deviennent :

$$X(f) \rightarrow \boldsymbol{X_q} = \frac{1}{4}\Delta t.\Delta x.(\boldsymbol{N_{XR}} + j\boldsymbol{N_{XI}}), \tag{3.62}$$

$$H(f) \rightarrow \boldsymbol{H_q} = \Delta H.(\boldsymbol{N_{HR}} + j\boldsymbol{N_{HI}}), \tag{3.63}$$

donc $Y(f) = X(f).H(f)$ devient :

$$Y(f) \rightarrow \boldsymbol{Y_q} = \frac{1}{4}\Delta t.\Delta x.\Delta H.(\boldsymbol{N_{YR}} + j\boldsymbol{N_{YI}}). \tag{3.64}$$

Ensuite, le multiplieur complexe [(2)] tronque de manière brutale sur 24 bits les valeurs réelles et imaginaires des sorties $\boldsymbol{N_{YR}}$ et $\boldsymbol{N_{YI}}$ sur 43 bits, 24 bits étant le nombre de bits maximal pour que le signal quantifié puisse être accepté en entrée du module IFFT de Xilinx. Cette troncature s'applique au vecteur produit $\boldsymbol{N_{YR}} + j\boldsymbol{N_{YI}}$. Par exemple, si

$$A = a_{42}.2^{42} + a_{41}.2^{41} + \ldots + a_{19}.2^{19} + a_{18}.2^{18} + \ldots + a_1.2^1 + a_0, \tag{3.65}$$

alors

$$A = (a_{42}.2^{23} + a_{41}.2^{22} + \ldots + a_{20}.2^1 + a_{19}).2^{19} + (a_{18}.2^{18} + \ldots + a_1.2^1 + a_0). \tag{3.66}$$

Ainsi, on a :

$$A = 2^{19}.B + r \text{ où } \begin{cases} B = a_{42}.2^{23} + a_{41}.2^{22} + \ldots + a_{20}.2^1 + a_{19}, \\ r = a_{18}.2^{18} + \ldots + a_1.2^1 + a_0. \end{cases} \tag{3.67}$$

On peut également exprimer B de la façon suivante :

$$B = \left\lfloor \frac{A}{2^{19}} \right\rfloor, \tag{3.68}$$

où $\lfloor x \rfloor$ représente le plus grand nombre entier inférieur ou égal à x, c'est-à-dire sa partie entière.

A l'entrée du module IFFT de Xilinx, on applique donc le vecteur produit $\boldsymbol{N_{YR}}+j\boldsymbol{N_{YI}}$ dont les valeurs sont tronquées sur 24 bits par la relation (3.68). On a donc le nouveau vecteur produit tronqué $\boldsymbol{N_{YT}}$:

$$\boldsymbol{N_{YT}} = \left\lfloor \frac{1}{2^{19}} \boldsymbol{N_{YR}} \right\rfloor + j \left\lfloor \frac{1}{2^{19}} \boldsymbol{N_{YI}} \right\rfloor \approx \frac{1}{2^{19}} (\boldsymbol{N_{YR}} + j\boldsymbol{N_{YI}}), \tag{3.69}$$

ce qui implique :

$$\boldsymbol{N_Y} = \boldsymbol{N_{YR}} + j\boldsymbol{N_{YI}} \approx 2^{19}.\boldsymbol{N_{YT}} \tag{3.70}$$

Ainsi, à la fonction $Y(f)$ correspond le vecteur $\boldsymbol{Y_q}$ exprimé par :

$$\boldsymbol{Y_q} = \frac{1}{4}.\Delta t.\Delta x.\Delta H.(\underbrace{\boldsymbol{N_{YR}} + j\boldsymbol{N_{YI}}}_{= \boldsymbol{N_Y} \approx 2^{19}.\boldsymbol{N_{YT}}}) \approx 2^{17}.\Delta t.\Delta x.\Delta H.\boldsymbol{N_{YT}}. \tag{3.71}$$

[(2)] La documentation du multiplieur complexe utilisé n'est plus disponible en ligne. En revanche, on peut se reporter à [75] car il s'agit de la mise à jour de cette documentation qui, à peu de chose près, est identique à celle utilisée.

Module IFFT

En général, un module IFFT théorique doit calculer le produit $\boldsymbol{Y}.\boldsymbol{W_I}$, avec $\boldsymbol{W_I}$ la matrice du bloc IFFT théorique vue précédemment. C'est à l'utilisateur de multiplier le résultat obtenu par Δf pour retrouver les valeurs du vecteur $\boldsymbol{y}$, proches de celles qui devraient être obtenues en utilisant la formule intégrale de la transformée de Fourier inverse.

Le bloc IFFT de Xilinx, quant à lui, utilise la matrice $\boldsymbol{W_{Iq}}$, la version quantifiée sur un certain nombre de bits de $\boldsymbol{W_I}$. A partir du vecteur $\boldsymbol{N_{YT}}$ de nombres entiers complexes et de la matrice inverse quantifiée $\boldsymbol{W_{Iq}}$, il calcule le vecteur $\boldsymbol{n_y}$ de nombres entiers par :

$$\boldsymbol{n_y} = \text{round}\{\boldsymbol{N_{YT}}.\boldsymbol{W_{Iq}}\}, \tag{3.72}$$

la fonction « round » permettant de retrouver des entiers dans le vecteur $\boldsymbol{n_y}$. On a donc :

$$\boldsymbol{n_y} \approx \boldsymbol{N_{YT}}.\boldsymbol{W_{Iq}}. \tag{3.73}$$

$\boldsymbol{n_y}$ étant le vecteur ligne d'entiers complexes écrits sur 34 bits, sortis de la IFFT. Comme il faut également multiplier le vecteur résultat par le facteur Δf afin de s'approcher du résultat théorique, on peut calculer le signal quantifié de sortie de la IFFT de Xilinx par :

$$\boldsymbol{y_q} \approx \Delta f.\boldsymbol{Y_q}.\boldsymbol{W_{Iq}} \tag{3.74}$$

et en remplaçant par les valeurs précédemment calculées :

$$\begin{aligned}
\boldsymbol{y_q} \approx \Delta f.(2^{17}.\Delta t.\Delta x.\Delta H.\boldsymbol{N_{YT}}).\boldsymbol{W_{Iq}} &= 2^{17}.(\Delta f \Delta t).\Delta x.\Delta H.\boldsymbol{N_{YT}}.\boldsymbol{W_{Iq}} && (3.75)\\
&= \frac{2^{17}}{N}.\Delta x.\Delta H.\boldsymbol{N_{YT}}.\boldsymbol{W_{Iq}} && (3.76)\\
&= 2^8.\Delta x.\Delta H.(\boldsymbol{N_{YT}}.\boldsymbol{W_{Iq}}) && (3.77)\\
&\approx 2^8.\Delta x.\Delta H.\boldsymbol{n_y}. && (3.78)
\end{aligned}$$

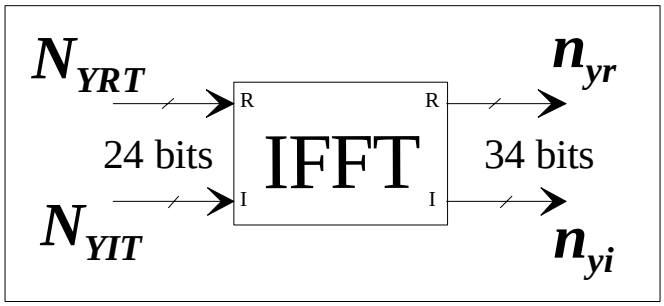

Fig. 3.11 – Utilisation du module IFFT de Xilinx.

Les échantillons du signal de sortie de la IFFT de Xilinx sont donc dans le vecteur ligne $\boldsymbol{y_q}$ tel que :

$$\boldsymbol{y_q} \approx 2^8.\Delta x.\Delta H.\boldsymbol{n_y} \approx 2^8.\Delta x.\Delta H.(\boldsymbol{n_{yr}} + j\boldsymbol{n_{yi}}). \tag{3.79}$$

Comme on sait que $y(t) \in \mathbb{R}$, par la suite on exploite seulement $\boldsymbol{n_{yr}}$ et on obtient $\boldsymbol{y_q}$ proche de $y(t)$:

$$\boldsymbol{y_q} \approx 2^8.\Delta x.\Delta H.\boldsymbol{n_{yr}}. \tag{3.80}$$

Ainsi, lorsque nous obtenons les échantillons du vecteur $\boldsymbol{n_{yr}}$ en sortie de la IFFT de Xilinx, après la chaîne que nous venons de décrire, il est nécessaire de les multiplier par les quantifications de x, de H et le facteur 2^8 pour retrouver les valeurs des échantillons comparables à ce que l'on doit obtenir en théorie.

Troncature et convertisseur numérique-analogique

Les échantillons du vecteur $\boldsymbol{n_{yr}}$ de la sortie réelle du module IFFT sur 34 bits sont ensuite tronqués sur 14 bits pour pouvoir être appliqués à l'entrée du convertisseur numérique-analogique.

En adaptant l'exemple de troncature brutale précédent (3.67), on obtient :

$$B = \left\lfloor \frac{A}{2^{20}} \right\rfloor \quad \text{et donc :} \quad \boldsymbol{n_{y_T}} = \left\lfloor \frac{\boldsymbol{n_{yr}}}{2^{20}} \right\rfloor. \tag{3.81}$$

$$\boldsymbol{n_{y_T}} \approx \frac{\boldsymbol{n_{yr}}}{2^{20}}. \tag{3.82}$$

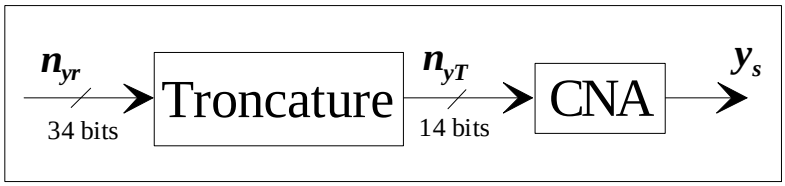

FIG. 3.12 – Utilisation de la troncature et du convertisseur numérique-analogique.

De plus, comme la quantification du CNA est la même que celle du CAN, le vecteur du signal de sortie quantifié $\boldsymbol{y_s}$ peut s'exprimer à l'aide de la relation :

$$\boldsymbol{y_s} = \Delta y.\boldsymbol{n_{y_T}} = \Delta x.\boldsymbol{n_{y_T}} \tag{3.83}$$

et avec les relations (3.82) puis (3.80), on obtient :

$$\boldsymbol{y_s} = \Delta x.\boldsymbol{n_{y_T}} \approx \Delta x.\frac{\boldsymbol{n_{yr}}}{2^{20}} \tag{3.84}$$

$$\approx \frac{\Delta x}{2^{20}}.\frac{\boldsymbol{y_q}}{2^8.\Delta x.\Delta H} \tag{3.85}$$

$$\approx \frac{\boldsymbol{y_q}}{2^{28}.\Delta H}, \tag{3.86}$$

ce qui donne :

$$\boldsymbol{y_q} = 2^{28}.\Delta H.\boldsymbol{y_s}. \tag{3.87}$$

Par conséquent, à partir du signal $\boldsymbol{y_s}$ quantifié, fourni par la sortie du CNA, il faut utiliser le facteur multiplicatif $2^{28}.\Delta H$ pour retrouver un signal quantifié dont les échantillons sont proches des vraies valeurs du signal théorique $y(t)$. Dans le cas de la troncature glissante, le facteur multiplicatif sera $2^{28+n_{LSB}}.\Delta H$.

Dans le cas où le signal de sortie aurait une faible amplitude qui n'exploiterait pas les 14 bits avant d'entrer dans le CNA, il est possible d'amplifier par un facteur multiplicatif les vraies valeurs de $H(f)$ pour amplifier le signal de sortie de façon à éviter la troncature glissante à la fin de la chaîne fréquentielle. La correction devrait être transmise vers le bloc de CAG qui va décider du facteur d'atténuation du signal d'entrée et du facteur correctif du signal de sortie $y(t)$ en fonction de l'atténuation de $x(t)$, de la valeur n_{LSB} de la troncature globale et du facteur d'amplification de $H(f)$ utilisé pour assurer une bonne précision des calcul.

3.2.3 Résultats et précision de calcul

Avant de détailler les résultats obtenus, il est nécessaire d'éclaircir un point important des algorithmes de FFT et de IFFT. La sortie du bloc FFT (de MATLAB et de Xilinx) fournit des échantillons complexes. Comme on a un vecteur réel en entrée du bloc FFT, on peut remarquer que les échantillons calculés par la FFT présentent une symétrie hermitienne par rapport à l'indice $\frac{N}{2}$ correspondant à la fréquence de Nyquist (relation (3.54)). Dans le but de comparer les résultats calculés par la FFT et les résultats obtenus par le calcul théorique, on retient seulement les résultats calculés par la FFT qui correspondent au spectre correct dans la bande [0, f_{NY}], avec la fréquence de Nyquist $f_{NY} = \frac{f_s}{2}$ (relation (3.53)). Entre [f_{NY}, f_s], on a en réalité le spectre correspondant à la bande [- f_{NY}, 0]. Ainsi, si on souhaite visualiser le tracé du module de la sortie complexe d'une FFT de 512 points, on ne garde que les résultats sur les 257 points représentant les valeurs de la bande [0, f_{NY}]. Dans notre cas, nous nous intéressons à la chaîne complète, on garde alors toutes les valeurs de la fenêtre temporelle W_t en sortie de la IFFT. Nous pouvons donc commenter les résultats obtenus en sortie de la chaine complète dont la mise en œuvre a été décrite dans la section précédente.

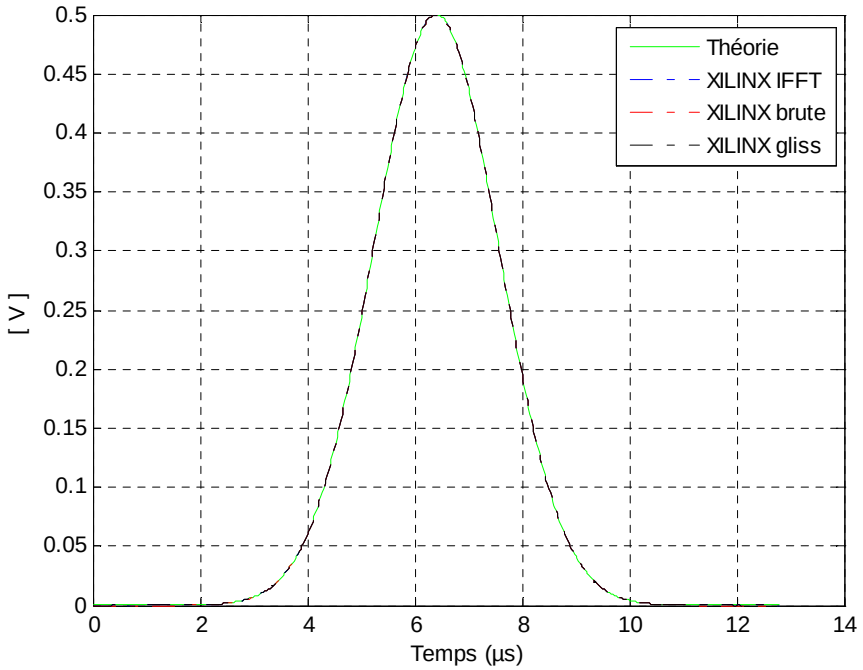

FIG. 3.13 – Tension de sortie de la chaîne fréquentielle (échelle linéaire).

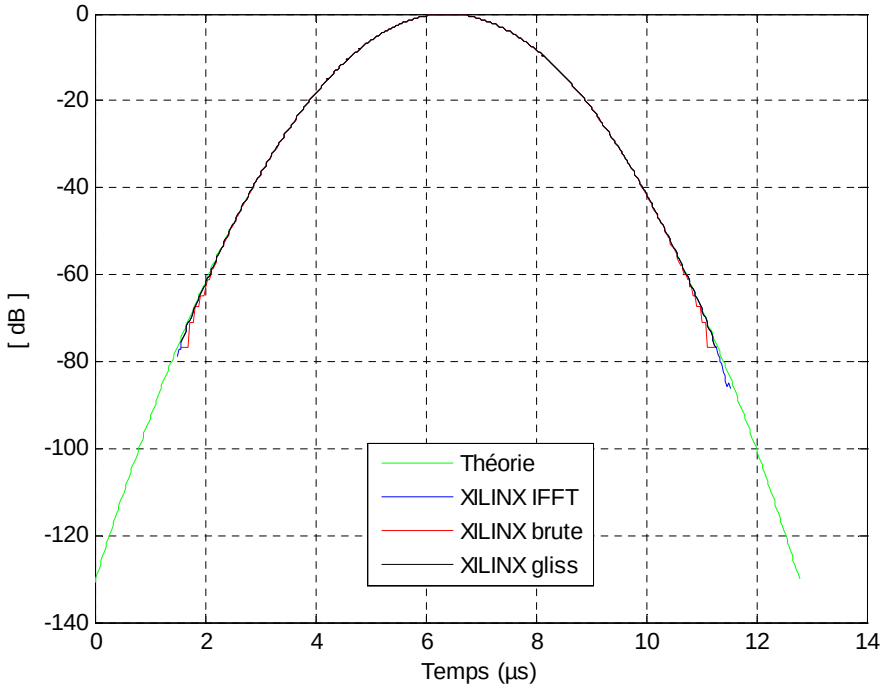

FIG. 3.14 – Tension de sortie de la chaîne fréquentielle (échelle logarithmique).

Le signal de sortie, au bout de la chaîne complète, est présenté aux figures 3.13 et 3.14, la première en échelle linéaire et la seconde en échelle logarithmique par rapport à sa valeur maximale.

Plusieurs signaux y sont en fait présentés : le tracé de la courbe « Théorie » montre le signal théorique tracé à partir de l'expression (3.32), c'est-à-dire le signal de sortie $y(t)$ duquel le signal de sortie calculé par la chaîne XILINX $y_c(t)$ doit se rapprocher. La courbe « XILINX IFFT » montre la sortie du module IFFT avant la troncature, ce sont donc les valeurs réelles sur 34 bits auxquelles on va appliquer ensuite la troncature nécessaire, détaillée à la section précédente. Il s'agit du cas qui doit être le plus proche de la théorie. La courbe « XILINX brute » représente la sortie après la troncature brutale sur 14 bits et la courbe « XILINX gliss » la sortie après la troncature glissante détaillée à la section précédente.

Il existe une très bonne concordance entre les signaux de sortie et la théorie, c'est pourquoi, afin de mieux discerner les différences, on détermine l'erreur relative appelée $\epsilon_r(k)$ en pourcentage et le rapport signal sur bruit (*SNR - Signal to Noise Ratio*) relatif appelé $SNR_r(k)$ en dB par rapport au signal gaussien théorique, pour les différents cas du signal de sortie. Si on appelle $y_P(k\Delta t)$ les échantillons des sorties des différents cas qu'on obtient en pratique (sans troncature, avec troncature brutale puis glissante) et $y_T(k\Delta t)$ les échantillons de la gaussienne théorique que l'on doit retrouver en sortie, on peut définir l'erreur relative en pourcentage par :

$$\epsilon_r(k) = \frac{|y_P(k\Delta t)| - |y_T(k\Delta t)|}{|y_T(k\Delta t)|}.100 \text{ [\%]} \tag{3.88}$$

et le SNR en dB par :

$$SNR_r(k) = 20.\log_{10} \frac{|y_T(k\Delta t)|}{|y_P(k\Delta t)| - |y_T(k\Delta t)|} \text{ [dB]} \tag{3.89}$$

Pour chaque échantillon du signal de sortie de la chaîne fréquentielle pour les différents cas, l'erreur relative et le SNR relatif sont calculés. On peut ensuite tracer les courbes qui leur correspondent.

Comme on peut le voir sur la figure 3.15 (respectivement 3.16), « XILINX IFFT » représente l'erreur relative (respectivement le SNR) des échantillons obtenus en sortie de la IFFT, par rapport aux échantillons du signal théorique $y(t)$. Quant aux tracés « XILINX brute » et « XILINX gliss », ils représentent les erreurs relatives (respectivement les SNR relatifs) des échantillons obtenus en sortie de la troncature brutale et glissante, toujours par rapport aux échantillons du signal théorique $y(t)$.

En comparant les figures 3.14 et 3.15, on peut remarquer que les plus grandes valeurs du signal de sortie, c'est-à-dire les valeurs supérieures à -40 dB par rapport à la valeur maximale du signal de sortie, ont une erreur relative inférieure à 1 % en valeur absolue.

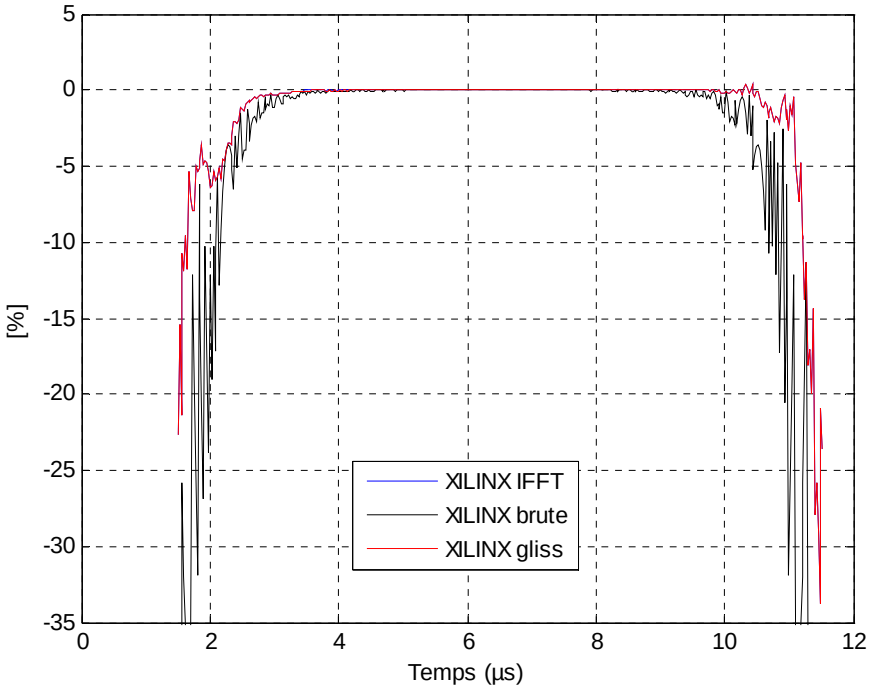

FIG. 3.15 – Erreur relative du signal de sortie de la chaîne fréquentielle.

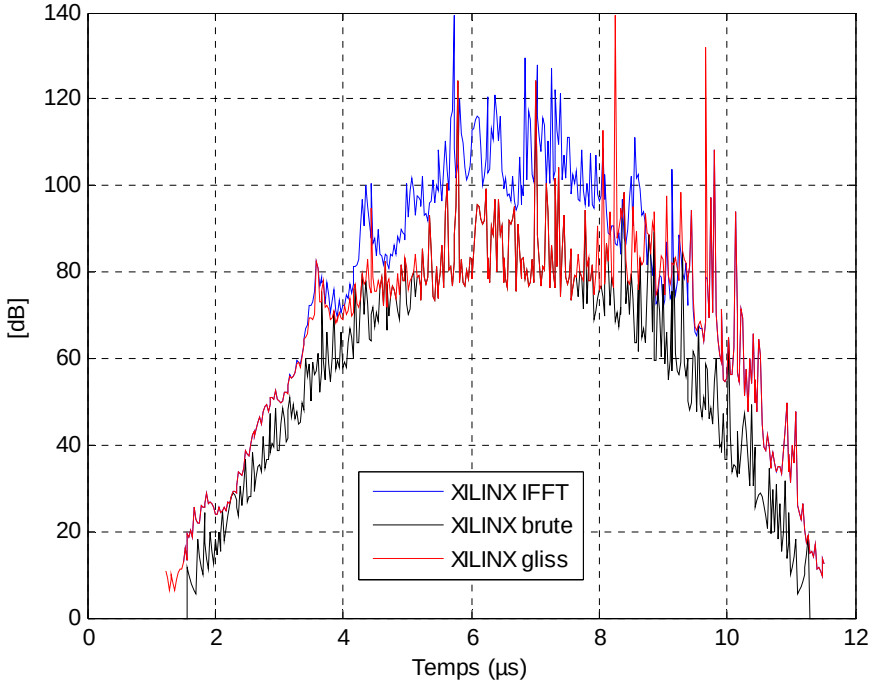

FIG. 3.16 – Rapport signal à bruit du signal de sortie de la chaîne fréquentielle.

L'erreur peut être beaucoup plus importante mais seulement pour des petites valeurs du signal de sortie car le signal gaussien de test est proche de 0.

De plus, on constate que l'erreur relative des échantillons de sortie obtenus par troncature glissante « XILINX gliss » est tellement proche de l'erreur relative des échantillons du signal obtenus en sortie de la IFFT « XILINX IFFT » qu'on ne peut pas les distinguer sur la figure 3.15. On peut donc conclure que la troncature glissante s'effectue avec très peu d'erreur à la sortie de la IFFT, et qu'elle est meilleure que la troncature brutale envisagée au départ, surtout pour les faibles valeurs du signal de sortie.

On peut faire les mêmes remarques sur la figure 3.16 qui présente le SNR, paramètre plus utilisé en ingénierie ou en traitement du signal que l'erreur relative, paramètre plus mathématique.

Par ailleurs, pour les faibles valeurs du signal de sortie, l'erreur relative calculée pour chaque échantillon peut atteindre -100 % (on se limite à - 35 % sur la figure 3.15), c'est pourquoi les valeurs globales de l'erreur relative ϵ et du rapport signal sur bruit *SNR* du signal de sortie, avant et après la troncature finale, sont également calculées avec les relations suivantes :

$$\epsilon = \frac{\| e \|}{\| y \|}.100 \text{ [\%]} \quad \text{et} \quad SNR = 20.\log_{10}\frac{\| y \|}{\| e \|} \text{ [dB]} \tag{3.90}$$

où $e = y_c - y$ avec y_c le signal de sortie calculé (avec ou sans troncature finale) et y le signal de sortie théorique en sachant que, pour un vecteur $x = [x_1, x_2, \ldots, x_N]$, sa norme euclidienne $\| x \|$ se calcule avec la relation :

$$\| x \| = \sqrt{\frac{1}{N}\sum_{k=1}^{N} x_k^2} \tag{3.91}$$

On peut observer les résultats mis en forme dans le tableau suivant :

	Erreur (%)	SNR (dB)
Sortie IFFT sans troncature	0.0058	84.79
Sortie IFFT avec troncature glissante	0.0114	78.86
Sortie IFFT avec troncature brutale	0.0228	72.83

TAB. 3.2 – Comparaison des valeurs globales de l'erreur relative et du SNR.

Les valeurs globales de l'erreur relative et du SNR sont effectivement meilleures avec la troncature glissante. La sortie après la troncature glissante est donc globalement plus proche du signal théorique.

Après avoir affiner les résultats, nous avons souhaité connaître le taux d'occupation du Virtex-IV SX35 pour la chaîne complète du domaine fréquentiel. Ce taux d'occupation est

donné dans le tableau 3.3 pour toute la chaîne dont l'architecture a été détaillée tout au long de ce chapitre.

Nombre de « Slices »	5996 sur 15360	39%
Nombre de LUT	9306 sur 30720	30%
Nombre de blocs RAM	29 sur 192	15%
Nombre de bascules	10828 sur 30720	35%
Nombre de « XtremeDSP »	56 sur 192	29%

TAB. 3.3 – Taux d'occupation du composant pour la chaîne fréquentielle.

On peut constater que la chaîne fréquentielle n'occupe pas la totalité du composant programmable. Par ailleurs, nous avons souhaité connaître la latence de cette chaîne fréquentielle. Cette latence est mesurée entre le moment où les données entrent dans le bloc FFT et le moment où le résultat est fourni en sortie du bloc IFFT, ce qui donne une latence d'environ 45 μs mesurée par l'outil Modelsim. Comme l'affichage de cette mesure complète n'est pas possible compte tenu de la largeur de l'écran, la figure 3.17 montre l'exemple de la mesure de la latence pour un seul module FFT 512.

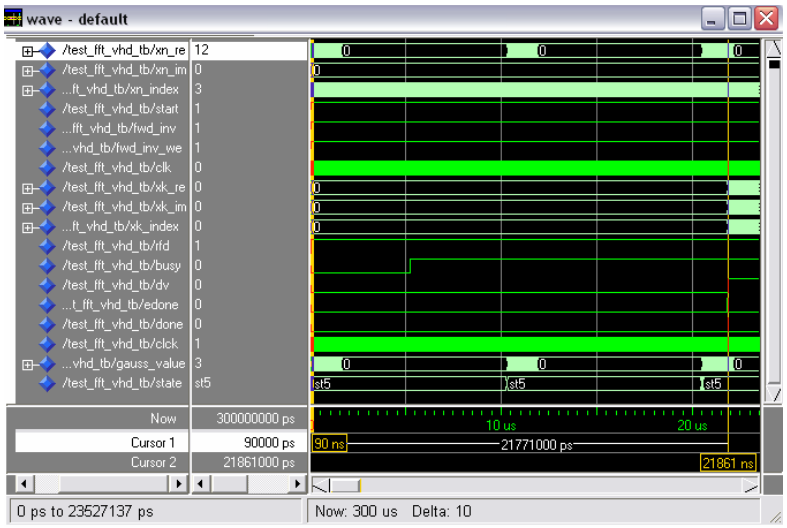

FIG. 3.17 – Exemple de détermination de la latence d'un seul module FFT avec Modelsim.

On peut voir, en bas de cette figure, la mesure de 21,7 μs de latence mesurée entre le moment où les données entrent dans le bloc FFT et le moment où le résultat est fourni.

De plus, pour la chaîne complète, il est nécessaire d'ajouter approximativement 38 ns de latence du convertisseur analogique-numérique, et 17 ns de latence du convertisseur numérique-analogique selon leur documentation constructeur respective mais ces valeurs sont négligeables par rapport aux 45 μs mesurées pour la latence de la chaîne fréquentielle.

En résumé, la mesure de la latence nous montre que nous devrions avoir, en réalité, une latence d'environ 45 μs pour le domaine fréquentiel, ce qui impose, si on considère la célérité de l'onde $c = 3.10^8$ m/s, une distance entre les antennes Tx et Rx augmentée de 13,5 km.

En effet, en reprenant la représentation d'un seul profil de réponse impulsionnelle, comme sur la figure 2.8, pour un instant t fixé, on sait qu'il n'y aura pas de latence supplémentaire due au calcul de la partie numérique si la latence de ce bloc est inférieure au retard absolu τ_0, ou au minimum des τ_0 si on considère plusieurs profils.

Or, ici, la latence L, mesurée de 45 μs, augmente le retard de la réponse impulsionnelle et tout se passe comme si la réponse impulsionnelle était retardée de la valeur de la latence L, comme nous pouvons le voir sur la figure 3.18.

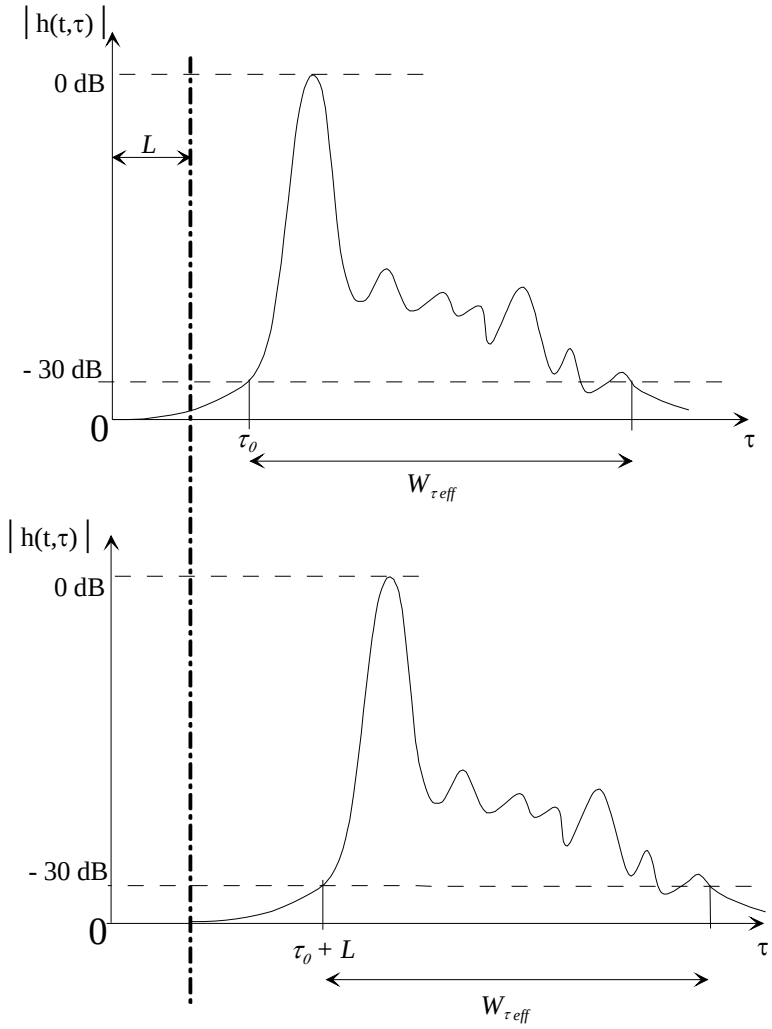

FIG. 3.18 – Interprétation de la latence

Sachant que $\tau_0 = \frac{d}{c}$, tout se passe comme si l'on avait la distance entre antennes qui était augmentée de la distance $d = c \times L$ soit ici 13,5 km. Le retard est alors trop élevé compte tenu de la puissance reçue qui est donc elle-même trop élevée pour des retards si importants. On ne simule donc pas le vrai canal. Pour des applications de simples transmissions uniquement, la latence peut ne pas être un problème. Mais, pour des applications de type géolocalisation qui utilisent le lien entre la puissance reçue et les retards, ce phénomène de latence doit être pris en compte.

Par ailleurs, on souhaite simuler un canal MIMO, composé de plusieurs canaux SISO dont chacun est décrit par la chaîne fréquentielle. Compte tenu du nombre de convertisseurs numériques-analogiques et analogiques-numériques présents sur la carte, il est possible de simuler un canal MIMO 2×2 unidirectionnel comme décrit à la figure 1.7.

Le tableau 3.4 montre le taux d'occupation du composant pour une telle architecture. Nous pouvons en conclure qu'une telle architecture numérique pour simuler, dans le domaine fréquentiel, un canal MIMO unidirectionnel 2×2 tient dans un seul composant programmable Virtex-4.

Nombre de « Slices »	14055 sur 15360	91%
Nombre de LUT	21830 sur 30720	71%
Nombre de blocs RAM	61 sur 192	31%
Nombre de bascules	25759 sur 30720	83%
Nombre de « XtremeDSP »	176 sur 192	91%

TAB. 3.4 – Taux d'occupation du composant pour un canal MIMO 2×2 unidirectionnel.

Le cas considéré ici est le cas le plus défavorable du canal radio-mobile rural car il a une grande fenêtre W_t d'observation de la réponse impulsionnelle. D'autre part, le cas de la gaussienne de longueur égale à la taille du bloc FFT est un cas particulier.

On considère maintenant un signal $x(t)$ continu, toujours exprimé par une loi gaussienne et quantifié comme vu à la section 3.2.2, mais de longueur suffisante pour être utilisé par le bloc FFT de Xilinx en mode continu, dit *streaming*. Par exemple, une longueur de trois fenêtres W_t devrait suffire et le bloc FFT se chargera de scinder en trois fois 512 échantillons le vecteur du signal $x(t)$ quantifié par le CAN.

On fixe la moyenne du signal d'entrée $x(t)$ à $m_x = \frac{3W_t}{2}$ et son écart type $\sigma_x = \frac{3W_t}{8}$ soit $\frac{m_x}{4}$. Par la relation 3.5, on obtient alors un vecteur de longueur trois fois 512 échantillons de la gaussienne $x(t)$. Les 512 premiers échantillons forment le vecteur $\boldsymbol{x_1}$, les 512 échantillons suivants forment le vecteur $\boldsymbol{x_2}$ et les 512 derniers échantillons forment le vecteur $\boldsymbol{x_3}$.

On applique ce signal à l'entrée d'un système linéaire (c'est-à-dire la chaîne fréquentielle étudiée à la section 3.2.2) dont la réponse impulsionnelle est un signal continu $h(t)$, toujours représenté par une gaussienne définie par la relation 3.31, mais avec $m_h = \frac{W_t}{2}$ et $\sigma_h = \frac{m_h}{4}$. En appliquant cette chaîne fréquentielle sur chaque vecteur $\boldsymbol{x_1}$, $\boldsymbol{x_2}$ et $\boldsymbol{x_3}$, on obtient en

sortie trois vecteurs $\boldsymbol{y_1}$, $\boldsymbol{y_2}$ et $\boldsymbol{y_3}$. La concaténation de ces trois vecteurs forme un vecteur d'une longueur de trois fois 512 échantillons.

En pratique, le découpage de la sortie en $\boldsymbol{y_1}$, $\boldsymbol{y_2}$ et $\boldsymbol{y_3}$ est assuré par le mode *streaming* du bloc FFT de Xilinx. Pour valider ce mode *streaming*, on compare, grâce à MATLAB, la concaténation des trois vecteurs $\boldsymbol{y_1}$, $\boldsymbol{y_2}$ et $\boldsymbol{y_3}$ avec le vecteur issu de l'expression théorique $y(t)$ donnée par 3.32 ou du résultat du produit de convolution entre les signaux $x(t)$ et $h(t)$. On remarque que les valeurs obtenues en concaténant les 3 sorties sont différentes des valeurs théoriques, ce qui ne permet pas de valider cette architecture dans le cas d'un signal continu appliqué en entrée de la chaîne fréquentielle. Cette différence est d'autant plus marquée en comparant les représentations graphiques de la figure 3.19.

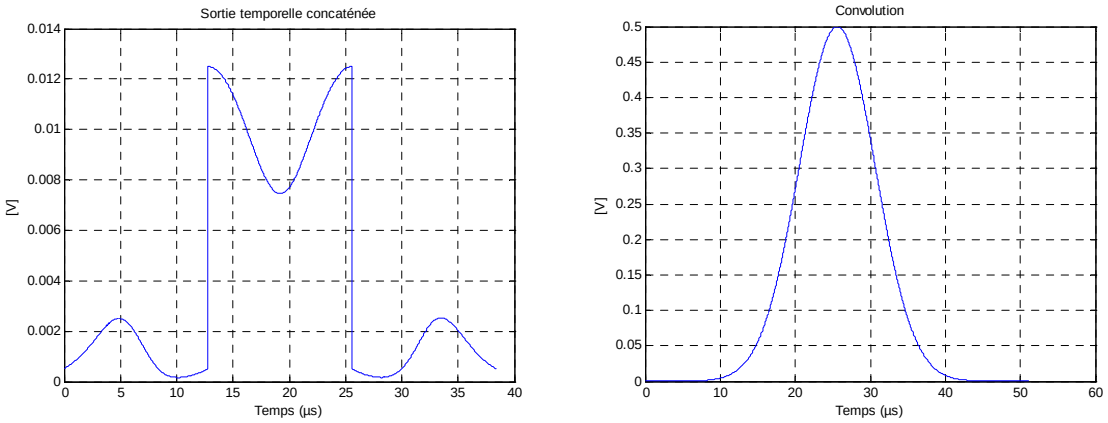

FIG. 3.19 – Comparaison de la sortie temporelle concaténée et du produit de convolution.

Pourtant, d'un point de vue théorique, on peut affirmer qu'un produit de convolution dans le domaine temporel et une tranformée de Fourier dans le domaine fréquentiel donnent des résultats équivalents. On peut donc se poser la question de savoir pourquoi cette affirmation n'est pas vérifiée par nos calculs. Par ailleurs, on voit que la succession des trois morceaux de y, qui forment la réponse globale de l'architecture fréquentielle, dure $3W_t$, tandis que le calcul théorique utilisant la convolution temporelle indique un résultat sur $4W_t$. Le problème doit donc venir du fait que la durée de représentation des trois réponses $y_1(t)$, $y_2(t)$ et $y_3(t)$ obtenues avec la solution fréquentielle est seulement de W_t, tandis qu'une convolution donnerait des réponses sur 2 W_t. Autrement dit, c'est comme si on « coupait » ces signaux avant qu'ils ne s'annulent. Le calcul suivant ne se fait donc pas en tenant compte de tous les échantillons du signal précédemment traité.

Pour que le bloc IFFT puisse donner toute la réponse temporelle à l'excitation, de $x_1(t)$ par exemple, on devrait lui permettre de donner sa réponse sur 2 W_t, comme pour le filtre FIR. Par conséquent, on rallonge chaque morceau $x_1(t)$, $x_2(t)$ et $x_3(t)$ du cas précédent avec une « queue » de 512 échantillons nuls, ce qui permet d'utiliser les blocs FFT pour

des signaux décrits sur 2 W_t. Cette « queue » nulle ne modifie pas la description des signaux $x_1(t)$, $x_2(t)$ et $x_3(t)$ mais permet aux blocs FFT et IFFT de travailler avec plus d'échantillons, ici le double soit 1024. Par conséquent, après le bloc IFFT, les réponses partielles $y_1(t)$, $y_2(t)$ et $y_3(t)$ seront décrites sur 2 W_t, comme dans le cas de l'utilisation d'un filtre FIR. Les échantillons nuls seront alors exploités en utilisant deux blocs FFT (et deux blocs IFFT) de 1024 points dans lesquels les échantillons entreront en alternance grâce à un signal de basculement (*switch*), comme on peut le voir à la figure 3.20.

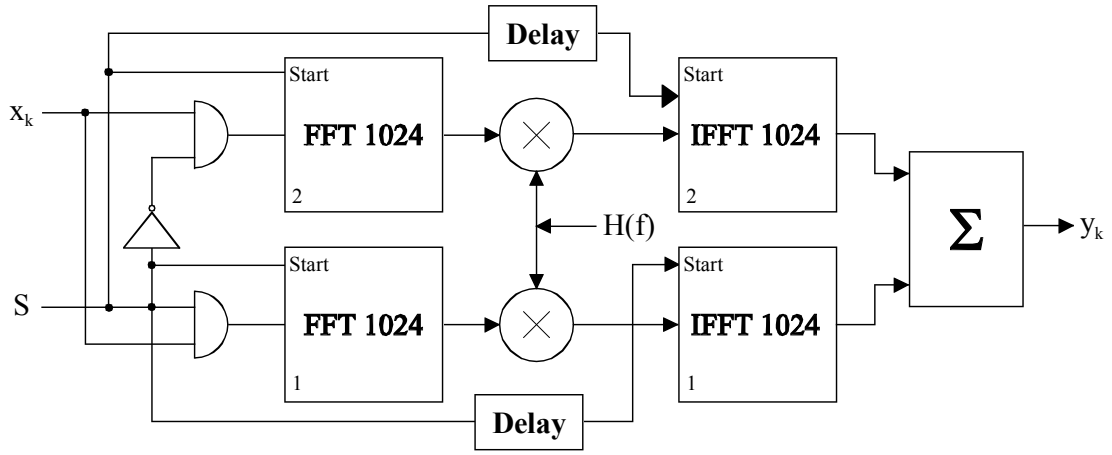

FIG. 3.20 – Architecture fréquentielle d'un canal SISO.

Pendant 512 coups d'horloge, si S vaut 1, 512 échantillons sont envoyés vers le bloc FFT1, tandis que le bloc FFT2 reçoit 512 échantillons nuls car la porte AND non validée met l'entrée des données sur 14 bits à zéro.

Après 512 coups d'horloge, le signal S commute et vaut 0. C'est alors le bloc FFT2 qui reçoit 512 « vrais » échantillons, tandis que le bloc FFT1 reçoit 512 échantillons nuls.

En général, un bloc FFT considère qu'en dehors des échantillons fournis, le signal d'entrée est nul. C'est ainsi que la formule générale (3.1) de calcul d'une transformée de Fourier peut se réduire au calcul de l'intégrale sur $[0, W_t]$, comme indiqué par la formule (3.2). Par conséquent, le rajout d'une partie nulle à chaque morceau du signal $x(t)$ ne modifie pas ces morceaux analogiques. C'est seulement leur description numérique qui se fait avec plus d'échantillons (on double le nombre d'échantillons). Leur somme donne toujours le signal initial $x(t)$.

Après les blocs FFT, on obtient alors les échantillons des transformées de Fourier des tranches du signal d'entrée $x_1(t)$, $x_2(t)$ et $x_3(t)$, prolongées avec une « queue » de 512 échantillons. Avec les relations de type :

$$Y_1(f) = X_1(f)H(f) \tag{3.92}$$

on multiplie ces échantillons avec la réponse fréquentielle du canal $H(f)$, calculée pour 1024 échantillons également, $h(t)$ ayant été complétée de la même façon avec 512 échantillons nuls. Les deux blocs « Delay » de la nouvelle architecture permettent d'assurer un retard qui tient compte du retard introduit par le multiplieur numérique. Pour des blocs FFT de $1024 = 2^{10}$ points, le retard devrait être de 10 coups d'horloge.

Les produits obtenus sont alors les transformées de Fourier des morceaux de sortie $y_1(t)$, $y_2(t)$ et $y_3(t)$. Ces derniers signaux sont calculés avec des blocs IFFT sur 1024 échantillons, fonctionnant sur le même principe de l'alternance que les blocs FFT. Cependant, comme ce bloc fournit le résultat à partir de l'origine indiquée par le début de la séquence d'entrée, il faut placer convenablement les signaux $y_1(t)$, $y_2(t)$ et $y_3(t)$ sur l'axe du temps. On a alors $y_1(t)$ sur $[0, 2\ W_t]$, $y_2(t)$ sur $[W_t, 3\ W_t]$ et $y_3(t)$ sur $[2\ W_t, 4\ W_t]$. Après avoir complété ces 3 signaux par des valeurs nulles sur $[0, 4\ W_t]$, on obtient $y_L1(t)$, $y_L2(t)$ et $y_L3(t)$, tous ces signaux étant décrits sur $[0, 4\ W_t]$.

Ces trois signaux $y_L1(t)$, $y_L2(t)$ et $y_L3(t)$, décrits sur $[0, 4\ W_t]$ et fournis par les deux blocs IFFT, sont alors additionné grâce à la somme effectuée entre les deux sorties des blocs IFFT pour donner la bonne réponse du canal SISO $y_L(t)$ sur $[0, 4\ W_t]$, comme le fait le filtre FIR, et comme on peut le voir sur les courbes de la figure 3.21.

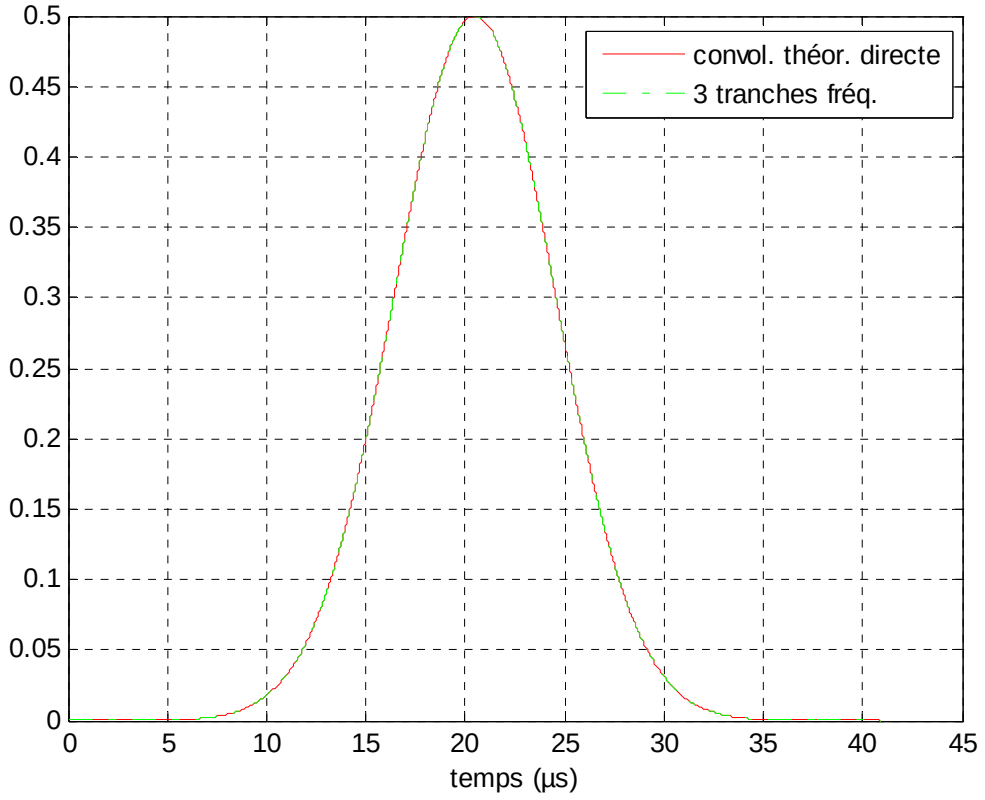

FIG. 3.21 – Architecture fréquentielle d'un canal SISO.

On constate alors sur ces courbes l'identité entre le calcul temporel (en rouge convolution théorique directe) et le calcul fréquentiel (en vert issu des trois tranches passées dans la nouvelle architecture fréquentielle présentée). De plus, on peut remarquer la durée plus longues de ces réponses étendues. C'est ici la différence par rapport à la première architecture qui utilisait des blocs FFT et IFFT de 512 points : les signaux de sortie étaient coupés, on n'avait pas l'évolution complète des signaux de sortie. Cette fois, les signaux de test ont une durée largement supérieure à W_t. Le mode *streaming* des blocs FFT et IFFT est convenablement utilisé, ce qui n'était pas le cas avec une gaussienne d'entrée d'une durée W_t.

Cette comparaison prouve que le mode *streaming* permet d'obtenir des résultats équivalents entre le calcul temporel et le calcul fréquentiel. Cependant, malgré les bons résultats donnés par cette nouvelle architecture fréquentielle, cette dernière n'est pas sans défaut. D'une part, les blocs FFT et IFFT utilisés sont plus nombreux à cause du fonctionnement alterné et ont eux-mêmes chacun une taille double par rapport à la solution précédente, ce qui augmente considérablement les ressources utilisées. D'autre part, le fonctionnement des multiplieurs et l'attente avant l'entrée dans les blocs IFFT introduisent une latence supplémentaire. Ces raisons suffisent à montrer que, même si d'un point de vue théorique cette solution est viable, il est préférable de s'orienter vers une solution moins gourmande en ressources et en latence. Dans ce cas, on présentera, dans le chapitre suivant, une solution d'architecture à base de filtres FIR 512 à 32 trajets. Il s'agira d'une solution de remplacement de cette solution fréquentielle pour les environnements impliquant de longues réponses impulsionnelles, c'est-à-dire dans le cas typique de l'UMTS rural.

3.3 Conclusion

Dans un milieu outdoor, comme les réponses impulsionnelles sont plus longues qu'en milieu indoor et plus souvent rafraîchies, la partie numérique du simulateur matériel de propagation a été réalisée, dans un premier temps, dans le domaine fréquentiel. Après avoir revu le fonctionnement d'un module FFT et le lien avec la transformée de Fourier, cette approche de la partie numérique dans le domaine fréquentiel a montré l'évolution du signal à travers les différents éléments la constituant, tels que le module FFT et le multiplieur complexe, entre autres. Ce choix d'architecture permet d'utiliser moins de ressources qu'une architecture à base de filtres FIR 512. En effet, si on avait choisit une architecture à base de FIR de même longueur 512 pour simuler ces réponses impulsionnelles de canaux, nous aurions eu besoin d'environ 16 000 slices, estimé à partir de l'outil « CORE Generator » de Xilinx, ce qui est de toute façon trop pour un Virtex-4. Cependant, l'inconvénient de cette approche fréquentielle est surtout le manque de précision des résultats si le nombre d'échantillons représentant le signal de sortie dépasse la taille des blocs FFT et IFFT. On a alors envisagé

une autre solution fréquentielle qui, grâce au prolongement des signaux partiels d'entrée avec une séquence nulle, permet de fournir les bons résultats, en sortie des blocs FFT et IFFT. Pourtant, cette solution implique, d'une part, une latence pouvant devenir trop grande selon l'application choisie et, d'autre part, une utilisation de ressources démesurée. C'est pourquoi, il est intéressant de s'orienter vers une solution où notamment la latence est plus faible en utilisant une approche temporelle de la partie numérique, comme nous allons le voir dans le chapitre suivant.

Chapitre 4

Approche temporelle

Sommaire

4.1 Description des différents blocs de la chaîne temporelle . . . 90

4.1.1 Etude de la convolution 91

4.1.2 Etude du bloc Mémoire 93

4.2 Mise en œuvre de la chaîne dans le domaine temporel 95

4.2.1 Quantification du signal d'entrée et de la réponse impulsionnelle 96

4.2.2 Filtre FIR 64 96

4.2.3 Troncature et convertisseur numérique-analogique 98

4.2.4 Résultats et précision de calcul 99

4.3 Solution de substitution de l'approche fréquentielle 104

4.3.1 Architecture du filtre FIR 512 à 32 coefficients 105

4.3.2 Chargement des coefficients 106

4.3.3 Résultats et précision de calcul 108

4.4 Conclusion 113

Dans un environnement « indoor », les réponses impulsionnelles sont courtes, de l'ordre de la centaine de nanosecondes (cf. tableau 2.2), ce qui peut être incompatible avec la latence des modules FFT et IFFT. Ainsi, la partie numérique du simulateur de canal pour ces environnements indoor peut être réalisée dans le domaine temporel par des opérations de convolution. L'approche temporelle de la partie numérique du simulateur matériel de propagation est détaillée dans ce chapitre.

Afin d'expliquer cette approche temporelle, nous considérons toujours un seul canal SISO puisque plusieurs canaux SISO composent un canal MIMO comme illustré à la figure 1.7. Comme pour l'approche fréquentielle, dans une première partie, chaque bloc qui compose l'architecture temporelle sera étudié d'un point de vue traitement du signal. Dans une seconde partie, nous verrons la chaîne temporelle en numérique dans sa globalité pour la tester, étudier sa précision, et commenter les résultats.

4.1 Description des différents blocs de la chaîne temporelle

Le schéma de principe de l'architecture numérique du simulateur matériel de propagation dans le domaine temporel est présenté à la figure 4.1. Ce schéma montre l'évolution du signal d'entrée d'un point de vue traitement du signal.

Comme nous traiterons cette chaîne temporelle en numérique, nous nous appuierons sur le schéma de la figure 4.2 qui décrit la même chaîne temporelle mais du point de vue de la représentation numérique de ces signaux.

Cette configuration met en œuvre un module effectuant la convolution du signal d'entrée avec les profils de réponses impulsionnelles du canal considéré, ainsi qu'un module de troncature permettant de représenter le signal de sortie sur un nombre de bits acceptable par le convertisseur numérique-analogique.

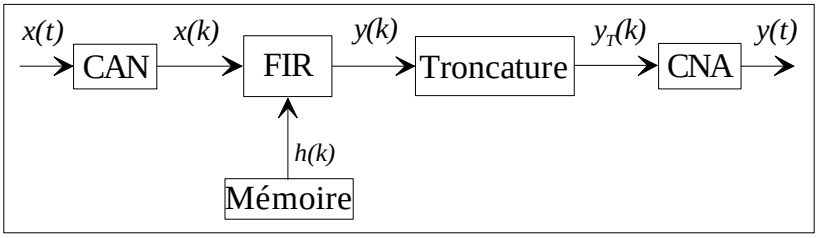

FIG. 4.1 – Signaux de la chaîne temporelle de simulation d'un canal SISO unidirectionnel.

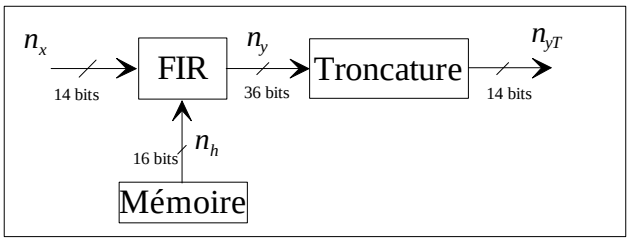

FIG. 4.2 – Diagramme de la chaîne numérique dans le domaine temporel pour un canal SISO unidirectionnel.

Avant de procéder à l'implémentation de cette architecture, il faut étudier chaque bloc qui la compose : le module qui effectue la convolution n'est autre qu'un filtre à réponse impulsionelle finie (*FIR - Finite Impulse Response*) que nous allons décrire. Le bloc Mémoire sera également expliqué. Le module de troncature, quant à lui, a déjà été détaillé dans le chapitre précédent et nous le réutiliserons.

4.1.1 Etude de la convolution

Comme pour l'approche fréquentielle, afin de simplifier la description des signaux, nous nous plaçons dans le cas simple d'un système linéaire invariant dans le temps.

4.1.1.1 Description de l'algorithme

Cette approche temporelle est basée sur le calcul du produit de convolution entre le signal d'entrée $x(t)$ et la réponse impulsionnelle du canal $h(t)$. La variable u étant la variable auxiliaire du produit de convolution, le signal de sortie $y(t)$ s'exprime par :

$$y(t) = h(t) * x(t) = \int_{-\infty}^{+\infty} h(u).x(t-u)du. \tag{4.1}$$

Comme le canal de propagation est assimilé à un système causal, c'est-à-dire $h(t) = 0$ pour $t < 0$, la définition de l'intégrale devient :

$$y(t) = \int_{0}^{+\infty} h(u).x(t-u)du. \tag{4.2}$$

En réalité, le signal n'est pas à durée finie. On ne garde que la partie du signal dont soit la valeur énergétique est de, par exemple, 99 % par rapport à la valeur maximale de l'énergie E, définie par :

$$E = \int_{0}^{+\infty} h^2(t)dt \tag{4.3}$$

soit l'amplitude est au-dessus d'un seuil, par exemple -30 dB par rapport à la valeur maximale de l'amplitude. Si cette partie du signal se situe à l'intérieur de la fenêtre temporelle $[0,\ W_t]$, on peut alors encore limiter l'intégrale et $y(t)$ devient :

$$y(t) = \int_{0}^{W_t} h(u).x(t-u)du. \tag{4.4}$$

Par ailleurs, le produit de convolution est réalisé à l'aide d'un filtre FIR. Le filtre FIR est un filtre numérique qui s'applique à un signal échantillonné. Ici, la période d'échantillonnage est $\Delta t = \frac{1}{f_s}$. On divise l'intervalle $[0,\ W_t]$ en N sous-intervalles de longueur Δt. Le nombre N ainsi obtenu représente la longueur du filtre FIR. On obtient alors les N points $t_k = k.\Delta t$ qui permettent de décrire l'intégrale sous la forme :

$$y(t) = \sum_{k=0}^{N-1} \int_{k.\Delta t}^{(k+1).\Delta t} h(u).x(t-u)du \tag{4.5}$$

$$\approx \Delta t \sum_{k=0}^{N-1} h(k.\Delta t).x(t-k.\Delta t). \tag{4.6}$$

Ainsi, pour toute valeur de $i \in \mathbb{N}$, on a :

$$y(i.\Delta t) \approx \Delta t \sum_{k=0}^{N-1} h(k.\Delta t).x((i-k).\Delta t), \tag{4.7}$$

si i $<$ k, on considère $x((i-k).\Delta t = 0$.

Cette représentation de l'échantillonnage de la fonction continue a son équivalent en discret qui s'exprime à l'aide des signaux quantifiés y_q, x_q et h_q que nous définirons ultérieurement. On obtient alors :

$$y_q(i) = \Delta t \sum_{k=0}^{N-1} h_q(k).x_q(i-k). \tag{4.8}$$

Chaque échantillon $x_q(i)$ est alors multiplié au fur et à mesure par les coefficients $h_q(0)$, $h_q(1)$, $\cdots$, $h_q(N-1)$. Ainsi, on obtient :

$$\text{1 terme : } y_q(0) \approx \Delta t.h_q(0)x_q(0) \tag{4.9}$$

$$\text{2 termes : } y_q(1) \approx \Delta t.\left[h_q(0)x_q(1) + h_q(1)x_q(0)\right] \tag{4.10}$$

$$\text{3 termes : } y_q(2) \approx \Delta t.\left[h_q(0)x_q(2) + h_q(1)x_q(1) + h_q(2)x_q(0)\right] \tag{4.11}$$

Pour i $\geq N$, N termes :

$$y_q(i) \approx \Delta t.\left[h_q(N-1)x_q(i-N+1) + h_q(N-2)x_q(i-N+2) + \cdots + h_q(0)x_q(i)\right]. \tag{4.12}$$

Cette dernière relation équivaut à la relation (4.8).

4.1.1.2 Etude du filtre FIR programmé

La chaîne temporelle complète sera simulée à partir des paramètres WLAN du tableau 2.2. En effet, la réponse impulsionnelle est la plus longue dans le cas « outdoor » du WLAN, qui correspond en réalité à un environnement indoor très grand comme un grand hangar ou un hall d'aéroport. Le produit de convolution de signaux continus se traduit en numérique par un filtre FIR. Ainsi, dans le pire des cas de l'environnement indoor du tableau 2.2, la longueur N du filtre FIR est de 64 points. D'un point de vue matériel, la société Xilinx fournit deux IP (définies au chapitre 2) pour réaliser un filtre numérique mais elles ne permettent pas un rafraîchissement dynamique des coefficients. Comme on souhaite rafraîchir les coefficients du filtre numérique avec une période définie au chapitre 2, nous avons développé notre propre filtre FIR. Compte tenu de la résolution des convertisseurs analogiques-numériques et numériques-analogiques présents sur la carte de prototypage choisie, notre filtre a une résolution de 14 bits pour les données d'entrée et les données

de sortie. Le filtre a une résolution de 16 bits pour les coefficients représentant la réponse impulsionnelle du canal, de façon à avoir une précision comparable à celle de l'entrée.

De plus, un filtre FIR 64 a besoin de 64 multiplieurs. Si on souhaite simuler par exemple un canal MIMO 2×2, il faudrait 4×64 = 256 multiplieurs, or le composant Virtex-4 utilisé n'en contient que 192. Donc, on a décidé de diviser par deux le nombre de multiplieurs utilisés, et de les faire fonctionner à 100 MHz. Comme les données d'entrée sont lues à $f_s = 50$ MHz, chaque multiplieur du filtre FIR est utilisé deux fois pendant deux cycles d'horloge de 100 MHz.

Par ailleurs, le taux d'occupation du composant pour un filtre FIR 64 peut être connu après avoir effectué les opérations de synthèse, de mapping et de placement-routage à partir du programme écrit en VHDL. Dans le tableau 4.1, on ne montre que l'occupation d'un seul filtre FIR 64 pris à part et on peut remarquer que les circuits logiques ne prennent pas énormément de place et qu'il n'y a bien que 32 multiplieurs d'utilisés. La fréquence maximale de fonctionnement du FIR, donnée par l'outil après placement routage, est de 119 MHz, ce qui est suffisant pour faire fonctionner les multiplieurs à deux fois la fréquence d'échantillonnage.

Nombre de « Slices »	1965 sur 15360	12 %
Nombre de LUT	2625 sur 30720	8 %
Nombre de blocs RAM	32 sur 192	16 %
Nombre de bascules	2168 sur 30720	7 %
Nombre de « XtremeDSP »	32 sur 192	16 %

TAB. 4.1 – Taux d'occupation du composant pour un seul filtre FIR 64.

4.1.2 Etude du bloc Mémoire

Le bloc Mémoire de la figure 4.2 sert à stocker les profils de réponses impulsionnelles dont on se servira pour alimenter les coefficients du filtre FIR.

4.1.2.1 Volume de données

Les échantillons des profils de réponses impulsionnelles utilisés sont codés sur 16 bits de façon à avoir une précision proche de la précision des valeurs d'entrée. Dans notre cas, une réponse impulsionnelle est représentée par 64 échantillons ou mots de 16 bits. En prenant, par exemple, un canal MIMO 2×2 unidirectionnel, comme celui de la figure 1.10, on a 4 canaux SISO unidirectionnels à simuler, donc on a besoin de 4 profils de réponses impulsionnelles. Si on a besoin de 4 profils de réponses impulsionnelles, on rajoute un mot à chaque profil de façon à indiquer le filtre FIR de destination du profil, car nous verrons plus loin que plusieurs blocs de mémoire sont utilisés pour charger un profil dans un filtre

FIR. On doit alors transmettre (65 × 4) mots de 16 bits, soit 260 mots de 16 bits donc environ 4160 bits à transmettre ou 520 octets.

Les profils contenant ces échantillons de 16 bits sont stockés dans un fichier texte sur le disque dur d'un ordinateur qui sera lu pour charger le bloc Mémoire. Comme nous l'avons vu au chapitre précédent, le bus PCI étant le plus rapide des bus disponibles, nous l'utilisons pour la lecture du fichier contenant les profils de réponses impulsionnelles. Comme nous avons vu au chapitre 2 que, dans le pire cas WLAN, la fréquence de rafraichissement $f_{raf_{WLAN}}$ est de 462,96 Hz, la période de rafraîchissement est donc d'environ 2,16 ms pendant laquelle on doit rafraîchir l'ensemble des 4 profils (pour notre exemple), c'est-à-dire 520 octets, ce qui fait un débit de :

$$\frac{520}{2,16.10^{-3}} = 235 \text{ ko/s}. \tag{4.13}$$

Par ailleurs, comme un canal SISO correspond ici à un profil de (64 × 16) bits, soit 128 octets, le débit de 30 Mo/s permettrait, pendant le temps de rafraîchissement de 2,16 ms, le chargement de 530 canaux SISO.

4.1.2.2 Chargement du bloc Mémoire

La procédure de chargement des profils est sensiblement la même que celle décrite dans le chapitre précédent pour l'approche fréquentielle. Elle comprend l'écriture du contenu du fichier de profils de réponses impulsionnelles sur le bus PCI, puis, la lecture par le Spartan-II et le transfert de ces profils vers le Virtex-4 à l'aide de l'IP HostInterface. Cette IP effectue ensuite le chargement des profils dans la FIFO. Cependant, la lecture de la FIFO pour le chargement des profils est différente car la distribution dans les blocs de mémoire ne se fait pas selon la même architecture, comme nous pouvons le voir sur la figure 4.3.

Notre FIR 64 contient 32 multiplieurs, comme nous l'avons vu précédemment, il faut donc 32 échantillons (ou coefficients) de profils disponibles en même temps pour être multipliés par les échantillons d'entrée. Nous avons donc besoin de 32 blocs de mémoire RAM. Nous pouvons remarquer qu'un profil est réparti sur les 32 blocs « RAM » à raison de deux coefficients par bloc. De façon à respecter le temps de rafraîchissement des profils, un deuxième profil est réparti de la même manière sur les 32 blocs « RAM », chacun contient donc au final 4 coefficients. Le signal 'Selector', écrit sur 5 bits, contrôle le démultiplexeur qui sélectionne l'un des 32 blocs « RAM » pour autoriser l'écriture des coefficients. Puis, le signal 'profil_selin', en scrutant le MSB à '1' du signal de lecture 'addr_r' (soit 'addr_r(1)'), sait quel profil est actif, donc utilisé par le filtre FIR. Alors, le signal 'profil_selout' positionne correctement le MSB à '1' du signal d'écriture 'addr_w' pour que les nouveaux coefficients soient écrits aux adresses de l'autre profil. Lorsque les

32 blocs sont remplis par un profil, le signal de lecture 'addr_r' est complémenté pour passer au profil suivant.

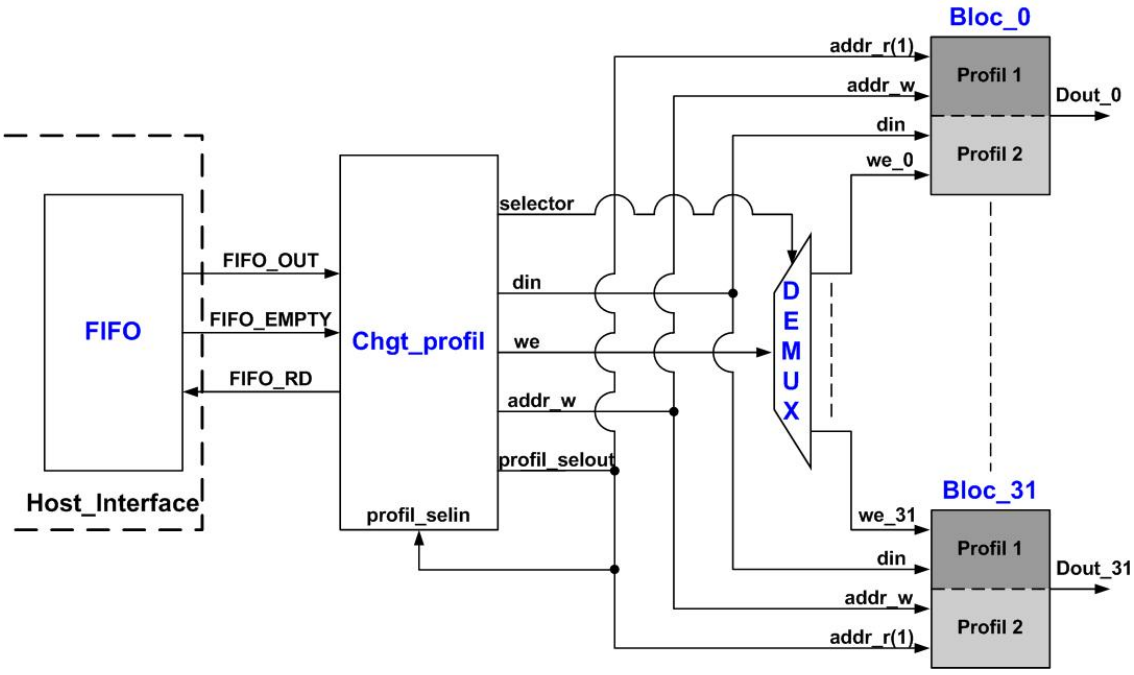

FIG. 4.3 – Chargement des profils de réponses impulsionnelles pour un canal SISO

La lecture et l'écriture des blocs « RAM » étant indépendantes, il est ainsi possible d'écrire des nouveaux coefficients alors que le filtre FIR continue de lire avec les anciens coefficients. Les 32 blocs « RAM » sont ainsi chargés et chaque sortie 'Dout_n', avec $n = 1, 2, \cdots, 31$, est dirigée en continu vers un multiplieur réel où les coefficients sont multipliés aux échantillons du signal d'entrée contenus dans un registre à décalage. Pendant 20 ns, le même multiplieur réel doit être utilisé deux fois pour multiplier chaque fois un coefficient avec un échantillon du signal d'entrée. Si le LSB du signal 'addr_r' vaut '0' alors un coefficient est utilisé pour la multiplication, si le LSB de 'addr_r' vaut '1' alors l'autre coefficient du même profil est utilisé. Puis, les échantillons d'entrée sont décalés. Le LSB de 'addr_r' est en fait un signal périodique de période deux fois la période d'horloge, c'est-à-dire la période d'échantillonnage. Les profils sont tous rechargés avec la période de rafraîchissement.

4.2 Mise en œuvre de la chaîne dans le domaine temporel

Pour la réalisation de la chaîne temporelle, nous réutilisons les signaux de tests présentés au chapitre précédent.

4.2.1 Quantification du signal d'entrée et de la réponse impulsionnelle

D'une part, la quantification du signal d'entrée a été expliquée au chapitre précédent, par la relation (3.43). On obtient alors les échantillons $n_x(k)$ de la relation (3.45) et les échantillons du signal quantifié $x_q(i)$ tels que :

$$x_q(i) = \Delta x.n_x(i) \approx x(i\Delta t). \tag{4.14}$$

D'autre part, seulement pour le test du filtre FIR, on utilise, pour représenter le canal, le signal réel gaussien $h(t)$ tel que :

$$h(t) = h_m e^{-\frac{(t-m_h)^2}{2\sigma_h^2}}, \ \forall\ t \in \mathbb{R} \tag{4.15}$$

Comme on souhaite obtenir le signal de sortie $y(t)$ de façon à ce qu'il soit dans la fenêtre temporelle W_t (dont la valeur est différente de celle de l'approche fréquentielle du chapitre précédent, comme on peut le voir dans le tableau 2.2), on fixe à nouveau sa moyenne $m_y = \frac{W_t}{2}$ et son écart type $\sigma_y = \frac{m_y}{4}$ car la gaussienne est considérée nulle au-delà de $4\sigma_y$. Ainsi, compte tenu des expressions (3.38) et (3.39), on choisit les valeurs de m_x, m_h, σ_x et σ_h. En utilisant (3.41), on calcule h_m.

Pour obtenir le facteur de quantification Δh des valeurs de $h(t)$ sur $M_h = 16$ bits, on utilise :

$$\Delta h = \frac{h_m}{2^{M_h-1}}, \tag{4.16}$$

Ainsi, à la place du vecteur ligne $\boldsymbol{h} = [h(0),\ h(\Delta t),\ \ldots,\ h((N-1)\Delta t)]$ où $N = 64$ des valeurs discrètes de $h(t)$, on obtient le vecteur ligne $\boldsymbol{n_h}$ des valeurs discrètes quantifiées sur 16 bits tel que :

$$\boldsymbol{n_h} = \text{round}\left\{\frac{\boldsymbol{h}}{\Delta h}\right\} \tag{4.17}$$

et un vecteur quantifié $\boldsymbol{h_q}$ qui lui correspond :

$$\boldsymbol{h_q} = \Delta h.\boldsymbol{n_h} \approx \boldsymbol{h}. \tag{4.18}$$

A la place du vecteur $\boldsymbol{h}$ formé par N échantillons, le filtre FIR reçoit comme coefficients les éléments du vecteur $\boldsymbol{n_h}$, vecteur à peu près proportionnel au vecteur $\boldsymbol{h}$ formé avec N échantillons de la réponse impulsionnelle discrétisée.

4.2.2 Filtre FIR 64

D'un point de vue numérique, on se réfère à la figure 4.2. On peut alors dire que le filtre FIR fait le produit de convolution entre le vecteur des échantillons d'entrée, qui est donc

un vecteur infini, et le vecteur des échantillons de la réponse impulsionnelle. Ainsi, on a :

$$\boldsymbol{n_y} = \boldsymbol{n_x} * \boldsymbol{n_h}, \tag{4.19}$$

que l'on peut exprimer aussi par une relation similaire à la relation (4.8) :

$$n_y(i) = \sum_{k=0}^{N-1} n_h(k).n_x(i-k), \forall\ i\ \in\ \mathbb{N} \tag{4.20}$$

En effet, il y a seulement N échantillons $n_h(k)$, avec $0 \leq k \leq N-1$, tandis que $n_x(i-k)$ est défini $\forall\ i\ \in\ \mathbb{N}$, i $\geq$ k. Pour i $<$ k, on a $n_x(i-k) = 0$.

Ce produit de convolution peut être illustré par la figure 4.4 qui montre l'architecture d'un filtre FIR pour un canal SISO unidirectionnel.

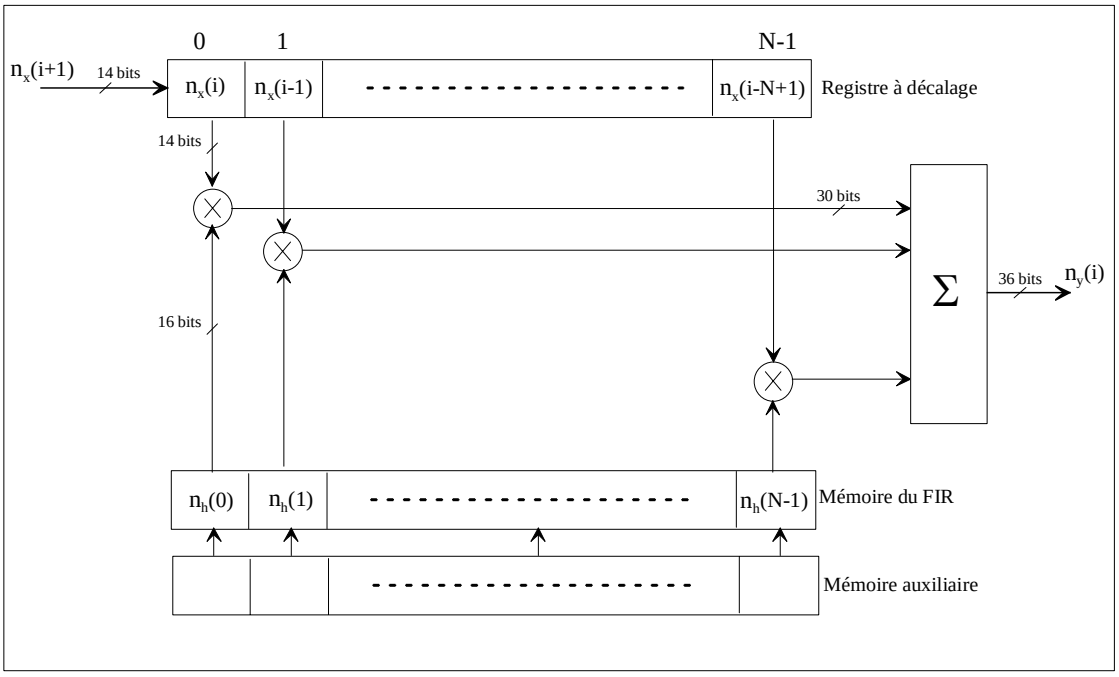

FIG. 4.4 – Architecture d'un filtre FIR pour un canal SISO unidirectionnel.

On peut distinguer les valeurs des échantillons du signal d'entrée quantifiés $n_x(i)$ dans le registre à décalage. Chaque échantillon est alors multiplié au fur et à mesure par les valeurs des coefficients $n_h(0), n_h(1), \cdots, n_h(N-1)$ contenus dans la mémoire. Une mémoire auxiliaire contient les valeurs des coefficients de la réponse impulsionnelle suivante.

Comme on sait que les signaux quantifiés ont pour expression :

$$x_q(i) = \Delta x.n_x(i) \approx x(i\Delta t) \tag{4.21}$$

et

$$h_q(k) = \Delta h.n_h(k) \approx h(k\Delta t) \quad \forall\ k\ \in\ \mathbb{N}\ \text{ et } 0 \leq k \leq\ N-1, \tag{4.22}$$

alors on peut retrouver les résultats réels de la sortie du filtre FIR, $n_y(i)$, qui sont des nombres *proportionnels* aux échantillons $y_q(i\Delta t)$:

$$y_q(i) = \Delta t \Delta x \Delta h \sum_{k=0}^{N-1} n_h(k).n_x(i-k) \quad \forall\ i\ \in\ \mathbb{N} \tag{4.23}$$

$$= \Delta t \Delta x \Delta h.n_y(i). \tag{4.24}$$

4.2.3 Troncature et convertisseur numérique-analogique

Les échantillons $n_y(i)$ de la sortie d'un filtre FIR sont codés sur 36 bits. Ces 36 bits sont dus aux 32 multiplieurs 14 bits $\times$ 16 bits dont les résultats sur 30 bits sont additionnés sur 5 étages d'additionneurs. De plus, il faut ajouter un accumulateur pour sommer les résultats des deux cycles de 32 multiplications et de 5 étages d'additions, ce qui fait que les échantillons de la sortie d'un filtre FIR 64 sont codés sur 36 bits. Ces échantillons sont ensuite tronqués sur 14 bits pour pouvoir entrer dans le convertisseur numérique-analogique, donc 22 bits sont éliminés. En adaptant la troncature brutale expliquée au chapitre 3, on obtient :

$$n_{y_T}(i) = \left\lfloor \frac{n_y(i)}{2^{22}} \right\rfloor \tag{4.25}$$

$$\text{soit :} \quad n_{y_T}(i) \approx \frac{n_y(i)}{2^{22}}. \tag{4.26}$$

Le signal quantifié tronqué $y_{q_T}(i)$ peut s'exprimer à l'aide de la relation (4.24) :

$$y_{q_T}(i) = \Delta t \Delta x \Delta h.n_{y_T}(i). \tag{4.27}$$

Ainsi, lorsque nous obtenons les échantillons $n_y(i)$ en sortie du filtre FIR, après la chaîne temporelle que nous venons de décrire, il est nécessaire de les multiplier par les quantifications de t, de x et de h pour retrouver les valeurs des échantillons quantifiés, comparables à ce que l'on doit obtenir en théorie.

En utilisant la relation (4.26), la relation (4.27) devient :

$$y_{q_T}(i) = \Delta t \Delta x \Delta h \left\lfloor \frac{n_y(i)}{2^{22}} \right\rfloor. \tag{4.28}$$

De plus, comme la quantification du CNA est la même que celle du CAN, on a :

$$y_s(i) = \Delta y.n_{y_T}(i) = \Delta x.n_{y_T}(i) \tag{4.29}$$

et avec les relations (3.82) puis (4.24), on obtient :

$$y_s(i) = \Delta x.n_{y_T}(i) \approx \Delta x.\frac{n_y(i)}{2^{22}} \tag{4.30}$$

$$\approx \frac{\Delta x}{2^{22}}\frac{y_q(i)}{\Delta t \Delta x \Delta h} \tag{4.31}$$

$$\approx \frac{y_q(i)}{2^{22}\Delta t \Delta h}, \tag{4.32}$$

ce qui donne :

$$y_q(i) \approx 2^{22}\Delta t \Delta h.y_s(i). \tag{4.33}$$

Par conséquent, à partir du signal $y_s(i)$ fourni par la sortie du CNA, il faut utiliser le facteur multiplicatif $2^{22}\Delta t \Delta h$ pour retrouver la vraie valeur du signal $y(t)$. Dans le cas de la troncature glissante, le facteur multiplicatif sera $2^{22+n_{LSB}}\,\Delta t \Delta h$.

Dans le cas où le signal de sortie aurait une faible amplitude qui n'exploiterait pas les 14 bits avant d'entrer dans le CNA, il est possible d'amplifier par un facteur multiplicatif les vraies valeurs de $h(t)$ et ainsi amplifier le signal de sortie de façon à éviter la troncature glissante à la fin de la chaîne temporelle.

4.2.4 Résultats et précision de calcul

4.2.4.1 Simulation du filtre FIR 64

Le signal obtenu en sortie de la chaine complète, dont la mise en œuvre a été décrite dans la section précédente, est présenté à la figure 4.5, à gauche en échelle linéaire et à droite en échelle logarithmique par rapport à sa valeur maximale.

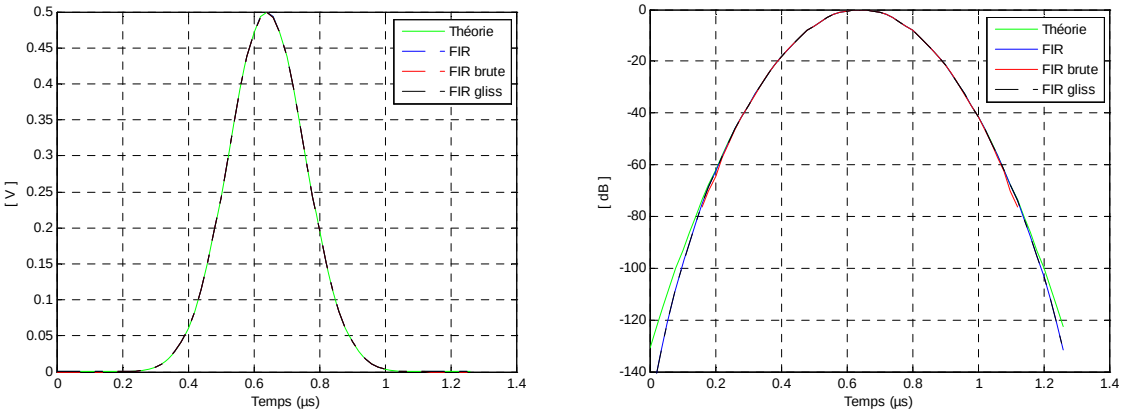

FIG. 4.5 – Tension de sortie de la chaîne temporelle en échelle linéaire et en échelle logarithmique.

De la même façon que pour l'approche fréquentielle, plusieurs signaux y sont en fait présentés : le tracé de la courbe « Théorie » montre le signal théorique tracé à partir de l'expression (3.32), c'est-à-dire le signal de sortie $y(t)$ duquel le signal de sortie calculé par la chaîne temporelle $y_c(t)$ doit se rapprocher. La courbe « FIR » montre la sortie du filtre FIR avant la troncature, ce sont donc les valeurs réelles sur 36 bits auxquelles on va appliquer la troncature nécessaire, brutale ou glissante. Il s'agit du cas qui doit être le plus proche de la théorie. La courbe « FIR brute » représente la sortie après la troncature brutale sur 14 bits et la courbe « FIR gliss » la sortie après la troncature glissante, détaillée au chapitre précédent.

Une nouvelle fois, on peut remarquer une excellente concordance entre les signaux calculés par le FPGA et le signal théorique. On détermine alors, avec les relations (3.88) et (3.89), l'erreur relative en pourcentage et le SNR en dB par rapport au signal gaussien théorique, pour les différents cas du signal de sortie et on peut ensuite tracer les courbes qui leur correspondent.

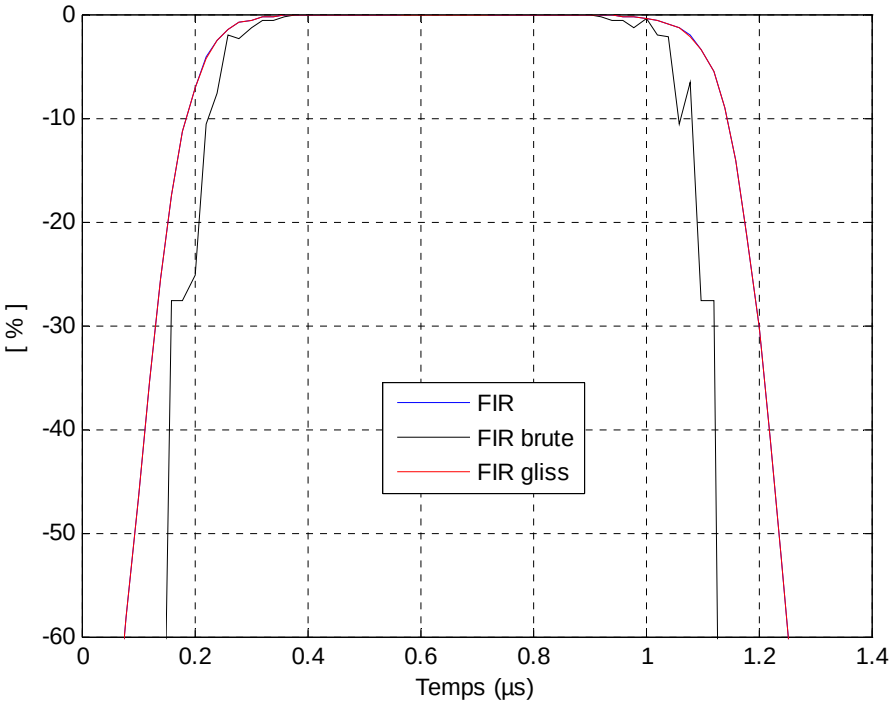

FIG. 4.6 – Erreur relative du signal de sortie de la chaîne temporelle.

Les figures 4.6 et 4.7 montrent l'erreur relative en pourcentage et le rapport signal sur bruit en dB du signal de sortie de la chaîne temporelle.

La légende de ces courbes est la même que pour les signaux de sortie c'est-à-dire : la courbe « FIR » montre l'erreur ou le SNR des résultats en sortie du FIR avant la

troncature, la courbe « FIR brute » représente ces données après la troncature brutale sur 14 bits et la courbe « FIR gliss » après la troncature glissante.

On peut faire les mêmes remarques sur les deux figures 4.6 et 4.7, c'est-à-dire, qu'en comparant ces figures avec la figure 4.5 de la tension en sortie, en échelle logarithmique, on peut remarquer que les plus grandes valeurs de tension en sortie, c'est-à-dire les valeurs supérieures à -40 dB par rapport à la valeur maximale de la tension, ont une erreur relative inférieure à 1 % en valeur absolue et un bon SNR. L'erreur peut être beaucoup plus importante mais seulement pour des petites valeurs de la tension de sortie.

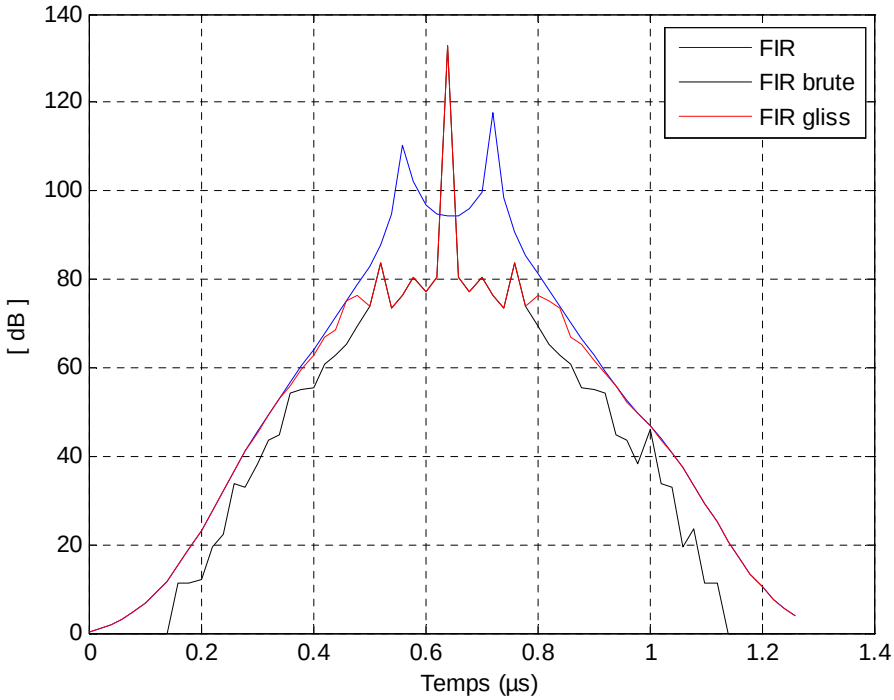

FIG. 4.7 – Rapport signal à bruit du signal de sortie de la chaîne temporelle.

De plus, on constate que l'erreur relative des échantillons de sortie obtenus par troncature glissante « FIR gliss » est tellement proche de l'erreur relative des échantillons du signal « FIR », obtenu en sortie du filtre FIR, qu'on ne peut pas les distinguer sur la figure 4.6. On peut donc conclure que la troncature glissante s'effectue avec très peu d'erreur à la sortie du filtre FIR, et qu'elle est meilleure que la troncature brutale envisagée au départ, surtout pour les faibles valeurs du signal de sortie.

Par ailleurs, pour les faibles valeurs de la tension du signal de sortie, l'erreur relative calculée pour chaque échantillon peut atteindre -100 %, c'est pourquoi les valeurs globales de l'erreur relative et du SNR des signaux de sortie, avant et après la troncature finale, sont également calculées avec les relations (3.90) et (3.91). Les résultats sont présentés dans le tableau 4.2 où on peut observer que les valeurs globales de l'erreur et du SNR sont

effectivement meilleures avec la troncature glissante. La sortie après la troncature glissante est globalement plus proche du signal théorique.

	Erreur (%)	SNR (dB)
Sortie FIR sans troncature	0.0105	79.56
Sortie FIR avec troncature glissante	0.0168	75.48
Sortie FIR avec troncature brutale	0.0305	70.32

TAB. 4.2 – Comparaison des valeurs globales de l'erreur relative et du SNR.

Par ailleurs, si l'on souhaite simuler un canal radio MIMO 2×2 unidirectionnel, quatre filtres FIR sont nécessaires. Afin de connaître le taux d'occupation du composant pour quatre filtres FIR et leur chargement de coefficients respectif, nous utilisons les outils logiciels de Xilinx. Après la synthèse, le placement et le routage, le tableau 4.3 montre le taux d'utilisation du composant programmable pour quatre filtres FIR de longueur 64 et leurs circuits additionnels utilisés pour le rechargement dynamique des coefficients des filtres FIR, c'est-à-dire les profils de réponses impulsionnelles du canal. La fréquence maximale de fonctionnement de cette configuration, c'est-à-dire la fréquence d'horloge, est de 130 MHz, selon l'outil de Xilinx.

Nombre de « Slices »	9135 sur 15360	59 %
Nombre de LUT	12877 sur 30720	41 %
Nombre de blocs RAM	129 sur 192	67 %
Nombre de bascules	9085 sur 30720	29 %
Nombre de « XtremeDSP »	128 sur 192	66 %

TAB. 4.3 – Taux d'occupation du composant programmable Virtex-IV SX35 pour 4 FIR 64

Un filtre FIR a 2 cycles de 32 multiplications et 5 cycles d'additions. Un cycle supplémentaire est nécessaire pour ajouter les résultats de chaque FIR. Ainsi, cela fait théoriquement 8 cycles de 100 MHz, donc 80 ns de latence pour les opérations de la partie numérique dans le domaine temporel. De plus, comme pour le domaine fréquentiel, il est nécessaire d'ajouter approximativement 38 ns de latence du CAN, et 17 ns de latence du CNA, indiquées sur leur documentation respective.

En résumé, quatre filtres FIR et les circuits environnants semblent bien s'accorder dans un seul FPGA, avec un temps de latence raisonnable d'environ 135 ns. Si on considère la célérité de l'onde $c = 3.10^8$ m/s, cette latence impose une distance minimale entre les antennes Tx et Rx augmentée de 40,5 m, ce qui correspond à la taille maximale de la plus petite cellule WLAN du tableau 2.2. Cette latence peut être mesurée à l'aide de l'outil Modelsim, comme expliqué au chapitre 3.

4.2.4.2 Expérimentation du filtre FIR 64 sur carte Nallatech

Les expérimentations sur carte ont été mises en œuvre avec la carte Nallatech décrite précédemment. Dans un premier temps, afin de vérifier le bon fonctionnement du filtre FIR, un seul canal SISO sera testé avec une seule fonction $h(t)$, sans rafraîchissement, ce qui correspond à la chaîne décrite à la figure 4.2. Les signaux de test $x(t)$ et $h(t)$ sont les gaussiennes présentées au chapitre précédent. Les échantillons quantifiés de la gaussienne temporelle d'entrée $x(t)$, contenus dans le programme VHDL, lui-même généré par l'exécution d'un programme MATLAB pour ce test, sont utilisés en entrée du filtre FIR 64. Les échantillons quantifiés de la gaussienne temporelle $h(t)$, représentant un profil de réponse impulsionnelle, sont stockés dans les blocs de mémoire RAM par leur initialisation, pour les besoins du test.

L'utilisation du logiciel Impact, outil de Xilinx, permet la programmation des deux composants programmables, comme nous l'avons vu au chapitre précédent, le Spartan-II pour la gestion de l'horloge et le Virtex-4 pour l'implémentation du filtre FIR.

La visualisation des sorties se fait avec un oscilloscope numérique. Le signal en entrée du filtre FIR 64 est visualisé via un CNA et le signal de sortie via le deuxième CNA disponible sur la carte Nallatech. La plate-forme de test est illustrée par la figure 4.8, où on peut remarquer la carte Nallatech en fonctionnement, les signaux de sortie sur l'écran de l'oscilloscope et les signaux simulés par l'outil Modelsim, sur l'écran de l'ordinateur.

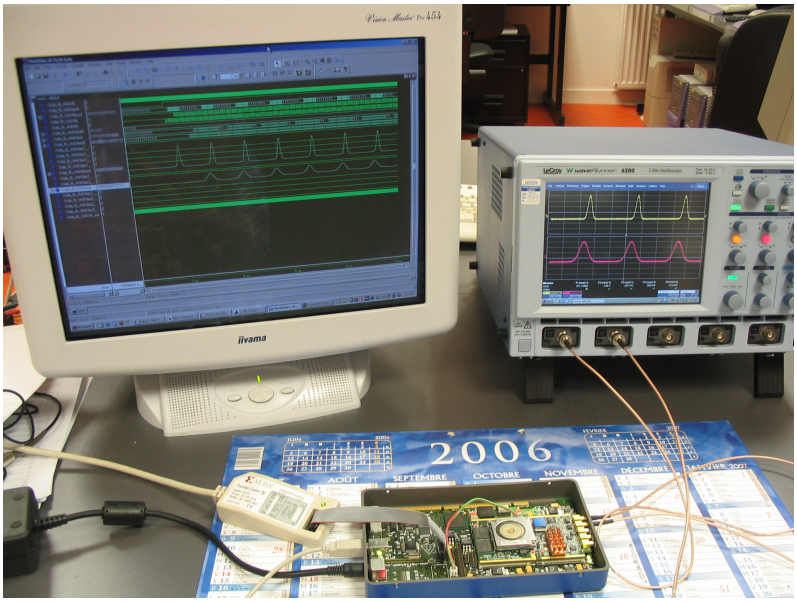

FIG. 4.8 – Implémentation sur carte Nallatech d'un filtre FIR 64 pour un canal SISO.

Pour valider notre architecture numérique, on observe alors les chronogrammes de sortie d'un filtre FIR 64, obtenus avec l'oscilloscope numérique comme on peut le voir

sur la figure 4.9. La différence de temps entre la valeur maximale du signal d'entrée et la valeur maximale du signal de sortie est la somme de la latence de l'architecture numérique et de la moyenne de la réponse impulsionnelle m_h. On mesure cette différence de temps par l'intervalle de temps, noté ΔX sur la figure 4.9 et on obtient 384 ns. Or, on connait la valeur de m_h, choisie comme indiqué à la section 3.2.1, on peut alors vérifier la latence du filtre FIR en pratique :

$$384 - 240 = 144 \text{ ns}, \tag{4.34}$$

au lieu de 135 ns, mesurée en simulation.

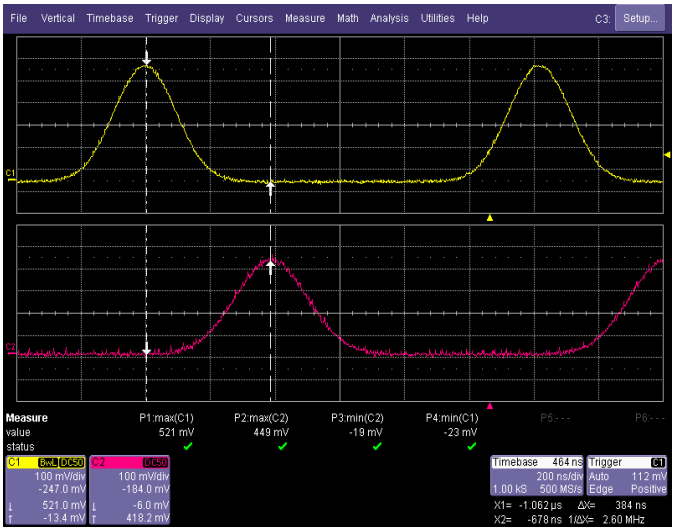

FIG. 4.9 – Test du filtre FIR 64 implémenté sur Virtex-4 (entrée en jaune et sortie en rose).

Nous pouvons donc dire que ce test répond à nos attentes.

4.3 Solution de substitution de l'approche fréquentielle

Pour les environnements « outdoor », afin de réduire la latence du bloc numérique étudié dans le chapitre précédent, et pour pallier au manque de précision dans le cas de réponses impulsionnelles très longues, nous avons envisagé une solution temporelle en considérant un filtre FIR 512, dont 32 coefficients seulement seraient utilisés. Ces 32 coefficients correspondent alors aux 32 trajets les plus significatifs d'un profil de réponse impulsionnelle, c'est-à-dire 32 trajets avec une amplitude assez grande et qui seraient déterminés selon le modèle de canal utilisé.

Par ailleurs, cette architecture est proposée pour le cas des applications UMTS (cf. tableau 2.2) ayant 40 MHz de fréquence d'échantillonnage, et donc de fréquence d'entrée des données. Comme seulement 32 coefficients sont nécessaires, les 32 multiplieurs travaillent

à 40 MHz également, contrairement à l'architecture à filtres FIR 64, où les multiplieurs devaient être deux fois plus rapides.

4.3.1 Architecture du filtre FIR 512 à 32 coefficients

La solution proposée consiste à stocker les valeurs de l'entrée dans des registres à décalage de façon à disposer, à un instant $t_i = i\Delta t$, de toutes les valeurs des échantillons du signal d'entrée $x(i\Delta t)$, $x((i-1)\Delta t)$, ..., $x((i-511)\Delta t)$. Cependant, la réalisation étant trop coûteuse en ressources, au lieu de stocker les valeurs d'entrée dans des registres à décalage, on les stocke dans autant de blocs RAM qu'il y a de multiplieurs. Ainsi, il est possible d'avoir tous les coefficients (les entrées retardées, en fait) en même temps pour effectuer les multiplications. La figure 4.10 présente l'architecture envisagée avec un filtre FIR 512 à 32 coefficients, pour un canal SISO.

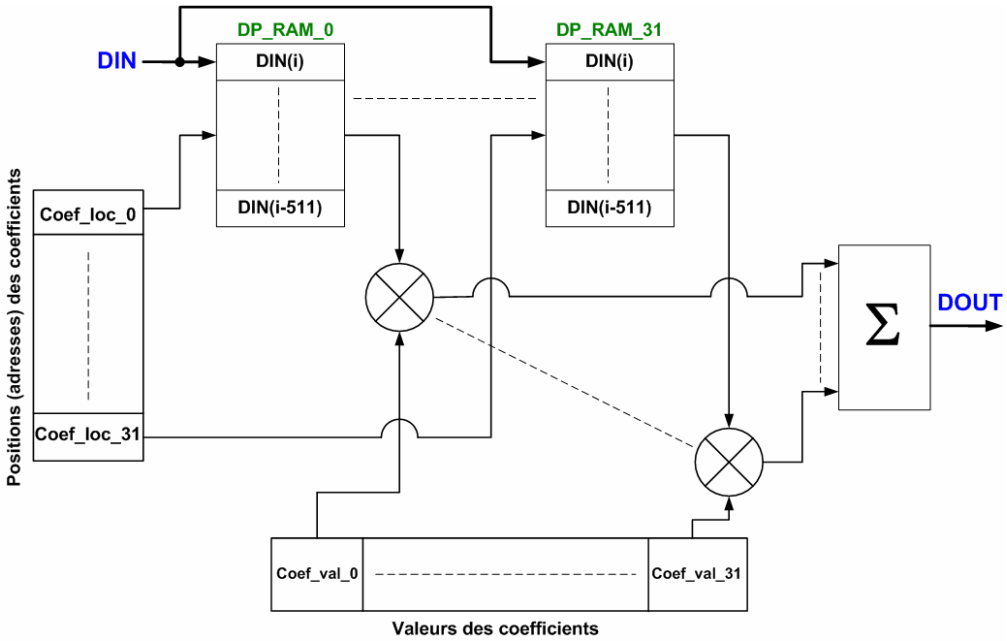

FIG. 4.10 – Architecture d'un filtre FIR 512 avec 32 coefficients pour un canal SISO.

D'une part, les blocs RAM ont tous la même entrée 'DIN' et contiennent exactement les mêmes échantillons du signal d'entrée. D'autre part, comme nous l'expliquerons plus tard, les coefficients sont stockés dans deux zones mémoires, l'une pour leur valeur et l'autre pour leur position. La position représente l'adresse du coefficient et indique la valeur d'entrée qui, en sortie du bloc RAM, sera multipliée par la valeur du coefficient. Par exemple, la valeur du coefficient 'coef_val_n', avec $0 \leq n \leq 31$, dont la position 'coef_loc_n' indique la valeur décimale '10', sera multipliée par le contenu de 'DIN$(i-10)$' qui se trouvera en sortie du bloc RAM.

La contribution de chacun des multiplieurs est ensuite additionnée pour fournir finalement la sortie 'DOUT', cette sortie sera représentée sur 35 bits, compte tenu des étages de multiplieurs et d'additionneurs utilisés.

4.3.2 Chargement des coefficients

La figure 3.3 reste valable pour ce filtre FIR 512 à 32 coefficients, à ceci près que les coefficients sont des mots de 25 bits stockés dans un fichier texte sur le disque dur de l'ordinateur. Ces 25 bits sont constitués de 16 bits représentant la valeur du coefficient et 9 bits représentant la position de son retard, ce qui fait 512 valeurs possibles de positions des retards. Les 25 bits des coefficients peuvent facilement transiter en même temps sur le bus PCI puisqu'il s'agit d'un bus de 32 bits.

Comme les multiplieurs doivent avoir accès, à un même instant t, à toutes les valeurs des 32 coefficients du FIR et à toutes les valeurs de l'entrée correspondant aux positions des retards des 32 coefficients, les deux parties des coefficients nécessitent d'utiliser une zone mémoire distincte pour chacune que l'on appelera « Coef_value_storage » pour la valeur des coefficients et « Coef_loc_storage » pour leur position, chaque zone mémoire contenant 32 mémoires doubles-ports. De plus, comme le stockage des échantillons d'entrée utilisent la plupart des blocs « RAM » doubles-ports, ils n'en restent pas suffisamment pour réaliser les deux zones de mémoire citées ci-dessus. On utilise donc de la mémoire dite « RAM distribuée », disponible dans le composant programmable choisi c'est-à-dire le Virtex-4 SX 35. Cette « RAM » se sert des slices pour réaliser des mémoires doubles-ports au lieu des blocs « RAM », comme dans le cas des filtres FIR 64 ou de la chaîne fréquentielle. Cependant, le fonctionnement est identique.

La procédure de chargement des coefficients est proche de celle du filtre FIR 64, sauf qu'il y a deux zones de mémoire à remplir mais qui fonctionnent de la même manière. Nous pouvons voir l'architecture de ce chargement des mémoires sur la figure 4.11.

De façon à réduire le temps de chargement des coefficients, le chargement des valeurs et des positions des coefficients doit se faire simultanément. Sur la figure 4.11, le bus de données en sortie du module « Chgt_profil » est un bus de 25 bits dont les 16 bits de poids forts correspondent à la valeur du coefficient et les 9 bits de poids faibles correspondent à la position du coefficient. Ces deux parties sont dirigées vers leur zone mémoire respective.

Le détail de la zone de mémoire « Coef_value_storage » est donné à la figure 4.12, sachant que la zone « Coef_loc_storage » a la même architecture.

Contrairement aux blocs « RAM » du filtre FIR 64 qui devait contenir deux valeurs de coefficients par profil, ici une seule valeur de coefficient ou une seule position par profil est suffisante puisqu'on n'a besoin que de 32 coefficients dans 32 « RAM ».

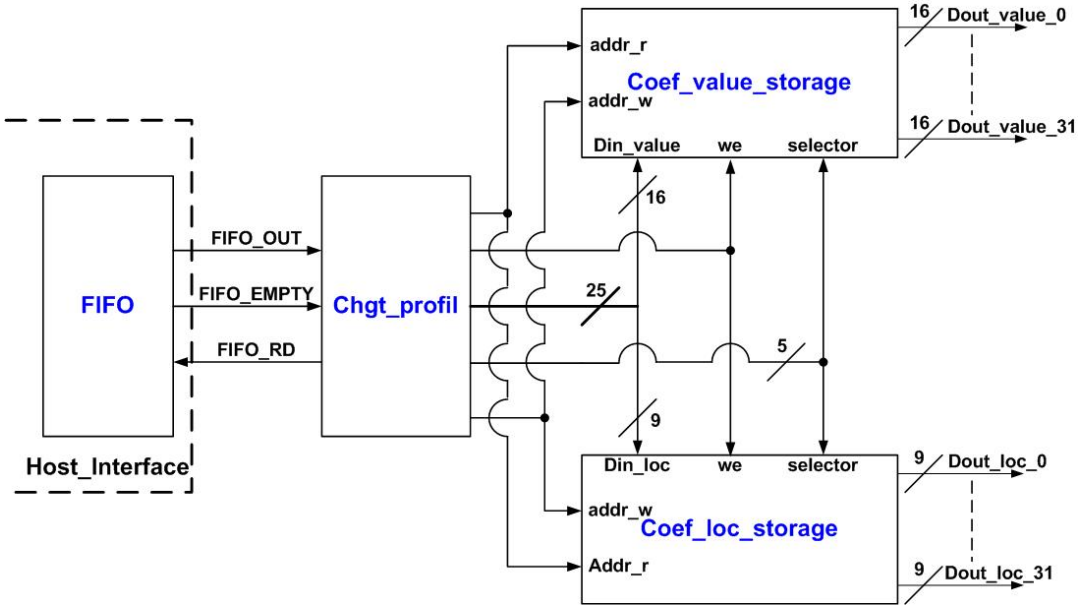

FIG. 4.11 – Chargement des profils de réponses impulsionnelles pour un canal SISO.

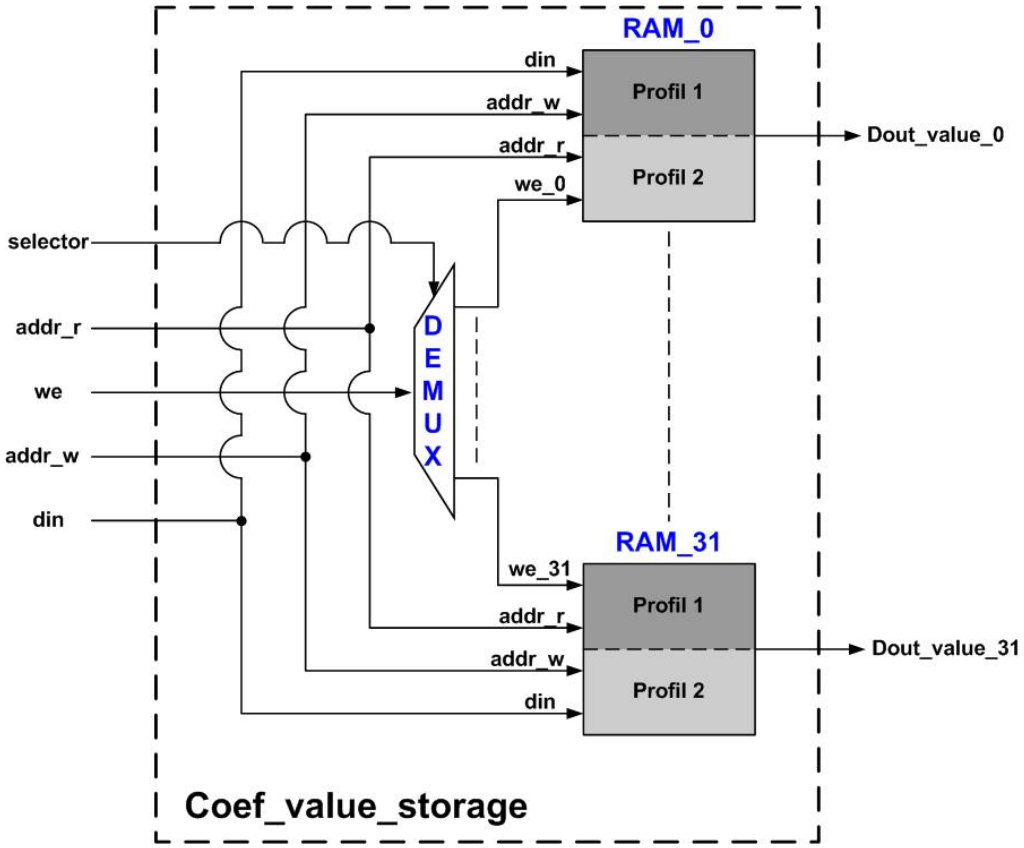

FIG. 4.12 – Détail de l'architecture du chargement des profils de réponses impulsionnelles.

Par contre, on a toujours deux profils répartis sur les 32 « RAM » de façon à fournir ces coefficients de manière ininterrompue et assurer ainsi le rafraîchissement des profils de réponses impulsionnelles.

Lors du chargement d'un profil, il faut dans un premier temps connaître le profil utilisé par le filtre FIR à l'aide de la valeur de l'adresse de lecture 'addr_r'. L'adresse d'écriture 'addr_w' sera alors la valeur complémentée de l'adresse de lecture. Le signal 'selector', écrit sur 5 bits, contrôle le démultiplexeur et permet de sélectionner une des 32 « RAM » pour l'écriture à l'adresse sélectionnée par 'addr_w'. En activant le signal d'autorisation d'écriture 'we', chaque nouveau coefficient est écrit (sa valeur et sa position) dans les « RAM » adéquates, puis, en incrémentant le signal 'selector', on passe à l'écriture du coefficient suivant, ce qui termine le chargement d'un profil. Ce nouveau profil chargé sera utilisé lorsque l'adresse de lecture 'addr_r' sera complémentée.

Comme les données d'entrée sont stockées dans une mémoire double-port, on peut ensuite venir lire la valeur de l'échantillon du signal d'entrée correspondant à la valeur de la position du retard, pour multiplier l'entrée avec le coefficient de la réponse impulsionnelle.

4.3.3 Résultats et précision de calcul

Afin de tester cette architecture numérique, nous avons utilisé, en entrée, la gaussienne de test définie par (3.30). Dans un premier temps et pour simplifier les tests, nous avons décidé de positionner deux coefficients non nuls parmi les 32 coefficients du filtre FIR 512. De façon à tester l'architecture avec des valeurs aussi bien positives que négatives, on souhaite avoir la valeur maximale de $y_s(t)$ égale à 0,5 V et la valeur minimale à - 0,25 V. Nous choisissons alors les paramètres de la gaussienne d'entrée, d'une part, et du signal de sortie composé de deux gaussiennes, d'autre part, de la même façon que celle présentée à la section 3.2.1. Ainsi, la première valeur non-nulle des échantillons $\boldsymbol{n_h}$ de $h(t)$ est positionnée comme étant le premier retard introduit, donc à l'indice 0, avec pour valeur 2^{14} de façon à avoir une valeur maximale de 0,5 V pour le signal en sortie du CNA. Le second coefficient est alors positionné comme étant à la moitié de la fenêtre temporelle, donc à l'indice 256, avec comme valeur -2^{13}. Pour le filtre FIR 512 à 32 coefficients, ces deux coefficients non nuls simulent un modèle à deux trajets dont les retards sont 0 et $256.\Delta t = 6,4\ \mu$s .

4.3.3.1 Simulation du filtre FIR 512 à 32 coefficients

Le signal obtenu en sortie de cette architecture est présenté à la figure 4.13, à gauche en échelle linéaire et à droite en échelle logarithmique par rapport à sa valeur maximale.

De la même façon que précédemment, plusieurs signaux y sont en fait présentés : le tracé de la courbe « Théorie » montre le signal théorique tracé à partir de l'expression de la somme d'une gaussienne positive et d'une gaussienne négative, compte tenu des coefficients choisis. La courbe « FIR » montre la sortie du filtre FIR avant la troncature,

ce sont donc les valeurs réelles sur 35 bits, comme expliqué à la section 4.3.1, auxquelles on va appliquer la troncature nécessaire, brutale ou glissante. La courbe « FIR brute » représente la sortie après la troncature brutale sur 14 bits et la courbe « FIR gliss » la sortie après la troncature glissante, détaillée au chapitre précédent.

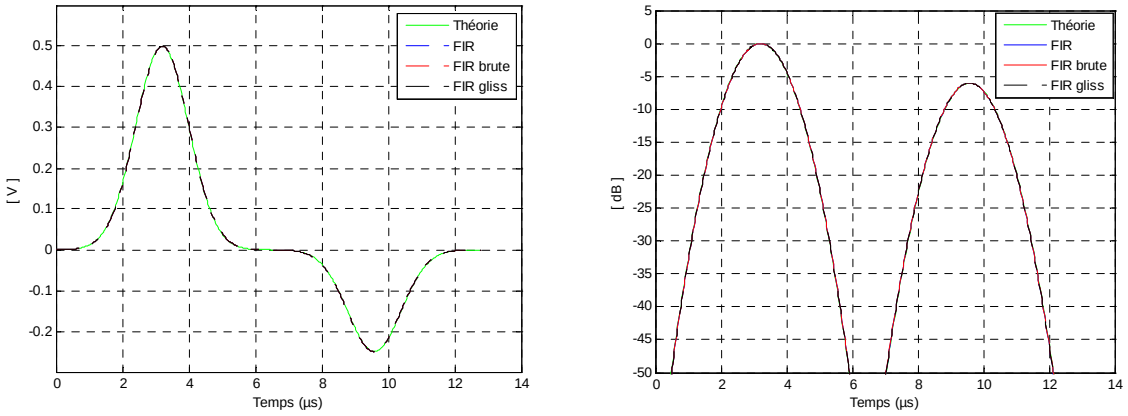

FIG. 4.13 – Tension de sortie de la chaîne complète avec le filtre FIR 512 à 32 coefficients.

Ici encore, on peut remarquer une excellente concordance entre les signaux calculés par le FPGA et le signal théorique. On détermine alors l'erreur relative en pourcentage et le SNR en dB par rapport au signal gaussien théorique avec les relations (3.88) et (3.89) et pour les différents cas du signal de sortie. On peut ensuite tracer les courbes qui leur correspondent.

Les figures 4.14 et 4.15 montrent l'erreur relative en pourcentage et le rapport signal sur bruit en dB du signal de sortie de la chaîne temporelle.

La légende de ces courbes est la même que pour les signaux de sortie c'est-à-dire que la courbe « FIR » montre l'erreur ou le SNR des résultats en sortie du FIR avant la troncature, la courbe « FIR brute » représente ces données après la troncature brutale sur 14 bits et la courbe « FIR gliss » après la troncature glissante.

En comparant ces deux figures avec la figure 4.13 de la tension de sortie en dB, on peut remarquer que si les valeurs de la tension de sortie sont supérieures à -50 dB par rapport à la valeur maximale de la tension de sortie, l'erreur reste inférieure à 5 % en valeur absolue et on a un bon SNR supérieur à 30 dB. L'erreur peut être beaucoup plus importante mais seulement pour des petites valeurs de la tension du signal de sortie lorsque le signal de sortie est proche de 0.

De plus, on constate que l'erreur relative des échantillons de sortie obtenus par troncature glissante « FIR gliss » est tellement proche de l'erreur relative des échantillons du signal « FIR », obtenu en sortie du filtre FIR, qu'on ne peut pas les distinguer sur les figures 4.14 et 4.15. On peut donc conclure que la troncature glissante s'effectue avec très peu d'erreur à la sortie du filtre FIR.

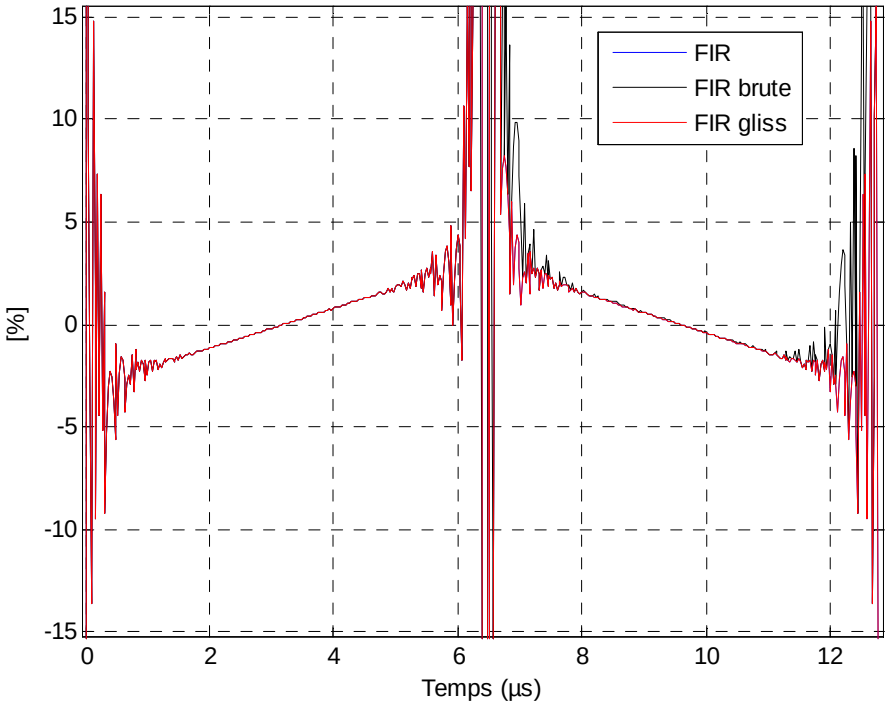

FIG. 4.14 – Erreur relative du signal de sortie de la chaîne temporelle.

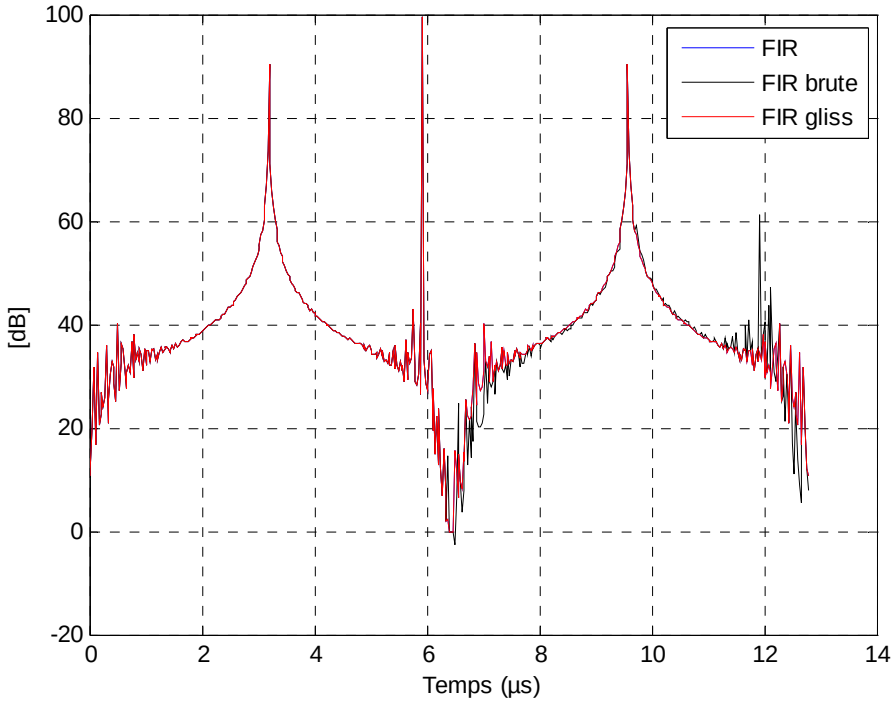

FIG. 4.15 – Rapport signal à bruit du signal de sortie de la chaîne temporelle.

Par ailleurs, pour les faibles valeurs du signal de sortie $y_s(t)$, l'erreur relative calculée pour chaque échantillon peut atteindre -100 %, c'est pourquoi les valeurs globales de l'erreur relative et du SNR des signaux de sortie, avant et après la troncature finale, sont également calculées avec les relations (3.90) et (3.91). Les résultats sont présentés dans le tableau 4.4. On peut observer que les valeurs globales de l'erreur et du SNR sont effectivement meilleures avec la troncature glissante, mais que l'amélioration apportée par la troncature glissante n'est pas significative dans ce cas.

	Erreur (%)	SNR (dB)
Sortie FIR sans troncature	0.5527	45.1505
Sortie FIR avec troncature glissante	0.5527	45.1505
Sortie FIR avec troncature brutale	0.5529	45.1471

TAB. 4.4 – Comparaison des valeurs globales de l'erreur relative et du SNR.

De plus, si l'on souhaite simuler un canal radio MIMO 2×2 unidirectionnel, quatre filtres FIR sont nécessaires. Après la synthèse, le placement et le routage, le tableau 4.5 montre le taux d'utilisation du composant programmable pour quatre filtres FIR de longueur 512 à 32 coefficients choisis, et leurs circuits additionnels utilisés pour le rechargement dynamique des coefficients des filtres FIR.

Nombre de « Slices »	10480 sur 15360	68%
Nombre de LUT	14915 sur 30720	48%
Nombre de blocs RAM	129 sur 192	67%
Nombre de bascules	7750 sur 30720	25%
Nombre de « XtremeDSP »	130 sur 192	67%

TAB. 4.5 – Taux d'occupation du composant programmable Virtex-IV SX35 pour 4 filtres FIR 512 à 32 coefficients

On remarque que cette solution est très gourmande en ressources car elle nécessite d'obtenir les 32 coefficients en même temps pour faire les multiplications, donc d'avoir toujours les 512 échantillons du signal d'entrée pour pouvoir positionner le coefficient à la bonne valeur du retard.

En utilisant Modelsim, on constate que la latence pour ces 4 filtres FIR 512 à 32 coefficients est de 150 ns, ce qui correspond, en considérant la célérité de l'onde, à une distance minimale entre un réseau d'antennes en émission et un réseau d'antennes en réception de 45 m, convenant largement au milieu outdoor. Cette latence est très faible par rapport à celle de l'architecture fréquentielle évoquée au chapitre précédent.

4.3.3.2 Expérimentation du filtre FIR 512 à 32 coefficients sur carte Nallatech

Pour vérifier le bon fonctionnement, la latence et la précision des calculs d'un filtre FIR 512 à 32 coefficients, on implémente l'architecture proposée sur le FPGA Virtex-4 de la carte Nallatech, de la même façon que celle décrite pour l'expérimentation du filtre FIR 64. Pour ce test, le signal d'entrée et les 32 coefficients sont choisis de la même façon que pour la simulation. Les positions et les valeurs des coefficients sont placées dans la mémoire RAM comme indiqué précédemment.

La sortie d'un CNA montre le signal d'entrée quantifié. La sortie du deuxième CNA permet de visualiser la sortie du filtre FIR 512 à 32 coefficients, après la troncature brutale, comme nous pouvons le voir sur la figure 4.16.

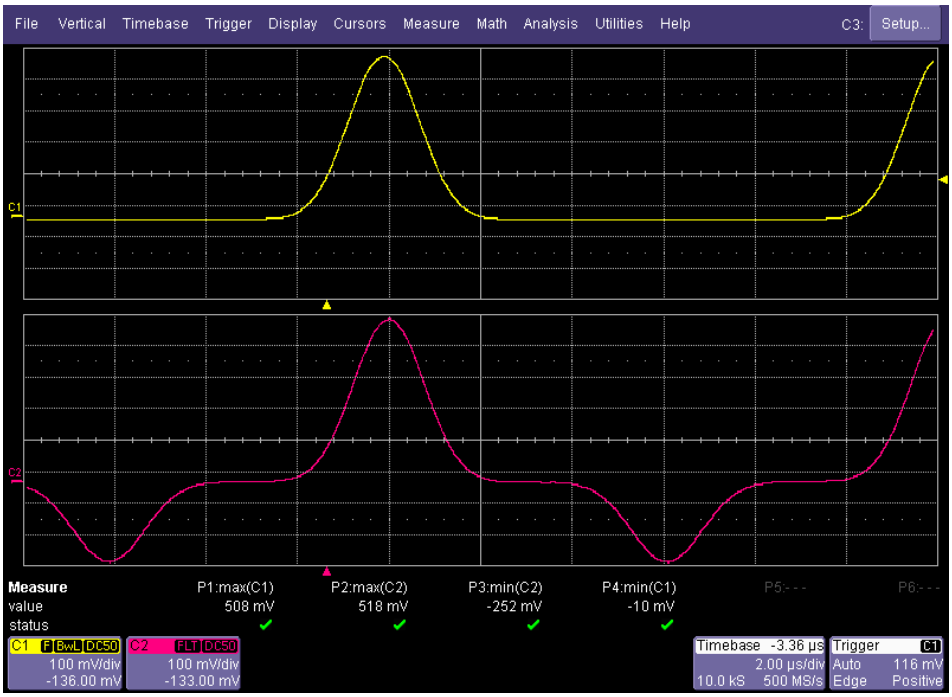

FIG. 4.16 – Signaux d'entrée et de sortie de la chaîne à base du filtre FIR 512 à 32 coefficients.

Pour valider notre architecture numérique, on observe alors les chronogrammes d'entrée et de sortie du CNA après troncature brutale (après le filtre FIR 512 à 32 coefficients), obtenus avec l'oscilloscope numérique comme on peut le voir sur la figure 4.17, équivalent à un zoom de la figure précédente.

La différence de temps entre la valeur maximale du signal d'entrée et la valeur maximale du signal de sortie est la latence de l'architecture numérique. On mesure cette différence de temps par l'intervalle de temps noté ΔX sur la figure 4.17 et on obtient 122 ns.

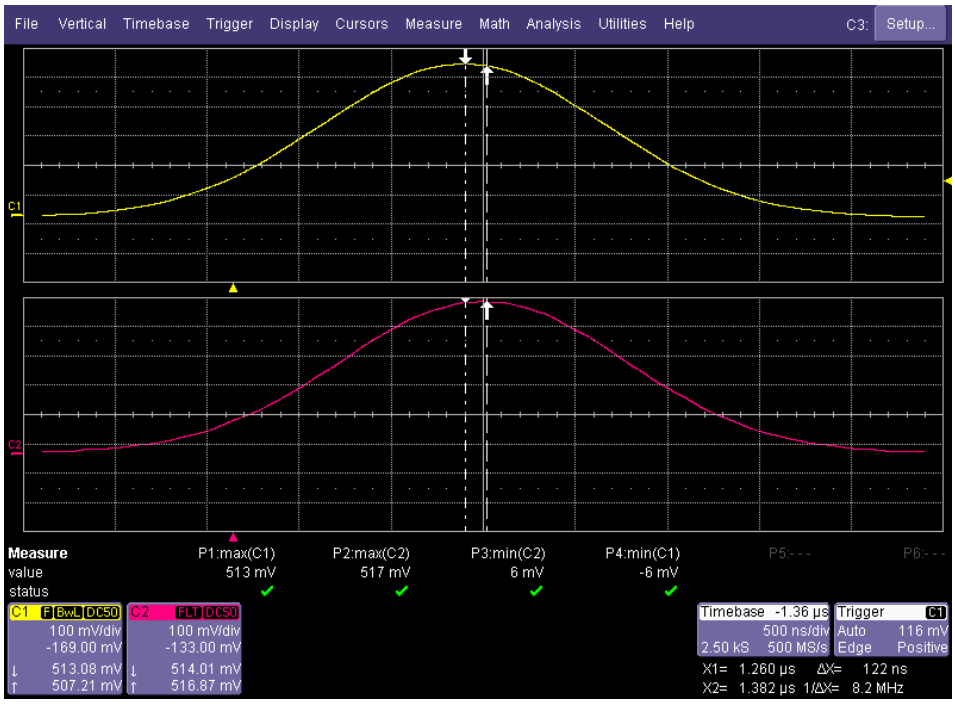

FIG. 4.17 – Mesure de la latence d'un filtre FIR 512 à 32 coefficients (entrée en jaune et sortie en rose).

A quelques ns près, cette valeur correspond bien à la latence d'un seul filtre FIR 512 à 32 coefficients qui est de 125 ns, mesurée par Modelsim. Nous pouvons donc dire que ce test répond à nos attentes.

4.4 Conclusion

Dans un milieu indoor, les réponses impulsionnelles sont généralement plus courtes qu'en milieu outdoor, moins de points sont nécessaires pour les décrire. La partie numérique du simulateur matériel de propagation a été réalisée dans le domaine temporel après avoir revu la théorie du produit de convolution en continu et en discret. Cette approche de la partie numérique dans le domaine temporel a montré l'évolution du signal à travers les différents éléments de la chaîne la constituant, comme par exemple le filtre FIR. Ce choix d'architecture permet un temps de calcul plus réduit mais utilise plus de ressources. Cela convient, cependant, pour des longueurs de filtre peu importantes, comme dans le cas des environnements « indoor ».

Dans un second temps, nous avons pu voir qu'une approche temporelle pour des environnements outdoor peut également être envisagée. En effet, en utilisant judicieusement les trajets significatifs d'un profil de réponse impulsionnelle, il est possible d'utiliser une

approche temporelle pour des environnements outdoor nécessitant de longues réponses impulsionnelles, en un minimum de temps. Cette solution permet également de palier au manque de précision de la solution fréquentielle présentée dans le chapitre précédent.

Conclusion

Ce travail de thèse, effectué dans le cadre du projet régional SIMPAA 2, a porté sur l'étude de faisabilité de la partie numérique d'un simulateur matériel pour les canaux de propagation MIMO.

La première étape a consisté, tout d'abord, à prendre connaissance du canal de propagation MIMO pour le caractériser et se familiariser avec quelques techniques de modélisation, le simulateur matériel devant être compatible avec des modèles aussi bien théoriques (déterministes ou statistiques) que standardisés, ou encore basés sur des mesures réelles. Puis, un bref état de l'art des simulateurs de canaux MIMO existants a été dressé. Les plus importants ont été détaillés, ce qui nous a permis de constater la complexité des appareils et leur coût.

Dans une seconde étape, le cahier des charges du simulateur matériel a été présenté à partir des objectifs du précédent projet RNRT SIMPAA, tout en apportant les éléments nécessaires à l'adaptation au projet actuel, notamment on a rajouté la gamme de fréquences WLAN à simuler. Les trois blocs principaux constituant le simulateur ont été présentés : le bloc radiofréquence permettant, d'une part, le passage des signaux RF vers des signaux réels, bas en fréquence, dont le spectre est compris entre 0 et la bande B et, d'autre part, le passage des signaux bas en fréquence vers des signaux RF, le bloc « Modèles de canal » permettant d'alimenter le simulateur avec différents modèles et le bloc « Partie numérique ». Afin d'avoir un bon compromis entre la complexité et la latence du système, deux solutions ont été envisagées pour concevoir la partie numérique du simulateur, selon que l'on se trouve en milieu indoor ou outdoor. Différents paramètres caractéristiques ont été introduits : durée des réponses impulsionnelles, décalage Doppler, rafraîchissement des profils de réponses impulsionnelles ou fréquentielles.

Cette seconde étape a également présenté les raisons du choix de la plate-forme de prototypage utilisée. Le choix du composant programmable FPGA Xilinx Virtex-4 y est tout d'abord présenté. Puis, on justifie le choix de la carte qui intègre, entre autres, ce composant programmable et une distribution adéquate des signaux d'horloge de façon

à pouvoir simuler la partie numérique du simulateur matériel de canaux de propagation MIMO.

La troisième étape est répartie sur deux chapitres. Elle montre la faisabilité de l'architecture numérique selon deux approches distinctes fréquentielle ou temporelle.

Tout d'abord, pour les environnements extérieurs (« outdoor »), une architecture dans le domaine fréquentiel a été envisagée. Après avoir rappelé le fonctionnement d'un module de transformée de Fourier rapide directe (FFT) et le lien avec la transformée de Fourier, l'évolution du signal a été montrée à travers les différents éléments constituant la chaîne numérique dans le domaine fréquentiel. L'importance des facteurs multiplicatifs en fin de chaîne a été mise en évidence de façon à retrouver le signal théorique que l'on souhaite obtenir. Pour cela, la comparaison des sorties en s'appuyant sur les calculs des erreurs relative et globale a permis de montrer les pourcentages d'erreurs introduits par les approximations successives introduites par les convertisseurs, les algorithmes de FFT et IFFT, la troncature du multiplieur complexe et les différentes troncatures brutale ou glissante de la sortie. Malgré les bonnes performances, en terme de ressources notamment, nous avons montré que cette première architecture fréquentielle n'était envisageable que si le signal de sortie avait une durée ne dépassant pas W_t, c'est-à-dire seulement si le nombre d'échantillons nécessaires pour représenter le signal de sortie ne dépassait pas la taille des blocs FFT et IFFT. Cette approche ayant pour défaut de tronquer prématurément les résultats en sortie des blocs FFT et IFFT, nous avons alors envisagé une autre solution fréquentielle. Cette nouvelle solution permet alors de fournir entièrement les réponses partielles en sortie des blocs FFT et IFFT, grâce au prolongement des signaux partiels d'entrée avec une séquence nulle. Les résultats obtenus sont alors très performants, au détriment malheureusement des ressources utilisées. En effet, le prolongement des signaux partiels oblige non seulement à doubler la taille des blocs FFT et IFFT ainsi que leur nombre, mais aussi à rajouter l'additionneur des signaux de sortie des blocs IFFT ainsi que la gestion des blocs FFT et IFFT en contretemps. L'autre inconvénient majeur de cette approche fréquentielle est la latence qui peut être trop grande selon l'application choisie.

Ensuite, une architecture dans le domaine temporel a été proposée pour les environnements à l'intérieur des bâtiments (« indoor »). Le produit de convolution et sa réalisation numérique à l'aide d'un filtre FIR y sont rappelés. On montre aussi l'évolution du signal à travers les différents éléments constituant la chaîne numérique et, de la même manière que pour l'approche fréquentielle, l'importance des facteurs multiplicatifs en fin de chaîne pour retrouver le signal théorique. Pour cette approche, la quantification du signal d'entrée et de la réponse impulsionnelle d'un canal SISO, le filtre numérique et la troncature finale introduisent une approximation du signal de sortie. Les mêmes calculs d'erreurs ou de rapport signal sur bruit ont permis d'évaluer la précision des calculs en tenant compte de ces approximations. Cette architecture utilise plus de ressources que la première proposition

d'architecture précédente, malgré l'économie de multiplieurs due à leur fréquence de fonctionnement que l'on choisit deux fois plus grande que la fréquence des échantillons du signal d'entrée. En revanche, la latence est plus faible et permet d'envisager des applications où la distance entre émetteur et récepteur est plus faible.

Enfin, nous avons envisagé une approche temporelle pour des applications en environnement extérieur avec un filtre FIR 512 dont 32 coefficients correspondent aux trajets les plus significatifs déterminés par des méthodes haute résolution, sujet d'un autre travail de thèse. Cette approche a montré qu'une architecture temporelle est envisageable pour des environnements outdoor, la latence étant beaucoup plus faible mais à condition de connaître les trajets les plus significatifs, et de ne pas être limité par les ressources utilisées.

On peut alors dresser une synthèse de ce travail en comparant les deux approches les plus significatives étudiées, leurs performances et leur utilité selon le contexte que l'on souhaite simuler.

	Approche temporelle FIR 64	Approche temporelle FIR 512 à 32 coefficients
Environnement	Indoor, pire cas WLAN	Pire cas rural
Retard maximal	1,28 µs	12,8 µs
Ressources pour un MIMO 2x2	4 filtres FIR 64 62 % du composant	4 filtres FIR 512 (32coeff.) 68 % du composant
Latence	135 ns	125 ns
Systèmes possibles	UMTS indoor, WLAN	UMTS
Améliorations possibles	Augmenter fclk et diminuer le nombre de bits	Un FPGA plus gros pour simuler plus de trajets
Utilisation	Environnement à courtes réponses impulsionnelles	Si la latence est critique mais pas le nombre de trajets simulés

FIG. 4.18 – Principales caractéristiques des deux approches étudiées.

En considérant les caractéristiques de notre simulateur proches de celles qu'on a vu dans le tableau 1.1, c'est-à-dire une précision de 14 bits, une largeur de bande de 20 MHz pour une bande de fréquence WLAN, une fois finalisé, notre simulateur pourrait entrer en concurrence avec le simulateur ACE 400NB d'Azimuth.

Ce travail a permis de montrer que, pour simuler un canal MIMO unidirectionnel 2×2, une architecture numérique est implémentable dans un seul composant programmable Virtex-4, ce qui n'avait pas été réalisé dans le précédent projet SIMPAA. Si l'on souhaite avoir une architecture avec plus d'antennes en émission ou en réception, il faudrait utiliser

plusieurs canaux SISO réalisés avec les mêmes blocs présentés. De plus, il faudra assurer la synchronisation entre les cartes Nallatech fournies.

Perspectives

Les perspectives de cette étude sont multiples. D'une part, il est possible d'améliorer la partie numérique de plusieurs manières.

Sachant que les filtres RF et FI utilisés ne sont pas parfaits, le filtre réel du signal en RF puis en FI va laisser passer un résidu du signal en dehors de la bande souhaitée de largeur B, en particulier les faibles fréquences du spectre RF. Le passage en basse fréquence, c'est-à-dire dans la bande [0, B] risque d'introduire un résidu autour de la fréquence nulle à cause du repliement du spectre. Il serait alors plus convenable de choisir f_0 entre $\frac{B}{2}$ et f_s.

Supposons qu'à la bande [0, B], on préfère la bande [d, $B+d$], avec d strictement positif. Après échantillonnage, on aura une bande inférieure [f_s - $(B+d)$, f_s - d] et une bande supérieure [f_s+d, $f_s+(B+d)$] autour de la fréquence f_0.

Afin d'éviter des problèmes de repliement de spectre lors du filtrage passe-bas du signal échantillonné, on doit respecter la condition « théorique » :

$$B+d < f_s-(B+d), \text{ donc } 0 < d < \frac{(f_s-2B)}{2}.$$

En réalité, ce filtrage ne sera pas idéal donc il faut assurer un écart, par exemple de d, entre $B+d$ et $f_s-(B+d)$. Dans ce cas, la condition « pratique » serait :

$$[f_s-(B+d)]-(B+d) > d, \text{ ce qui donne : } d < \frac{(f_s-2B)}{3}.$$

Par exemple, en WLAN, on a $B = 20$ MHz et $f_s = 50$ MHz. On devrait alors avoir d inférieur à $\frac{10}{3}$ soit 3,33 MHz. Si on considère $d = 3$ MHz, on pourrait choisir f_0 tel que :

$$f_0 = d + \frac{B}{2} = 3 + 10$$

soit 13 MHz à la place de $\frac{B}{2} = 10$ MHz, donc le signal réel passe-bas aura le spectre dans la bande [3 MHz, 13 MHz] au lieu d'être dans la bande [0, 10 MHz]. Les fréquences résiduelles comprises entre [0, 3 MHz] ne vont pas modifier le spectre « utile » du signal. Par contre, à partir d'un signal échantillonné, on devrait utiliser un filtre passe bande et non plus un filtre passe-bas. De plus, comme le décalage spectral de $d = 3$ MHz approche les bandes [d, $B+d$] et [$f_s-(B+d)$, f_s-d], il se peut que l'ordre du filtre passe-bande soit un peu augmenté.

Ensuite, l'exploitation du bus PCI de 32 bits peut être optimisée, par exemple, en envoyant les valeurs des échantillons de $h(t)$ par groupe de deux, chacune étant écrite sur

14 bits, et les 4 bits restant pour commander, d'une part, la distribution des profils dans les mémoires et, d'autre part, le rechargement des profils par la modification d'un bit.

Pour gagner en latence, une possibilité serait de diminuer la précision des calculs si on estime que des convertisseurs sur 10 ou 12 bits peuvent suffir à l'application envisagée.

Par ailleurs, en poussant les limites de la fréquence d'horloge acceptée par le FPGA, il serait possible d'avoir une fréquence d'horloge de 200 MHz pour l'architecture temporelle à base de filtres FIR et d'utiliser quatre fois les multiplieurs au lieu de deux fois, ce qui permettrait également une économie de ressources pour pouvoir éventuellement mettre plus de filtres FIR à l'intérieur d'un même composant programmable Virtex-4, pour le moment, ou Virtex-5 pour des applications futures, afin de pouvoir considérer des canaux MIMO de taille plus grande.

D'autre part, on peut compléter ce travail en développant les cartes radiofréquences évoquées dans le cahier des charges. Les parties RF pour les réseaux WLAN doivent être créées, tandis que celles pour l'UMTS sont à modifier, notamment la partie concernant le contrôle automatique de gain (CAG). De plus, pour les cartes UMTS, il faut prévoir un signal de sélection de l'oscillateur local selon le mode de fonctionnement FDD ou TDD.

Par ailleurs, le bloc numérique du simulateur matériel de propagation de canaux MIMO est implémenté comme un ensemble de cartes électroniques placé dans un ordinateur. Une interface utilisateur graphique (GUI) devra être développée pour permettre à l'utilisateur de configurer les paramètres de canal : type d'environnement, modèle de canal, fenêtre temporelle, vitesse du mobile, etc.

Enfin, de nombreuses campagnes de mesures doivent être effectuées avec le sondeur de canal MIMO réalisé à l'IETR, pour différents types d'environnements (intérieur, extérieur, pénétration) et pour les bandes de fréquences UMTS et WLAN. L'objectif final de ces mesures est d'obtenir des réponses impulsionnelles du canal MIMO réalistes et fiables, afin d'alimenter le bloc numérique du simulateur de matériel. Ce bloc numérique pourra également être alimenté par d'autres modèles de canaux théoriques (déterministe ou statistique) ou standardisés.

Ce travail devrait se poursuivre, début 2008, en partie grâce au projet régional CPER intitulé « PALMYRE 2 » et portant sur la réalisation d'une plate-forme expérimentale de développement de systèmes radioélectriques MIMO.

Communications et rapports

Communications internationales

- S. Picol, G. Zaharia, D. Houzet, G. El Zein, « Hardware Simulator for MIMO Radio Channels : Design and Features of the Digital Block ». Accepté pour être présenté au *IEEE 68th Vehicular Technology Conference (VTC-fall 2008)*, septembre 2008.
- S. Picol, G. Zaharia, D. Houzet, G. El Zein, « Features of the Digital Block of a Hardware Simulator for MIMO Radio Channels ». In *Proc. IEEE 8th International Symposium on Signals, Circuits and Systems (ISSCS 2007)*, pages 473-476, juillet 2007.
- S. Picol, G. Zaharia, D. Houzet, G. El Zein, « Design of the Digital Block of a Hardware Simulator for Mimo Radio Channels ». In *The 17th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications Proceedings (PIMRC'06)*, septembre 2006.
- S. Picol, G. Zaharia, G. El Zein, « MIMO Hardware Simulator Design For UMTS And WLAN Applications ». In *8th European Microwave Week 2005 Conference Proceedings - European Conference on Wireless Technology (ECWT 2005)*, pages 355-358, octobre 2005.
- S. Picol, G. Zaharia, G. El Zein, « Further Steps Towards the Development of a Hardware Simulator for MIMO Radio Channels». In *The 16th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications Proceedings (PIMRC'05)*, pages 557-561, septembre 2005.
- S. Picol, G. Zaharia, G. El Zein, « Towards the Development of a Hardware Simulator for MIMO Radio Channels ». In *Proc. IEEE International Symposium on Signals, Circuits and Systems (ISSCS 2005)*, juillet 2005.

Communications nationales

- S. Picol, H. Farhat, G. Zaharia, G. El Zein, « Vers le développement d'un simulateur matériel pour les canaux de propagation MIMO ». In *Proc. XIVèmes Journées Nationales Micro-Ondes*, mai 2005.
- S. Picol, H. Farhat, G. Zaharia, T. Quiniou, G. El Zein, D. Houzet, « La communication sans fil dans la vie de la cité ». In *Proc. Les Nouvelles Technologies dans la Cité, Cité & Tic 2004*, pages 343-352, décembre 2004.
- S. Picol, H. Farhat, G. Zaharia, T. Quiniou, G. El Zein, D. Houzet, « Simulateur Matériel de Propagation pour les Antennes Adaptatives (SIMPAA2) ». In *Proc. Les Nouvelles Technologies dans la Cité, Cité & Tic 2005*, pages 343-353, juin 2005.

Communication sans acte

- G. El Zein, H. Farhat, S. Picol, G. Zaharia, D. Houzet, F. Sagnard, « Mesures et Simulation du Canal de Propagation MIMO ». 1ères Journées PRACOM MIMO - ENST Bretagne, juin 2006.

Rapports scientifiques - PRIR SIMPAA 2

- S. Picol, H. Farhat, T. Quiniou, G. Zaharia, G. El Zein, « Rapport d'avancement - PRIR SIMPAA2 - bilan étape 1 », décembre 2004.
- S. Picol, H. Farhat, G. Zaharia, D. Houzet, G. El Zein, « Rapport d'avancement - PRIR SIMPAA2 - bilan étape 2 », décembre 2005.
- S. Picol, G. Zaharia, D. Houzet, G. El Zein, « Rapport d'avancement - PRIR SIMPAA2 - bilan étape 3 », décembre 2006.

Table des figures

1.1 Cellules radioélectriques 6
1.2 Propagation radioélectrique - Nature des interactions de l'onde électromagnétique avec le milieu 9
1.3 Fluctuations de la puissance reçue en fonction de la position du récepteur par rapport à l'émetteur 10
1.4 Module de la fonction d'autocorrélation de la réponse fréquentielle du canal 12
1.5 Illustration du phénomène Doppler 13
1.6 Module de la fonction d'autocorrélation de la réponse impulsionnelle du canal 14
1.7 Représentation d'un canal MIMO dans une configuration 2×2. 15
1.8 Les fonctions caractéristiques de Bello. 16
1.9 Profils de réponses impulsionnelles et fréquentielles. 16
1.10 Canal MIMO (2 x 2 canaux SISO unilatéraux). 17
1.11 Modélisation théorique du canal de propagation. 18
1.12 Simulateur matériel SmartSim d'ARC 22
1.13 Simulateur matériel SR5500 de Spirent Communications 23
1.14 Simulateur matériel ACE 400 NB d'Azimuth 24
1.15 Simulateur matériel Propsim C8 d'Elektrobit 25
1.16 Simulateur matériel TrueChannel de LitePoint dans sa configuration 4×4. . 26

2.1 Schéma fonctionnel du simulateur de canal MIMO unilatéral. 31
2.2 Schéma fonctionnel d'un bloc RF UMTS pour un canal SISO unidirectionnel. 32
2.3 Bloc RF UMTS d'un canal SISO unidirectionnel (réception et émission). . 33
2.4 Emetteur et récepteur du sondeur temporel réalisé à l'IETR. 34
2.5 Enveloppe complexe de la réponse impulsionnelle du canal. 35
2.6 Schéma fonctionnel de la partie numérique dans le domaine fréquentiel. . . 38
2.7 Schéma fonctionnel de la partie numérique dans le domaine temporel. . . . 39
2.8 Représentation d'un profil de réponse impulsionnelle pour un instant t fixé. 40
2.9 Evolution de la famille Virtex de Xilinx. 43

2.10 Kit de développement XtremeDSP pour Virtex-4. 44
2.11 Diagramme fonctionnel du kit de développement XtremeDSP pour Virtex-4. 45
2.12 Connexions des cartes de prototypages pour une configuration de 12 canaux SISO unidirectionnels. 47

3.1 Diagramme de la chaîne dans le domaine fréquentiel pour un canal SISO unidirectionnel. 52
3.2 Diagramme de la chaîne dans le domaine fréquentiel en numérique pour un canal SISO unidirectionnel. 52
3.3 Lien entre le PC et la carte Nallatech. 60
3.4 Chargement des profils de réponses fréquentielles pour un canal SISO . . . 61
3.5 Exemple de troncature. 63
3.6 Schéma de la troncature glissante. 63
3.7 Exemple de troncature glissante. 64
3.8 Amplification du signal d'entrée. 68
3.9 Utilisation du bloc FFT 512 de Xilinx. 69
3.10 Utilisation du multiplieur complexe de Xilinx. 71
3.11 Utilisation du module IFFT de Xilinx. 73
3.12 Utilisation de la troncature et du convertisseur numérique-analogique. . . . 74
3.13 Tension de sortie de la chaîne fréquentielle (échelle linéaire). 76
3.14 Tension de sortie de la chaîne fréquentielle (échelle logarithmique). 76
3.15 Erreur relative du signal de sortie de la chaîne fréquentielle. 78
3.16 Rapport signal à bruit du signal de sortie de la chaîne fréquentielle. 78
3.17 Exemple de détermination de la latence d'un seul module FFT avec Modelsim. 80
3.18 Interprétation de la latence 81
3.19 Comparaison de la sortie temporelle concaténée et du produit de convolution. 83
3.20 Architecture fréquentielle d'un canal SISO. 84
3.21 Architecture fréquentielle d'un canal SISO. 85

4.1 Signaux de la chaîne temporelle de simulation d'un canal SISO unidirectionnel. 90
4.2 Diagramme de la chaîne numérique dans le domaine temporel pour un canal SISO unidirectionnel. 90
4.3 Chargement des profils de réponses impulsionnelles pour un canal SISO . . 95
4.4 Architecture d'un filtre FIR pour un canal SISO unidirectionnel. 97
4.5 Tension de sortie de la chaîne temporelle en échelle linéaire et en échelle logarithmique. 99
4.6 Erreur relative du signal de sortie de la chaîne temporelle. 100
4.7 Rapport signal à bruit du signal de sortie de la chaîne temporelle. 101

4.8 Implémentation sur carte Nallatech d'un filtre FIR 64 pour un canal SISO. 103
4.9 Test du filtre FIR 64 implémenté sur Virtex-4 (entrée en jaune et sortie en rose). 104
4.10 Architecture d'un filtre FIR 512 avec 32 coefficients pour un canal SISO. . 105
4.11 Chargement des profils de réponses impulsionnelles pour un canal SISO. . . 107
4.12 Détail de l'architecture du chargement des profils de réponses impulsionnelles. 107
4.13 Tension de sortie de la chaîne complète avec le filtre FIR 512 à 32 coefficients. 109
4.14 Erreur relative du signal de sortie de la chaîne temporelle. 110
4.15 Rapport signal à bruit du signal de sortie de la chaîne temporelle. 110
4.16 Signaux d'entrée et de sortie de la chaîne à base du filtre FIR 512 à 32 coefficients. 112
4.17 Mesure de la latence d'un filtre FIR 512 à 32 coefficients (entrée en jaune et sortie en rose). 113
4.18 Principales caractéristiques des deux approches étudiées. 117

Liste des tableaux

1.1 Tableau comparatif des principales caractéristiques de différents simulateurs matériels. 27

2.1 Résumé des spécifications du simulateur matériel de canaux de propagation MIMO. 31
2.2 Tableau de paramètres. 41

3.1 Taux d'occupation du composant pour un seul module FFT 512. 58
3.2 Comparaison des valeurs globales de l'erreur relative et du SNR. 79
3.3 Taux d'occupation du composant pour la chaîne fréquentielle. 80
3.4 Taux d'occupation du composant pour un canal MIMO 2×2 unidirectionnel. 82

4.1 Taux d'occupation du composant pour un seul filtre FIR 64. 93
4.2 Comparaison des valeurs globales de l'erreur relative et du SNR. 102
4.3 Taux d'occupation du composant programmable Virtex-IV SX35 pour 4 FIR 64 102
4.4 Comparaison des valeurs globales de l'erreur relative et du SNR. 111
4.5 Taux d'occupation du composant programmable Virtex-IV SX35 pour 4 filtres FIR 512 à 32 coefficients 111

Bibliographie

[1] P. Guguen et G. El Zein, *Les techniques multi-antennes pour les réseaux sans fil.* Hermes Sciences Publications, 2004.

[2] P. Guguen, *Techniques multi-antennes émission-réception - Applications aux réseaux domestiques sans fil. Disponible à l'adresse http ://tel.archives-ouvertes.fr/docs/00/04/55/42/PDF/tel-00003348.pdf.* Thèse de Doctorat, IETR - INSA de Rennes, janvier 2003.

[3] G. J. Foschini et M. J. Gans, « On limits of wireless communications in a fading environment when using multiple antennas ». *Wireless Personal Communications*, vol. 6, pages 311–335, mars 1998.

[4] J.-M. Conrat, « 1-1PrapportSpecifFonctA1-0_FTR&D.pdf, spécifications fonctionnelles ». *Rapport technique*, 2000.

[5] M. Résibois, « PlAte-forme de déveLoppeMent et d'évaluation de sYstèmes RadioElectriques. Etat d'avancement au 1er mai 2005 ». *Rapport technique.*

[6] T. S. Rappaport, *Wireless Communications Principles & Practice.* Prentice Hall, 1996.

[7] C. E. Shannon, « Communication in the presence of noise ». In *Proceedings of the IRE*, vol. 37, pages 10–21, janvier 1949.

[8] L. Boithias, *Propagation des ondes radioélectriques dans l'Environnement Terrestre.* Dunod, 1984.

[9] A. Glavieux et M. Joindot, *Communications numériques - Introduction.* Masson, 1996.

[10] J. G. Proakis, *Digital Communications.* Mc Graw-Hill, 1995.

[11] M. Schwartz, *Principles of electrodynamics.* McGraw-Hill, 1972.

[12] L. Boithias, *Radio wave propagation.* McGraw-Hill, 1987.

[13] W. C. Jakes, *Microwave Mobile Communications.* IEEE Press, Wiley, 1974.

[14] S.J. Howard et K. Pahlavan, « Measurement and Analysis of the Indoor Radio Channel in the Frequency Domain ». *IEEE Trans. on Instrumentation & Measurements*, vol. Vol.39, pages 751–755, octobre 1990.

[15] Gh. Zaharia, *Caractérisation de la Propagation Radioélectrique à l'Intérieur des Bâtiments à 900 MHz - Implications sur la Qualité d'une Transmission Numérique CT2.* Thèse de Doctorat, INSA de Rennes, février 1997.

[16] P.A. Bello, « Characterization of randomly time-variant linear channels ». *IEEE Trans. on Commun. Syst.*, vol. volume 11, pages 360–393, Décembre 1963.

[17] R. Cosquer, *Conception d'un sondeur de canal MIMO - Caractérisation du canal de propagation d'un point de vue directionnel et doublement directionnel. Disponible à l'adresse http ://tel.ccsd.cnrs.fr/docs/00/04/74/09/PDF/tel-00007659.pdf.* Thèse de Doctorat, IETR - INSA de Rennes, octobre 2004.

[18] J. Guillet, *Caractérisation et modélisation spatio-temporelles du canal de propagation radioélectrique dans le contexte MIMO. Disponible à l'adresse http ://tel.ccsd.cnrs.fr/documents/archives0/00/00/80/11.* Thèse de Doctorat, IETR - INSA de Rennes, juillet 2004.

[19] P. Almers, E. Bonek, A. Burr, N. Czink, M. Debbah, V. Degli-Esposti, H. Hofstetter, P. Kyösti, D. Laurenson, G. Matz, A. F. Molisch, C. Oestges et H. Özcelik, « Survey of Channel and Radio Propagation Models for Wireless MIMO Systems ». In *EURASIP Journal on Wireless Communications and Networking*, pages 56–74, janvier 2007.

[20] W.C. Lee, *Mobile communications engineering.* McGraw-Hill, 1997.

[21] R. Ertel, P. Cardieri, K. Sowerby, T. S. Rappaport et J. Reed, « Overview of spatial channel models for antenna array communication systems ». *IEEE Personal Communications*, pages 10–22, février 1998.

[22] K. Yu et B. Ottersten, « Models for MIMO propagation channels : a review ». *Wireless Communications and Mobile Computing*, vol. 2, pages 653–666, août 2002.

[23] D. Gesbert, M. Shafi, D. Shiu, P. J. Smith et A. Naguib, « From theory to practice : An overview of MIMO space-time coded wireless systems ». *IEEE Journal on Selected Areas in Communications*, vol. vol. 21, no. 3, pages 281–301, avril 2003.

[24] M. A. Jensen et J. W. Wallace, « A review of antennas and propagation for MIMO wireless communications ». *IEEE Transactions on Antennas and Propagation*, vol. AP-52, n° 11, pages 2810–2824, novembre 2004.

[25] L. Schumacher, L. T. Berger et J. Ramiro-Moreno, « Recent advances in propagation characterisation and multiple antenna processing in the 3 GPP framework ». In *Proceedings of the XXVIth URSI General Assembly, Maastrich, The Netherlands*, août 2002.

[26] K.A. Remley, A.Weisshaar et H.R. Anderson, « A Comparative study of ray tracing and FDTD for indoor propagation modeling ». *Proc. 48th IEEE Annu. Vehicular Technology Conf. ,Ottawa, Ont., Canada*, pages 865–869, mai 1998.

[27] Y. Wang, S. Safavi-Naeini et S. K. Chaudhuri, « A combined ray tracing and FDTD method for modeling indoor radio wave propagation ». *IEEE Antennas and Propagation Society International Symposium*, vol. 1998 Digest-Vol. 3, pages 1668–1671, Atlanta, GA, juin 1998.

[28] A. L. Swindlehurst, G. German, J. W. Wallace et M. A. Jensen, « Experimental measurements of capacity for MIMO indoor wireless channels ». In *Proceedings IEEE the 3rd Workshop on Signal Processing Advances in Wireless Communications (SPAWC'01)*, pages 30–33, mars 2001.

[29] S. Vergerio, *Recherche des caractéristiques optimales d'antennes multi-capteurs pour les systèmes MIMO.* Thèse de Doctorat, Université de Provence, Institut Fresnel, Marseille, juillet 2007.

[30] J.-P. Rossi et Y. Gabillet, « A mixed ray launching/tracing method for full 3-D propagation modeling and comparison with wide-band measurements ». *IEEE Transactions on Antennas and Propagation*, vol. 50, pages 517–523, avril 2002.

[31] M. Salehudin, *Modélisation statistique du canal radiomobile large bande à 910 MHz en site urbain.* Thèse de Doctorat, Université de Rennes 1, avril 1989.

[32] G. El Zein, *Etude et réalisation d'une liaison numérique radiomobile à étalement de spectre en site urbain.* Thèse de Doctorat, Université de Rennes I, juin 1988.

[33] A. A. M. Saleh et R. A. Valenzuela, « A Statistical Model for Indoor Multipath Propagation ». *IEEE Journal on Selected Areas in Communications*, vol. 5, n°2, pages 128–137, 1987.

[34] *www.etsi.org.*

[35] ETSI, « Broadband Radio Access Networks (BRAN), "High PErformance Radio Local Area Network - Type 2", System Overview. Technical Report TR 101 683 ». Rapport, 1998.

[36] *www.itu.int/ITU-R.*

[37] *www.3gpp.org.*

[38] *www.3gpp2.org.*

[39] 3GPP, « Spatial Channel Model for MIMO simulations, Release 6 ». 2003.

[40] L. M. Correia (Ed.), *Mobile Broadband Multimedia Networks. Techniques, Models and Tools for 4G. COST 273 MIMO channel model.* Academic Press (Elsevier), 2006.

[41] J. Kolu, T. Jämsä et A. Hulkkonen, « Real time simulation of measured radio channels ». *Vehicular Technology Conference. IEEE 58th VTC 2003-Fall*, vol. 1, pages 183–187, octobre 2003.

[42] A. Alcocer-Ochoa, V. Ya. Kontorovitch et R. Parra-Michel, « A MIMO Channel Simulator Applying the Universal Basis ». *Vehicular Technology Conference. IEEE 64th VTC 2006-Fall*, septembre 2006.

[43] M. Pätzold et B. O. Hogstad, « A Wideband Space-Time MIMO Channel Simulator Based on the Geometrical One-Ring Model ». *Vehicular Technology Conference. IEEE 64th VTC 2006-Fall*, septembre 2006.

[44] J.-M. Conrat et P. Pajusco, « A versatile propagation channel simulator for MIMO link level simulation ». In *EURASIP Journal on Wireless Communications and Networking*, 2007.

[45] *www.smartsim.at.*

[46] *www.spirentcom.com.*

[47] *www.azimuthsystems.com.*

[48] *www.elektrobit.com.*

[49] *www.litepoint.com.*

[50] *www.smartsim.at/details/smartsim_ details.htm.*

[51] L. M. Correia (Ed.), *Wireless Flexible Personalised Communications. COST 259 : European Co-operation in Mobile Radio Research.* John Wiley & Sons, 2001.

[52] *www.spirentcom.com/analysis/technology.cfm ?media=7&WS=325&SS=156&D=4.*

[53] *www.azimuthsystems.com/platforms-channel-400nb.htm.*

[54] *www.elektrobit.com/index.php ?207.*

[55] *www.litepoint.com/content/view/107/142/.*

[56] *www.rnrt.org.*

[57] H. Farhat, R. Cosquer, J. Guillet et G. El Zein, « Mobile Radio Channels Characterization based on Wide-Band SIMO 1 x 8 Measurements at 2.2 GHz ». *Proceedings of the 5th International Conference on ITS Telecommunications (ITST '05)*, pages 321–324, juin 2005.

[58] H. Farhat et G. El Zein, « Antenna Arrays Design and Calibration for High Resolution MIMO channel sounding at 3.5 GHz ». *Proceedings of the 66th IEEE Vehicular Technology Conference 2007 (VTC '07)*, septembre 2007.

[59] S. Salous, « Multiple input multiple output systems : capacity and channel measurements ». In *Proceedings of the Seventh World Multiconference on Systemics, Cybernetics and Informatics (SCI2003)*, pages 1–5, juillet 2003.

[60] *www.xilinx.com.*

[61] *www.xilinx.com/products/silicon_ solutions/fpgas/virtex/virtex4/overview/index.htm.*

[62] *http ://direct.xilinx.com/bvdocs/publications/ds112.pdf.*

[63] *www.xilinx.com/xlnx/xebiz/designResources/ip_ product_ details.jsp ?key=DO-DI-DSP-DK4.*

[64] *www.analog.com/en/prod/0,2877,AD6645,00.html.*

[65] *www.analog.com/en/prod/0,2877,AD9772A,00.html.*

[66] « Xilinx Virtex-II Platform FPGAs : Complete Data Sheet ». *Xilinx, Inc., 2100 Logic Drive, San Jose, CA 95124, http ://direct.xilinx.com/bvdocs/publications/ds031.pdf*, 2005.

[67] *www.xilinx.com/bvdocs/userguides/ug_xtremedsp_devkitIV.pdf.*

[68] J. W. Cooley et J. W. Tukey, *An Algorithm for the Machine Computation of Complex Fourier Series*, vol. 19. Mathematics of Computation, avril 1965.

[69] P. Duhamel et M. Vetterli, « Fast Fourier transforms : a tutorial review and a state of the art ». *IEEE Transaction on Signal Processing*, vol. 19, n°4, pages 259–299, 1990.

[70] M. Bellanger, *Traitement numérique du signal.* Dunod, 1998.

[71] J. G. Proakis et D. G. Manolakis, *Digital Signal Processing Principles, Algorithms and Applications.* Second Edition, Maxwell Macmillan International, 1992.

[72] *Xilinx Fast Fourier Transform v3.1, DS260*, november 2004.

[73] *Complex Multiplier v2.0, DS291*, november 2004.

[74] *www.xilinx.com/ipcenter/catalog/logicore/docs/xfft.pdf.*

[75] *www.xilinx.com/bvdocs/ipcenter/data_sheet/cmpy.pdf.*

INSA de Rennes
Service des formations

AVIS DU JURY SUR LA REPRODUCTION DE LA THESE SOUTENUE

Titre de la thèse : Conception et réalisation de la partie numérique d'un simulateur matériel pour les canaux de propagation MIMO.

Nom Prénom de l'auteur : PICOL Sylvie

Membres du jury :

	Monsieur	BOUTILLON
	Monsieur	VAUZELLE
	Monsieur	EL ZEIN
	Monsieur	DEGAUQUE
	Monsieur	GUILLOUARD
	Monsieur	HOUZET
	Monsieur	PAJUSCO
	Monsieur	ZAHARIA

Président du jury : DEGAUQUE

Date de la soutenance : 05/12/2007

Reproduction de la thèse soutenue :

- [x] Thèse pouvant être reproduite en l'état
- [ ] Thèse ne pouvant être reproduite
- [ ] Thèse pouvant être reproduite après corrections suggérées

Le Directeur,

A. JIGOREL

Rennes, le 05/12/2007

Signature du Président du jury

Résumé

Le besoin d'améliorer la capacité des réseaux sans-fil, pour la transmission de données et les applications multimédia des systèmes de communications numériques, a amené à s'intéresser aux techniques de communication MIMO (Multiple Input Multiple Output) pour augmenter le débit et la qualité des transmissions. Les systèmes MIMO se servent de réseaux d'antennes, à la fois à l'émission et à la réception d'une communication radio et ainsi exploitent la dimension spatiale pour la transmission de l'information. Les performances expérimentales d'un système de communication peuvent être évaluées en utilisant un simulateur matériel du canal de propagation. Cette thèse a pour objectif la conception et la réalisation de la partie numérique d'un simulateur matériel pour les canaux de propagation MIMO, pour les réseaux UMTS et WLAN. La simulation matérielle du canal radio assure la vitesse de traitement nécessaire. D'autres avantages de la simulation matérielle sont l'évaluation des performances en temps réel et la comparaison de divers systèmes de radiocommunication dans un canal de propagation variant dans le temps, dans les mêmes conditions de test. L'étude de faisabilité a permis d'établir deux approches, l'une pour les environnements indoor, l'autre pour les environnements outdoor. Les architectures numériques proposées ont été étudiées et implémentées dans un composant programmable FPGA. L'évaluation des performances et de la complexité montre l'intérêt de ces architectures. Ce travail a été réalisé dans le cadre du projet PRIR SIMPAA 2, avec le soutien de la Région Bretagne.

Mots clés : Simulateur matériel - Communications numériques - Canal de propagation - MIMO - UMTS - WLAN - Architecture numérique - FPGA.

Abstract

The need to improve the capacity of wireless networks, for the data transmission and multimedia applications of the digital communication systems, has to focus on MIMO communication techniques to increase the data rate and the quality of services. MIMO systems use antenna arrays at both the transmit and the receive sides of a radio link and exploit the spatial dimension for the transmission. The performance of a communication system can be evaluated using a hardware simulator of radio propagation channels. This Ph.D. deals with the design and the implementation of the digital block of a hardware simulator of MIMO propagation channels, for both UMTS and WLAN networks. The hardware simulation of radio channel provides the necessary processing speed. Other advantages of the hardware simulation are the real time performance evaluation and the comparison of different systems in a time varying radio propagation channel under the same test conditions. Two approaches have been considered, one for indoor environments, the other for outdoor environments. The proposed digital architectures were studied and implemented within FPGA. Performance and complexity evaluation shows the interest of these architectures. This work was carried out under the project SIMPAA 2, with the support of the « Région Bretagne ».

Index Term : Hardware simulator - Digital communications - Propagation channel - MIMO - UMTS - WLAN - Digital architecture - FPGA.