

Université du Québec  
Institut National de la Recherche Scientifique  
Centre Énergie Matériaux Télécommunications

**PROTOTYPAGE RADIO-LOGICIEL RAPIDE  
DU MODULE SENSORIEL D'UN NOUVEL ÉMETTEUR-RÉCEPTEUR  
COGNITIF POUR LES RÉSEAUX SANS-FIL 5G**

Par  
Yassine Selmi

Mémoire présenté pour l'obtention du grade de  
*Maître es Sciences*, M.Sc.  
en Télécommunications

**Jury d'évaluation**

Examineur externe	Prof. Paul Fortier Université Laval
Examineur interne	Prof. Tarek Djerafi INRS-EMT
Directeur de recherche	Prof. Sofiène Affes INRS-EMT
Co-directeur de recherche	Dr. Abdelaziz Samet INRS-EMT

*À mes parents.*

# Remerciements

Ce travail a été élaboré au sein de l'équipe Wireless Lab <[www.wirelesslab.ca](http://www.wirelesslab.ca)> du Centre Énergie, Matériaux et Télécommunications de l'Institut National de la Recherche Scientifique.

Je profite, par le biais de ce mémoire, pour exprimer mes vifs remerciements à toutes personnes qui ont contribué au succès de ce travail.

Tout d'abord, je tiens à exprimer mes remerciements à mon directeur de recherche le Professeur Sofiène Affes qui m'a offert l'opportunité d'intégrer l'équipe Wireless Lab et qui m'a guidé avec ces prodigieux conseils et directives tout au long de l'accomplissement de mes travaux de recherche. Sa confiance en moi m'a permis de mieux connaître ma valeur et mon potentiel.

J'exprime également ma profonde gratitude à Dr. Abdelaziz Samet, mon co-directeur de recherche, pour son aide, sa disponibilité et ses conseils précieux. Son expertise et sa sympathie sont les ingrédients secrets de ce succès.

Un remerciement particulier à Dr. Faouzi Bellili pour son écoute, ses explications pertinentes et sa volonté d'aider.

Je tiens à remercier aussi CMC Microsystems pour la fourniture de produits et de services qui ont facilité ce travail de recherche, y compris les outils de CAO et les plateformes de prototypage.

Mes remerciements vont également aux membres du Jury pour l'attention qu'ils ont portée à ce mémoire.

Finalement, je présente ma reconnaissance à toute personne qui a contribué de près ou de loin à ce travail.



# Résumé

Dans une infrastructure de communication radio-mobile multi-connectée, multi-technologie et omniprésente, comme prévu pour le 5G, l'intelligence artificielle et la cognition joueront un rôle majeur. Un objectif important des futures infrastructures mobiles est que celles-ci prennent conscience de leur écosystème pour qu'elles puissent s'adapter à leur contexte, du lien physique aux couches applicatives, de sorte à optimiser leurs performances selon de nombreux critères. Ce qui donne naissance à un nouveau paradigme connu sous le nom de la radio cognitive « context-aware » (CR). Les émetteurs-récepteurs CR (CTR) intègrent principalement un module de décision s'appuyant sur diverses entités sensorielles, qui fournissent des informations suffisantes sur la qualité de la liaison par l'estimation de divers paramètres du canal sans-fil. Deux paramètres importants et nécessaires dans plusieurs architectures CTR: le rapport signal-à-bruit (SNR) et l'étalement de Doppler. Nous proposons donc, dans ce projet une technique de prototypage rapide d'un estimateur conjoint à maximum de vraisemblance (ML) assisté par pilotes du SNR et de l'étalement de Doppler récemment développé par les membres de notre équipe Wireless Lab <[www.wirelesslab.ca](http://www.wirelesslab.ca)> et qui a démontré sa capacité de surpasser les principales solutions existantes dans la littérature à la fois en termes de précision et de complexité. Une architecture parallèle et efficace pour l'estimateur en question a été implémentée sur une plateforme radio-logicielle à base de FPGA. Ensuite, le design proposé a été testé en temps réel et dans des conditions de propagation réalistes, construites autour d'un émulateur de canal ultra-puissant. Les co-simulations « Hardware-in-the-loop » (HIL) ont été utilisées pour évaluer les performances du design proposé. En comparant les résultats obtenus par simulations MATLAB à virgule flottante, ceux de la co-simulation HIL suggèrent une perte négligeable même en présence de plusieurs imperfections matérielles, validant ainsi et confirmant le potentiel de l'architecture proposée pour être intégrée dans des systèmes cognitifs de la future génération 5G.

**Mots-clés** 5G; Radio cognitive; radio-logicielle; estimation de paramètre; SNR; étalement de Doppler; FPGA.



# Abstract

In a multi-connected, multi-technology, and pervasive mobile infrastructure, such as what is being planned for 5G, artificial intelligence and cognition will play a major role. An important goal of future mobile infrastructures is to self-adapt their characteristics to the general context, at the physical link, as well as at network and application layers, which gives rise to a new paradigm known as context-aware cognitive radio (CR). CR transceivers (CTRs) mostly incorporate a cognitive engine that relies on various sensorial entities, which attempt to provide sufficient information about the quality of the link through the estimation of various key channel parameters. Two important parameters are required in a wide range of CTR architectures: the signal-to-noise ratio (SNR) and the Doppler spread. Within this context, we propose a hardware rapid prototyping technique of a joint data-aided (DA) maximum likelihood (ML) SNR and Doppler spread estimator recently shown to outperform main state-of-the-art solutions both in terms of accuracy and complexity. We implement this new joint DA ML estimator on an FPGA-based software defined radio (SDR) platform using a deep-pipelined and resource-efficient design. And we validate and test the latter in real-time under realistic propagation conditions reproduced by a high-scalability channel emulator. Compared to its original MATLAB floating-point version, hardware-in-the-loop (HIL) co-simulation results suggest negligible losses in performance of the proposed design despite the existence of several hardware impairments, thereby confirming its very strong potential and attractiveness for possible integration on future 5G CTRs.

**Keywords** 5G; Context-awareness; cognitive radio; software defined radio; parameter estimation; SNR; Doppler spread; FPGA.





# Table des matières

<b>Remerciements</b>	<b>iii</b>
<b>Résumé</b>	<b>v</b>
<b>Abstract</b>	<b>vii</b>
<b>Table des matières</b>	<b>ix</b>
<b>Liste des figures</b>	<b>xi</b>
<b>Liste des tableaux</b>	<b>xiii</b>
<b>Liste des abréviations</b>	<b>xv</b>
<b>Introduction générale</b>	<b>1</b>
<b>1 Méthodologies de conception et de prototypage rapide</b>	<b>5</b>
1.1 Introduction . . . . .	5
1.2 Présentation de la radio-logicielle . . . . .	6
1.2.1 Principe de la radio-logicielle . . . . .	6
1.2.2 Architecture de la radio-logicielle . . . . .	7
1.3 Méthodologies et outils de conception matérielle: de l'algorithme à l'architecture . . . . .	10
1.3.1 Méthodologies de conception . . . . .	10
1.3.2 Outils de conception . . . . .	12
1.4 Présentation de la station expérimentale . . . . .	14
1.4.1 La miniBEE4 de BEEcube . . . . .	14
1.4.2 Émulateur de canal EB PropSim FS8 . . . . .	15
1.5 Conclusion . . . . .	17
<b>2 Présentation du CTR et des estimateurs de l'étalement de Doppler et du SNR</b>	<b>19</b>
2.1 Introduction . . . . .	19
2.2 Présentation de l'émetteur-récepteur cognitif CTR . . . . .	20
2.2.1 Architecture globale du CTR . . . . .	20
2.2.2 Règles de décision . . . . .	23
2.3 Choix d'estimateurs de l'étalement de Doppler et du SNR . . . . .	23
2.3.1 Choix de l'estimateur de l'étalement du Doppler . . . . .	24
2.3.2 Choix de l'estimateur du SNR . . . . .	26
2.4 Modèle du système et formulation mathématique des estimateurs choisis . . . . .	26
2.4.1 Modèle du système . . . . .	26

2.4.2	Présentation de l'estimateur de l'étalement de Doppler . . . . .	28
2.4.3	Présentation de l'estimateur du rapport signal-à-bruit . . . . .	30
2.5	Conclusion . . . . .	31
<b>3</b>	<b>Architecture matérielle proposée pour le module sensoriel du CTR</b>	<b>33</b>
3.1	Introduction . . . . .	33
3.2	Conception matérielle: compromis et optimisations . . . . .	34
3.2.1	Contraintes de conception d'un circuit numérique . . . . .	34
3.2.2	Stratégies d'optimisation . . . . .	35
3.3	Architecture globale . . . . .	36
3.4	Architecture proposée pour l'estimateur ML du SNR . . . . .	39
3.4.1	Structure interne du module « Estimation du canal » . . . . .	39
3.4.2	Structure interne du module « Estimation de la puissance du bruit » . . . . .	41
3.4.3	Structure interne du module « Estimation du SNR » . . . . .	42
3.5	Architecture proposée pour l'estimateur ML de l'étalement de Doppler . . . . .	43
3.5.1	Conception et optimisation de l'architecture de LLF . . . . .	44
3.5.2	Structure interne du sous-module Argmax . . . . .	50
3.5.3	Optimisation de la technique de recherche sur le maximum de la LLF . . . . .	50
3.6	Conclusion . . . . .	52
<b>4</b>	<b>Implémentation et évaluation de performances</b>	<b>53</b>
4.1	Introduction . . . . .	53
4.2	Implémentation du module sensoriel . . . . .	53
4.3	Évaluation de la complexité matérielle . . . . .	55
4.4	Évaluation de la consommation d'énergie . . . . .	57
4.5	Évaluation de la latence computationnelle . . . . .	58
4.6	Évaluation de la précision de l'estimation . . . . .	60
4.6.1	Résultats de l'identification du canal de transmission . . . . .	60
4.6.2	Résultats de l'estimation du SNR instantanée . . . . .	62
4.6.3	Résultats de l'estimation de l'étalement de Doppler . . . . .	64
4.7	Conclusion . . . . .	68
	<b>Conclusion générale</b>	<b>69</b>
	<b>Références</b>	<b>71</b>

# Liste des figures

1.1	Évolution de la radio-matérielle à la radio-logicielle. . . . .	7
1.2	Architecture globale de la radio définie par logiciel. . . . .	9
1.3	Diagramme de Gajski-Kuhn: perspectives et niveaux d'abstraction pour la conception des systèmes numériques. . . . .	13
1.4	Architecture interne de la plateforme radio-logicielle miniBEE4. . . . .	15
1.5	Flot de conception et d'implémentation du design sur la plateforme miniBEE4. . . . .	16
1.6	L'émulateur de canal EB Propsim FS8. . . . .	17
2.1	Architecture du CTR d'un point de vue système. . . . .	22
2.2	Règles du décision du CTR basées sur l'estimation de canal ML et LS. . . . .	24
2.3	Comparative de l'estimateur choisi avec les benchmarks en termes de l'erreur quadratique moyenne normalisée (NMSE) de l'estimation de la fréquence de Doppler, à SNR = 0 dB et avec 1000 échantillons reçus. . . . .	25
2.4	Disposition temporelle des données et des symboles pilotes sur $M$ fenêtres d'approximation locale. . . . .	28
3.1	Diagramme en bloc de l'architecture globale. . . . .	37
3.2	Diagramme d'états de la « MSA fenêtrage ». . . . .	38
3.3	Architecture interne du module « Estimation du canal ». . . . .	40
3.4	Design interne du module « Estimation de la puissance du bruit ». . . . .	42
3.5	Design interne du module « Estimation de la puissance du bruit ». . . . .	42
3.6	Diagramme en bloc de la structure interne du module « Estimation de l'étalement de Doppler ». . . . .	45
3.7	Exemple illustratif de la technique de recherche de l'exposant et la mantisse d'un signal représenté en virgule fixe. . . . .	47
3.8	Architecture proposée pour le calcul du logarithme d'un nombre réel. . . . .	47
3.9	Exemple d'architecture interne optimisée pour le calcul de la fonction sinus. . . . .	48
3.10	Diagramme d'état de la « MSA estimation de l'étalement de Doppler ». . . . .	51
4.1	Variation du facteur de latence $Q$ en fonction du pas $fine\_step$ pour les deux approches de recherche sur le maximum, avec $coarse\_step = 10 \times fine\_step$ , $max = 10000$ , et $min = 0$ . . . . .	59
4.2	Canal estimé vs. canal vrai pour (a) $f_D = 400$ Hz, (b) $f_D = 800$ Hz, and (c) $f_D = 1200$ Hz. . . . .	61
4.3	NMSE du SNR instantané obtenue par la co-simulation mBEE4/EB Propsim contre les simulations MATLAB en fonction du SNR moyen, avec $f_D = 200$ Hz and $L = 3$ . . . . .	62

4.4	Histogramme de l'estimée du SNR instantané à $f_D = 200$ Hz, SNR = 0 dB, et $M_c = 10000$ .	63
4.5	CDF de la NMSE du SNR instantané à $f_D = 200$ Hz et SNR = 0 dB.	64
4.6	NMSE de la fréquence maximale de Doppler obtenue par la co-simulation mBEE4/EB Propsim contre les simulations MATLAB en fonction du SNR moyen, avec $f_D = 200$ Hz and $L = 3$ .	65
4.7	NMSE de la fréquence maximale de Doppler obtenue par la co-simulation mBEE4/EB Propsim contre les simulations MATLAB en fonction de la fréquence de Doppler réelle, à SNR = 0 dB.	66
4.8	Histogramme de l'estimée de la fréquence maximale de Doppler à SNR = 0 dB et $M_c = 10000$ .	67
4.9	CDF de la NMSE de la fréquence maximale de Doppler à $f_D = 200$ Hz et SNR = 0 dB.	67

# Liste des tableaux

1.1	Classification des radio-logicielles selon leur degré de reconfiguration . . . . .	8
4.1	Configuration des paramètres de la co-simulation HIL . . . . .	55
4.2	Complexité matérielle à $N = 128$ , $M = 8$ , et $L = 3$ . . . . .	56
4.3	Résultats de l'estimation de la puissance consommée et la fréquence maximale d'opération à la température $25^{\circ}\text{C}$ . . . . .	57
4.4	Latence computationnelle de l'architecture proposée à une fréquence d'opération de $61.44\text{ MHz}$ , $N = 128$ , $M = 8$ , $L = 3$ , $min = 0$ , $max = 10000$ , $coarse\_step = 400$ , et $fine\_step = 40$ . . . . .	58



# Liste des abréviations

ASIC	Application Specific Integrated Circuit
BEE	Berkeley Emulation Engine
BPS	BEEcube Platform Studio
BRAM	Block Read Access Memory
CAN	Convertisseur Analogique Numérique
CDF	Cumulative Distribution Function
CLB	Configurable Logic Block
CNA	Convertisseur Numérique Analogique
CORDIC	COordinate Rotation DIgital Computer
CTF	Coarse-To-Fine
CTR	Cognitive Transceiver
DA	Data-Aided
DSP	Digital Signal Processor
FFT	Fast Fourier Transform
FMC	FPGA Mezzanine Card
FPGA	Field-Programmable Gate Array
GPP	General Purpose Processor
HDL	Hardware Description Language
HIL	Hardware In the Loop
HLS	High Level Sythesis
IFFT	Inverse Fast Fourier Transform
LCR	Level-Crossing Rate
LLF	Log-Likelihood Function
LS	Least Square
LTE	Long-Term Evolution
LTE-A	Long-Term Evolution Advanced

LUT	Look-Up Table
MAC	Multiply And Accumulate
MIMO	Multiple Input Multiple Output
ML	Maximum Likelihood
MSA	Machine Sequentielle Algorithmique
NDA	Non-Data-Aided
NMSE	Normalised Mean Square Error
OFDM	Orthogonal Frequency Devision Multiplexing
PCI	Peripheral Component Interconnect
PLB	Processor Local Bus
PSD	Power Spectrum Density
QAM	Quadrature Amplitude Modulation
QoE	Quality of Experience
RF	Radio Fréquence
RTL	Register Transfer Level
SDR	Software Defined Radio
SIMO	Single Input Multiple Output
SNR	Signal-to-Noise Ratio
USRP	Universal Software Radio Peripheral
VHDL	VHSIC Hardware Description Language
VLSI	Very Large Scale Integration



# Introduction générale

La croissance exponentielle de la demande de services de communication mobile omniprésents et à très large bande a poussé les fournisseurs de services et les industries à déployer de nouvelles infrastructures et techniques de traitement de signal dans le but de fournir des services hautement efficaces, sécurisés et ultra-fiables et d'améliorer la qualité d'expérience (QoE) de l'utilisateur final [1–3]. Des applications telles que la réalité augmentée/virtuelle, les véhicules autoguidés et les réseaux de capteurs sans-fil de très haute densité, seront les principaux moteurs de l'adoption des nouvelles technologies dans les futurs réseaux 5G [3, 4]. Cependant, plusieurs défis doivent être relevés pour les performances requises en ce qui concerne le débit, l'efficacité énergétique, la latence, la durée de vie des batteries, etc.

Cela implique, en effet, la nécessité d'intégrer une infrastructure flexible et modulaire, ainsi que la mise en place d'un certain niveau d'intelligence dans le réseau. Pour cette raison, la cognition est considérée comme une caractéristique fondamentale des futurs systèmes de communication mobile 5G. Cette caractéristique fournit un ensemble de règles et de stratégies qui permettent au système de communication de s'adapter automatiquement aux différentes techniques de transmission grâce à l'intégration d'un ensemble de capacités sensorielles avec lesquelles il prend conscience du contexte de transmission à tous les niveaux, des couches physique et réseaux, aux couches transport et applicatives. En fait, la conscience du contexte, ou en anglais « context-awareness », est considérée comme une réponse à un ensemble d'informations liées au contexte telles que l'état du réseau, la vitesse du mobile, le niveau de batterie, la consommation d'énergie, etc.; informations qui sont obtenues grâce à des entités sensorielles, physique ou virtuelles, embarquées au préalable dans le système sous-jacent.

Récemment, une nouvelle architecture d'un émetteur-récepteur cognitif (CTR) a été conçue au sein de notre équipe Wireless Lab [5]. Cet émetteur-récepteur est capable d'adapter sa structure de traitement d'antennes et sa configuration interne d'une façon dynamique et optimale. Plus précisément, il est capable de sélectionner le meilleur triplet entre trois modes d'utilisation de pilotes, deux algorithmes d'identification de canal, et deux modes de détection de données. Notre objectif ultime est de concevoir un prototype matériel complet de CTR. Pour accomplir cette mission, nous suivons l'approche d'ingénierie largement adoptée « divide-and-conquer » dans laquelle les principaux modules du CTR sont validés séparément. Ce choix est motivé par le fait que l'intégration logicielle du nouveau CTR a montré des gains en débits considérables par rapport aux architectures traditionnelles. Ces gains peuvent atteindre jusqu'à 700% et 40%, respectivement, aux niveaux lien et système.

En outre, pour sélectionner le meilleur triplet de combinaison d'utilisation pilote, d'identification de canal et de mode de détection de données, le module de décision du CTR demande la connaissance au préalable de deux paramètres clés: le SNR et l'étalement Doppler. Par conséquent, l'estimation de ces deux paramètres est cruciale dans la construction du module sensoriel du CTR afin de lui fournir l'information suffisante sur le contexte pour qu'il puisse prendre une décision en temps réel, selon des règles établies lors de la phase d'apprentissage présentée dans [5].

Plusieurs estimateurs de ces deux paramètres ont été proposés dans la littérature, chacun ayant ses propres avantages et inconvénients. Cependant, ces estimateurs sont généralement présentés par un ensemble d'équations mathématiques dont les algorithmes sont décrits par des scripts en MATLAB ou Python. De ce fait, ils ne peuvent être intégrés et exécutés directement dans un système de communication réel. Pour faire face à ce problème, les plateformes radio-logicielles ont été largement adoptées pour le prototypage rapide des modules de communication sans-fil. Mis à part leur importante capacité de calcul, ces plateformes sont généralement basées sur des architectures flexibles, modulaires et reprogrammables, permettant ainsi la réduction du coût d'ingénierie et du temps de la mise du produit sur le marché; deux contraintes sévères exigées par les futurs systèmes de communication 5G.

C'est dans ce contexte que le présent mémoire s'insère. Nous présentons une méthode de prototypage rapide du module sensoriel du nouveau CTR sur une plateforme radio-logicielle. Plusieurs travaux d'implémentation matérielle et de validation sur les ondes des circuits numériques de trai-

tement de signal ont été réalisés au sein de notre équipe Wireless Lab [6, 7]. Le travail présenté dans ce mémoire se situe ainsi dans le cadre des travaux de prototypage et de « proof-of-concept » des algorithmes et des architectures principalement développés par des membres de notre équipe.

Le présent mémoire est structuré comme suit: le chapitre 1 est un chapitre introductif qui sert à présenter le concept de la radio-logicielle et les différentes techniques et outils de prototypage rapide des systèmes de communication de la nouvelle génération. Dans le deuxième chapitre, nous détaillons l'architecture du nouveau CTR et les formulations mathématiques des estimateurs du SNR et de l'étalement Doppler choisis. Au chapitre 3, nous présentons l'architecture proposée des estimateurs sous-jacents et les différentes techniques d'optimisation qui permettent d'achever le meilleur compromis entre complexité, efficacité énergétique, latence, et précision. Nous terminerons au chapitre 4 par discuter les différentes étapes d'implémentation et les évaluations de performances de l'architecture proposée et du prototype matériel développé.



# Chapitre 1

## Méthodologies de conception et de prototypage rapide des systèmes de communication sans-fil: Les plateformes radio-logicielles

### 1.1 Introduction

Les systèmes de communication de nouvelle génération promettent de mettre en œuvre une grande variété des nouvelles fonctionnalités comme l'amélioration de l'autonomie des batteries, la minimisation du temps de latence et l'assurance d'une communication de très-haut débit. Ainsi, les fournisseurs de service sont obligés de développer et d'intégrer de nouveaux algorithmes plus complexes dans le but de fournir ces nouvelles fonctionnalités à leurs clients. Cependant, ces algorithmes sont généralement représentés par des équations mathématiques ou des diagrammes en bloc, qui ne peuvent pas être intégrés directement dans un système de communication réel. C'est pourquoi, le prototypage rapide de ces algorithmes sur des plateformes radio-logicielles est considéré comme une solution efficace et à faible coût pour tester et valider leur fonctionnement et accélérer leur mise en marché.

Dans ce chapitre, nous présentons le principe de la radio-logicielle et son architecture matérielle et logicielle. Ensuite, nous introduisons les différentes méthodologies et outils de conception matérielle. Finalement, nous présentons notre station expérimentale pour le prototypage rapide et la validation du module sensoriel du CTR.

## 1.2 Présentation de la radio-logicielle

### 1.2.1 Principe de la radio-logicielle

La radio-logicielle, ou en anglais Software Radio, est un concept technologique introduit par Joseph Mitola dans les années 90s [8]. Elle est définie comme un système de communication sans-fil configurable qui utilise plusieurs techniques de traitement de signal sur des plateformes numériques programmables. Dans une radio-logicielle, la configuration de la fréquence porteuse, de l'accès au réseau, de la bande passante du signal, et de la modulation se fait grâce à un logiciel. Autrement dit, la radio-logicielle vise à dématérialiser complètement l'interface radio pour exécuter des fonctions différentes selon le type d'application [9–11]. L'évolution de la radio matérielle à la radio logicielle est illustrée dans la figure 1.1. Nous remarquons à partir de la figure 1.1 que seuls la conversion A/N et N/A, les mélangeurs, les amplificateurs et les filtres à haute fréquences sont exécutés par des circuits matérielles tandis que tout le traitement en bande de base, à savoir la démodulation, la gestion des trames et la gestion applicatives sont effectués grâce à un logiciel.

La radio-logicielle tire profit de sa flexibilité qui lui permet de s'adapter aux différents protocoles de communications sans-fils et de basculer entre plusieurs formes d'onde, éventuellement dans différentes bandes spectrales, sans avoir besoin de changer le matériel. Cette caractéristique lui permet aussi de répondre aux besoins et aux exigences des futurs systèmes sans-fils, tels qu'offrir une communication à faible latence et assurer un service à très haute vitesse, n'importe où, n'importe quand, et peu importe l'environnement de transmission. Ceci explique bien la nécessité d'une recherche profonde dans le domaine du développement des standards ouverts [12]. Un autre avantage de la radio-logicielle est la portabilité. En effet, grâce à ce principe, il est devenu possible de transférer, facilement, le design d'un composant d'un système à un autre, sans avoir besoin de le reproduire à nouveau.

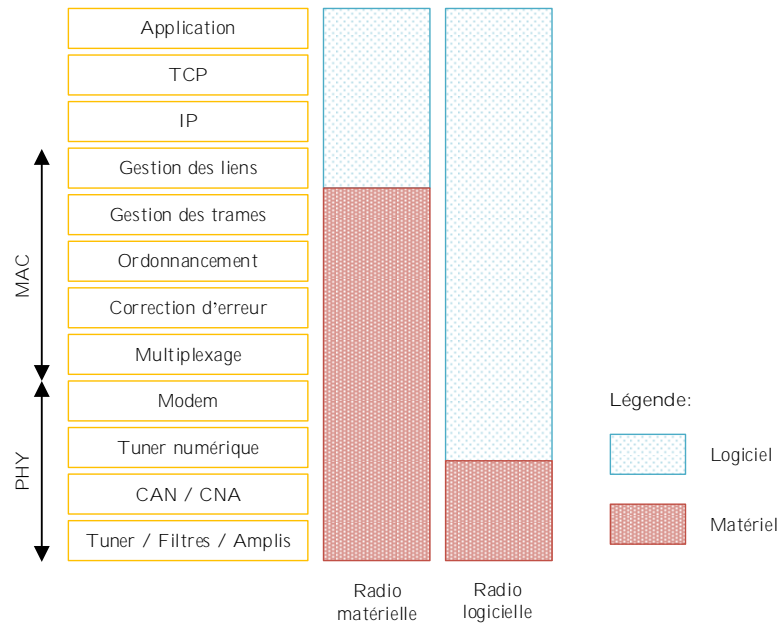


Figure 1.1 – Évolution de la radio-matérielle à la radio-logicielle.

### 1.2.2 Architecture de la radio-logicielle

Entre la radio matérielle et la radio cognitive (l'évolution ultime de la radio-logicielle) il existe plusieurs classes intermédiaires. Cette classification, élaborée par le *Wireless Innovation Forum* (WInnF) et représentée dans le tableau 1.1, est établie selon le degré de reconfiguration; c'est-à-dire selon la répartition des différentes opérations exécutées dans les parties matérielles et logiciels du système.

Dans ce travail, nous nous intéressons à la catégorie II de la radio-logicielle: la radio définie par logiciel, dont son architecture est représentée dans la figure 1.2 [13].

Un système basé sur la radio définie par logiciel est constitué de:

- **Tête RF analogique:** elle représente les parties frontales du système qui se composent de filtres, amplificateurs de puissance à large bande et à faible bruit, mélangeurs, coupleurs, et oscillateurs locaux à fréquence intermédiaire. Les frontaux RF sont configurables et modulaires permettant de s'adapter aux différentes bandes de fréquences;

Tableau 1.1 – Classification des radio-logicielles selon leur degré de reconfiguration

Catégorie	Dénomination	Degré de reconfiguration
0	Radio matérielle	Reconfiguration se fait seulement par échange des composants; implémentée; architecture matérielle complète.
I	Radio contrôlée par logiciel	Reconfiguration très limitée à quelques paramètres, tels que la puissance d'émission, le gain d'amplification, interconnexions, etc.
II	Radio définie par logiciel	Contrôle logiciel total de la couche physique (forme d'ondes, (dé)modulations, filtrages, fréquences, bande passante, etc.).
III	Radio logicielle idéale	À ce niveau, le système entier est reconfigurable. La conversion analogique est effectuée uniquement au niveau de l'antenne.
IV	Radio logicielle ultime	Le système est entièrement programmable et comprends tout type de trafic d'information.

- **Étage de conversion:** il assure la conversion analogique/numérique avec le CAN, ou la conversion numérique/analogique avec le CNA. La performance de cet étage est mesurée par la fréquence d'échantillonnage et la résolution numérique;
- **Étage numérique:** il assure la mise en forme du spectre et le traitement numérique en bande de base, voire la (dé)modulation, décodage/encodage, etc.;
- **Section logicielle:** elle permet de contrôler et configurer les différentes sections via un logiciel.

Dans l'étage numérique, les opérations sont exécutées sur des processeurs de calcul à haute performance, tels que les processeurs de traitement numérique du signal (DSPs), les circuits intégrés pour les applications spécifiques (ASIC), les circuits intégrés programmables (FPGA), et les processeurs à usage général (GPP). Le choix de la meilleure solution matérielle pour la mise en œuvre d'une plateforme radio définie par logiciel est une tâche importante pour les ingénieurs de prototypage. Chaque solution présente un ensemble d'avantages et d'inconvénients. Des éléments



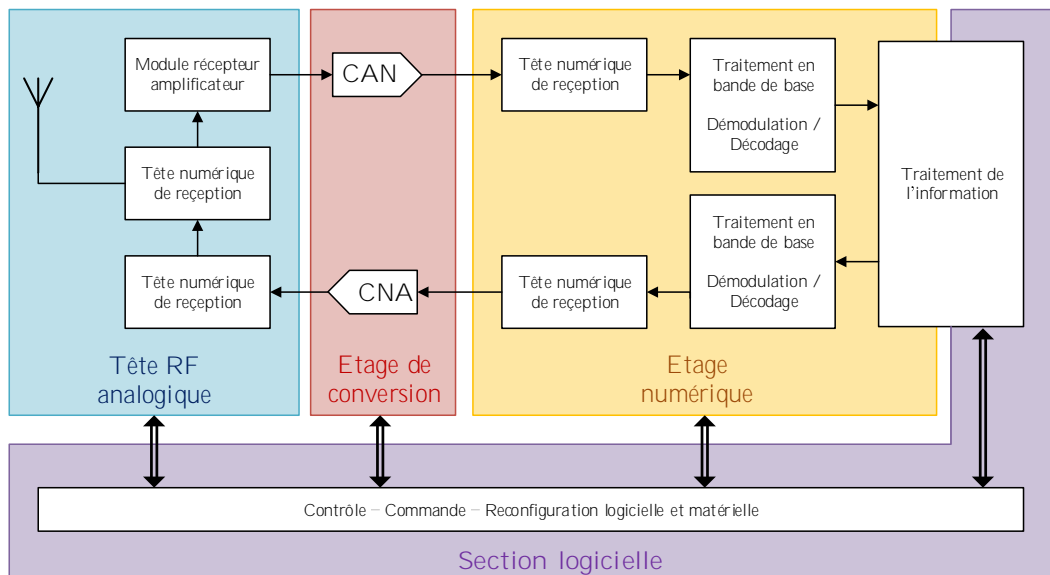


Figure 1.2 – Architecture globale de la radio définie par logiciel.

tels que le coût, la vitesse de traitement, et la flexibilité, ainsi que la puissance consommée, tous doivent être pris en considération lors du choix de la solution matérielle.

Dans la littérature ouverte, plusieurs études comparatives entre les différentes solutions matérielles ont été réalisées [14–17]. Grâce à leur flexibilité et leur capacité énorme de calcul, les FPGAs ont été fortement recommandées dans ces travaux pour construire des plateformes radio-logicielles de haute performance. C’est pourquoi durant les dernières années plusieurs plateformes radio-logicielles à base de FPGA ont pu voir le jour. Nous trouvons des plateformes destinées pour des applications militaires, d’autres qui sont utilisées dans l’industrie, et certaines sont réservées pour la recherche et développement. À titre d’exemple, nous citons la plateforme « bladeRF », la « RTL-SDR », la « miniBEE » et la « nanoBEE » de BEEcube inc., et la plateforme « USRP » de Ettus, etc. Le choix de la plateforme cible revient principalement à trouver le meilleur compromis entre le coût et la performance en termes de la largeur de bande couverte, la résolution des convertisseurs, et le nombre des frontaux disponibles. Une autre considération moins importante est le niveau d’abstraction et les outils de conception qui peuvent influencer en quelque sorte sur le coût d’ingénierie et le temps de mise en marché.

Nous présenterons dans la partie suivante les différentes méthodologies et les outils de conception des prototypes sur des plateformes radio-logicielles à base de FPGA.

## 1.3 Méthodologies et outils de conception matérielle: de l'algorithme à l'architecture

Les chercheurs et les développeurs d'algorithmes en communication sans-fil toujours ont tendance à penser seulement dans un cadre mathématique, dont l'algorithme, d'un point de vue du traitement de signal, n'est qu'un ensemble d'équations. La modélisation de tels algorithmes est généralement réalisée avec une représentation à virgule flottante, même quant une plage numérique limitée est suffisante pour certaines applications. Ceci n'est pas acceptable pour la conception des architectures VLSI et l'implémentation de ces algorithmes dans des circuits numériques à faible coût. Par conséquent, la forte interaction mutuelle entre les développeurs des algorithmes (ou les chercheurs) et les concepteurs d'architectures impose une collaboration étroite et précoce entre eux. Ceci explique parfaitement la nécessité de l'adoption des méthodologies de conception et l'utilisation des outils dédiés qui facilitent en quelque sorte le transfert technologique de l'algorithme à l'architecture, avec le moindre coût d'ingénierie.

### 1.3.1 Méthodologies de conception

Le prototypage rapide des différents modules des systèmes de communication consiste à intégrer son algorithme dans un système qui s'exécute sur un prototype matériel. Pour ce faire, plusieurs stratégies ont été proposées dans le but d'accélérer la phase de conception et de faciliter la tâche de simulation, de débogage, et d'intégration. Bien qu'elles présentent plusieurs points communs, chaque stratégie a ses propres avantages et inconvénients. Selon le sens de la démarche intellectuelle, nous pouvons distinguer deux approches de conception: l'approche ascendante et l'approche descendante [18–20].

#### **Approche de conception ascendante: « Bottom-Up »**

L'approche « Bottom-up », littéralement appelée « de bas en haut », est une méthode de conception et de développement qui consiste à consolider et à opérer une synthèse à partir des éléments de base initialement identifiés. En d'autres termes, elle privilégie la conception et la validation des modules élémentaires avant ceux du niveau hiérarchique plus haut.

L'approche ascendante présente plusieurs avantages. En effet, elle permet de mettre l'accent sur la livraison initiale des modules élémentaires et de commencer, immédiatement, à développer les parties critiques du système. Avec cette méthodologie, la définition et la vérification des interactions entre les différents modules, d'un point de vue système, sont réalisées dans une étape tardive dans le cycle de conception. Par conséquent, la moindre erreur dans la synthèse des différents éléments, bien qu'ils sont unitairement vérifiés et validés, peut provoquer plusieurs itérations additionnelles, qui sont à leur tour, coûteuses en termes de temps de mise du produit au marché.

En outre, bien que l'approche « Bottom-up » est basée sur le principe de la réutilisation des modules élémentaires, la migration d'un environnement de conception/vérification à un autre représente toujours un vrai obstacle. Néanmoins, la réutilisation peut ne pas être optimale au cas où un nouveau système aurait besoin d'une architecture différente à celle conçue dès le départ.

### **Approche de conception descendante: « Top-Down »**

L'approche de conception descendante est radicalement opposée à la celle de « Bottom-Up ». Cette méthodologie consiste tout d'abord à faire une analyse macro-architecturale et décomposer le système en éléments plus détaillés, pour déboucher un aperçu sur ses sous-systèmes, tout est dans le même esprit de l'ingénierie inverse (*reverse engineering* en anglais). Autrement dit, en suivant cette approche, les concepteurs commencent, tout d'abord, à formuler un aperçu sur le système en spécifiant, mais sans détailler, les sous-systèmes de premier niveau hiérarchique. Ensuite, chaque sous-système est affiné plus en détail, jusqu'à ce que la spécification se réduit à des fonctions de base.

Grâce à cette démarche, tout problème peut être conceptualisé rapidement et subdivisé en sous-problèmes aisément manipulables. Elle permet, par conséquent, de tracer une vue globale du produit final et d'estimer, bien qu'approximativement, son coût et sa complexité. En outre, un autre avantage de l'approche « Top-Down » est celui de la minimisation des itérations de test et de vérification des interactions entre les différents sous-systèmes, puisque leurs spécifications sont toujours décrites avant de détailler chacun entre eux.

### 1.3.2 Outils de conception

Dans la microélectronique, la description d'un algorithme donné peut être traitée de différents niveaux d'abstraction [18]:

- **Niveau système:** spécification des relations entrées/sorties;
- **Niveau architecture:** modélisation fonctionnelle des sous-tâches;
- **Niveau transfert de registre:** description du transfert des données et des opérations internes;
- **Niveau logique:** établissement des tables de vérité et des diagrammes d'état;
- **Niveau circuit:** représentation par des fonctions de transfert.

Par ailleurs, il existe trois perspectives de conception matérielle: comportementale, structurelle, et physique. Nous représentons dans la figure 1.3 le diagramme de Gajski-Kuhn qui résume ces trois perspectives, avec lesquelles nous pouvons regarder le processus de conception d'un système numérique.

Dans ce contexte, plusieurs outils de conception ont été proposés. Chaque outil s'intéresse à une ou plusieurs perspectives à partir de différents niveaux d'abstraction, dans le but d'automatiser le processus de conception et la génération du programme matériel à implémenter.

Globalement, il existe deux grandes familles d'outils de conception: la famille « Register Transfer Level » (RTL) et la famille « High Level Synthesis » (HLS) [21].

#### Outils de conception basés sur RTL

Avec cette famille d'outils, le comportement et la structure d'un système numérique sont modélisés sur le plan de flots des signaux numériques et de transferts de données entre les registres et les opérations logiques qui manipulent ces signaux. Généralement, cette modélisation est réalisée grâce aux langages HDLs, à savoir le VHDL et le Verilog. Cependant, cette technique présente plusieurs inconvénients comme la nécessité de connaître au préalable le matériel cible, la complexité énorme du code descriptif, et la difficulté de trouver un environnement collaboratif entre les ingénieurs de traitement du signal et les concepteurs des circuits numériques, etc. [18].

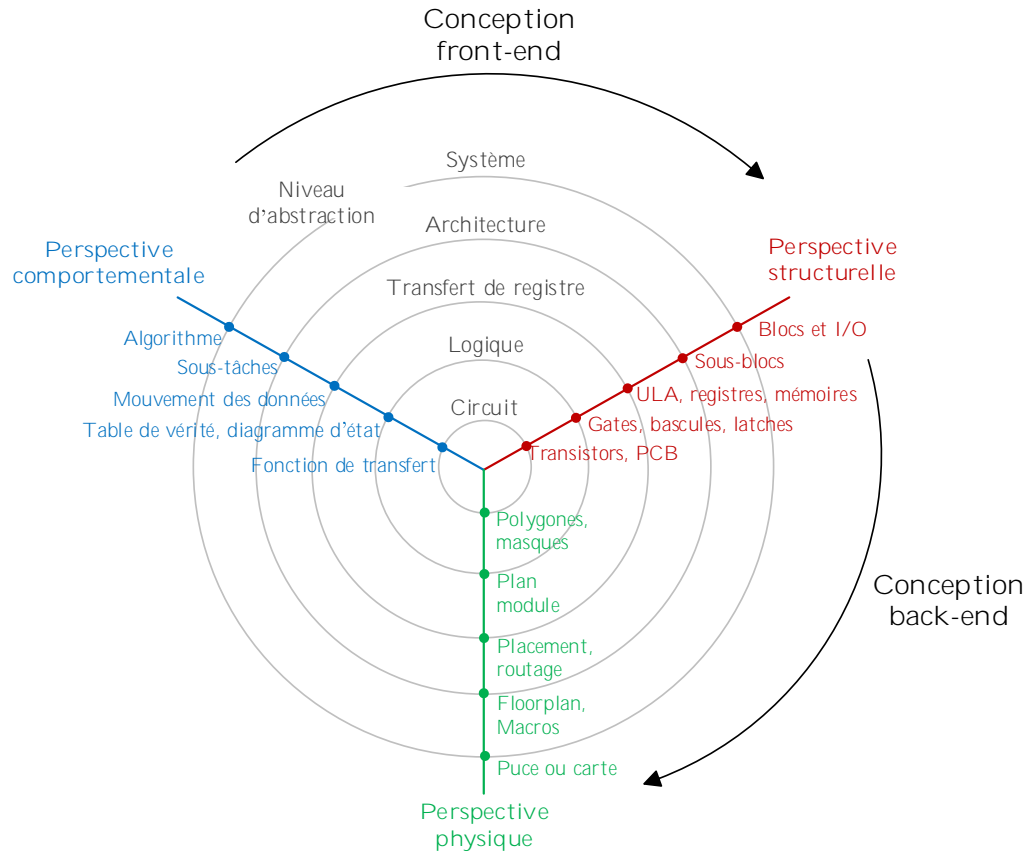


Figure 1.3 – Diagramme de Gajski-Kuhn: perspectives et niveaux d'abstraction pour la conception des systèmes numériques [18].

Les environnements de développement les plus connus basés sur la conception RTL sont Xilinx ISE/Vivado Design Suite<sup>1</sup> et Quartus<sup>2</sup>.

### Outils de conception basés sur HLS

Les outils de conception basés sur HLS offrent la possibilité de décrire le comportement et la structure d'un système numérique avec des langages de haut niveau d'abstraction qui seront traduits en langage matériel (HDL) pour l'implémenter finalement sur des FPGAs [18, 21]. Le principal avantage de cette technique est de réduire le volume total du projet ce qui, par conséquent, facilite la tâche de débogage et réduit le temps de conception.

1. Pour les FPGAs de Xilinx, Inc.

2. Pour les FPGAs de Altera, Inc.

Aujourd’hui, il existe deux catégories de conception HLS: les langages *C-like* et les *frameworks* de conception basée sur modèle (MBD). La première catégorie consiste à décrire l’algorithme avec des langages de programmation de niveau d’abstraction plus haut que les HDLs, tels que les langages C, C++, SystemC [22], SpecC [23], Handel-C [24], etc. La deuxième catégorie s’agit d’outils avec des interfaces graphiques qui permettent de décrire le comportement et la structure d’un système numérique avec des modèles graphiques instanciés à partir des bibliothèques pré-définies. Dans cette catégorie, on trouve MATLAB Simulink HDL Coder, Xilinx System Generator, GNU Radio, NI LabView, etc.

## 1.4 Présentation de la station expérimentale

Notre station de travail se compose de la plateforme « miniBEE4 » de BEEcube et de l’émulateur de canal « EB Prosim FS8 » de Keysight Technologies<sup>3</sup>.

### 1.4.1 La miniBEE4 de BEEcube

La miniBEE4 est une plateforme radio-logicielle à base de FPGA qui est en communication *Peripheral Component Interconnect* (PCI) directe avec un processeur embarqué de type Intel i7 et une carte FPGA Mezzanine modèle 111 (FMC111). Le diagramme en bloc de l’architecture matérielle interne de la miniBEE4 est représenté dans la figure 1.4.

Le processeur embarqué représente la partie logicielle de la plateforme qui fournit un environnement de développement complet conçu autour d’une distribution Linux CentOS, appelé BEEcube Platform Studio (BPS). Il permet d’accéder en lecture et en écriture, via les pilotes de NectarOS, aux registres et aux mémoires de la FPGA ainsi que de configurer la carte FMC111 à partir de l’environnement MATLAB.

La FMC111 représente la partie RF frontale de la plateforme miniBEE4. Elle comprend des convertisseurs numérique/analogique et analogique/numérique de haute résolution (14-bit et 16-bit respectivement), des mélangeurs, et des amplificateurs à faible bruit.

---

3. Anciennement Agilent, Inc.

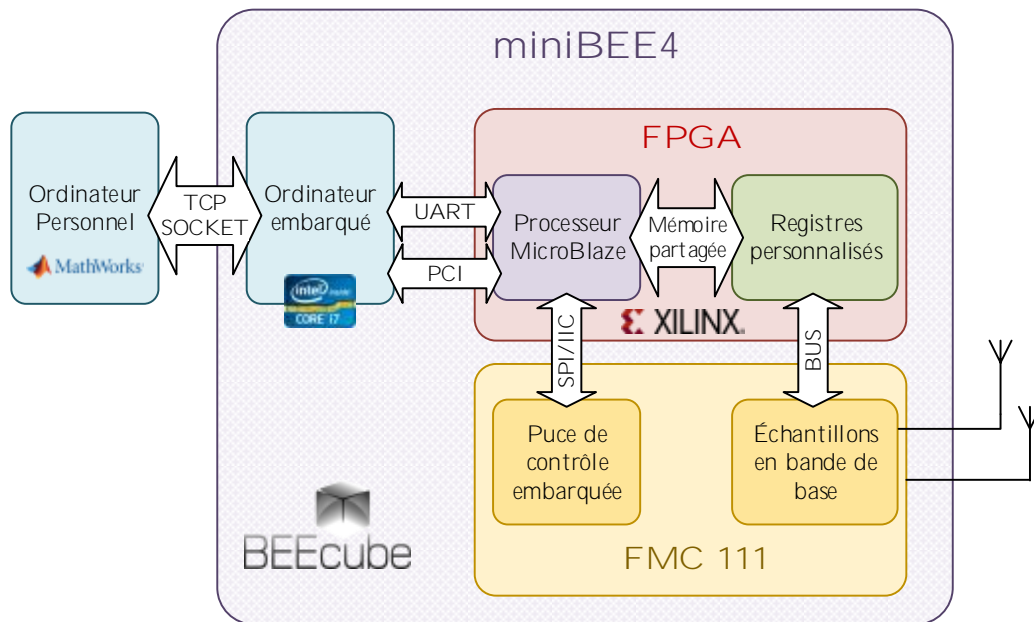


Figure 1.4 – Architecture interne de la plateforme radio-logicielle miniBEE4 [25].

La conception matérielle est réalisée avec l’outil Xilinx System Generator. En outre, BEEcube fournit un ensemble de bibliothèques pour ajouter au design des mémoires et des registres personnalisés par l’utilisateur selon sa propre logique. Ensuite, cette logique est synthétisée sous la forme d’un périphérique interne, auquel le contrôleur de la FMC111, les blocs mémoires (BRAM) ainsi que les registres personnalisés sont attachés. Tous ces composants sont connectés au processeur embarqué MicroBlaze à travers un bus local de processeur (PLB).

Un autre avantage de l’environnement BPS est la génération automatique du fichier de programmation de la FPGA. Le flot de conception et d’implémentation est représenté dans la figure 1.5.

### 1.4.2 Émulateur de canal EB Prosim FS8

La méthode traditionnelle de l’expérimentation matérielle dans le domaine de la communication sans-fil consiste à réaliser plusieurs expériences sur un terrain de communication réel, que ce soit à l’intérieur ou à l’extérieur, urbain ou rural. Cette méthode nécessite un travail de mesure préliminaire, ainsi qu’une connaissance parfaite et un contrôle total des différentes conditions de

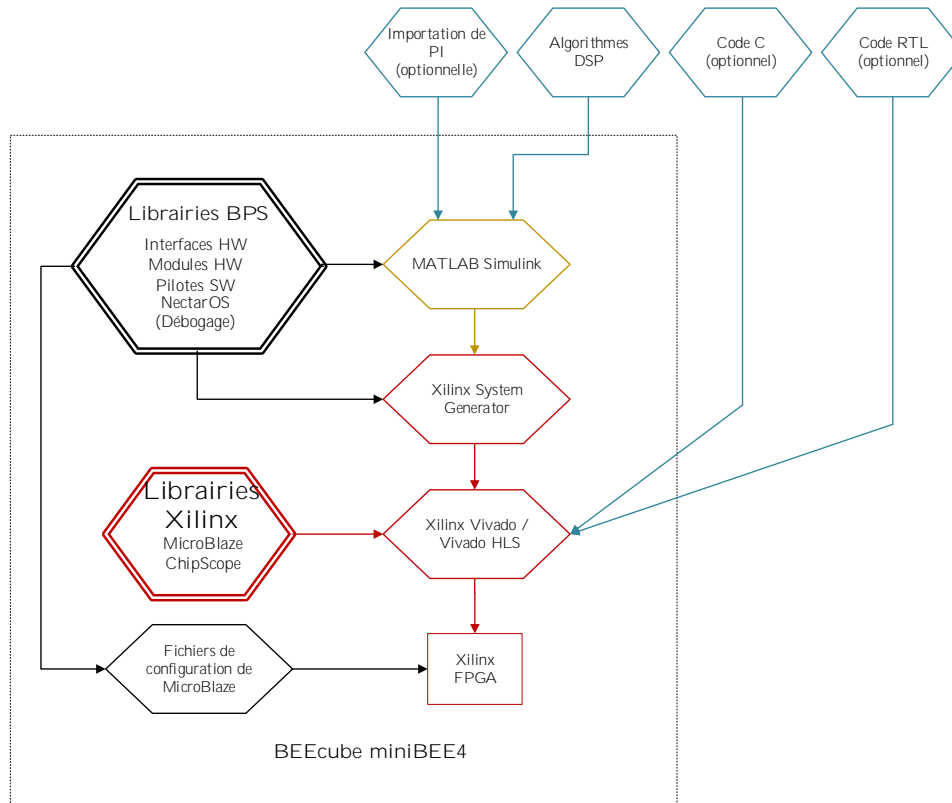


Figure 1.5 – Flot de conception et d’implémentation du design sur la plateforme miniBEE4 [25].

propagation. D’autre part, il est presque impossible de reproduire ou de répéter la même expérience plusieurs fois puisque la variation du canal sans-fil, due à la mobilité des personnes et des objets physiques, reste incontrôlable. C’est pour cela que cette technique est considérée coûteuse et moins précise pour effectuer une expérimentation idéale d’un système de communication sans-fil de nouvelle génération.

Pour faire face à ce problème, une nouvelle méthode d’expérimentation a été proposée. Elle consiste à émuler le canal sans-fil avec des bancs de test matériels et/ou logiciels, idéalement reconfigurables et évolutifs. Dans l’état de l’art, on trouve de nombreuses plateformes d’émulation du canal de propagation [26–29]. La plateforme d’émulation utilisée dans ce travail est la EB Propsim FS8 de Keysight Technologies (Figure 1.6). Avec ses 8 canaux RF, cette plateforme permet d’émuler d’une manière réaliste plusieurs modèles déterministes et statistiques des canaux sans-fil. Elle offre aussi la possibilité de configurer manuellement ou choisir un modèle de canal standardisé comme celui de GSM, UMTS, LTE, etc.





Figure 1.6 – L’émulateur de canal EB PropSim FS8 [30].

Grâce à une interface graphique simple et ergonomique, l’émulateur de canal EB PropSim FS8 permet de définir des scénarios de test et modifier pendant l’expérience plusieurs paramètres tels que le type d’évanouissement, la vitesse du mobile, le niveau du SNR, etc. À la fin de la configuration, les valeurs de tous les paramètres de l’émulation sont enregistrées dans un fichiers *\*.xml* qui assure la répétabilité et la contrôlabilité de l’expérience.

## 1.5 Conclusion

Dans ce chapitre, nous avons introduit le concept de la radio-logicielle et les solutions qu’il apporte pour le prototypage rapide des systèmes de communication sans-fil. Dans une deuxième partie, nous avons présenté les différentes méthodologies de conception des architectures matérielles sur des plateformes radio-logicielles à base de FPGA, ainsi que les outils permettant d’automatiser la tâche de synthèse, d’optimisation et de programmation de ces FPGAs. Finalement, nous avons présenté notre station expérimentale sur laquelle nous avons conçu, implémenté, et testé le module sensoriel du CTR.

Dans le chapitre suivant, nous allons présenter l’architecture du nouveau CTR ainsi que les règles de décision pour sélectionner le meilleur triplet du mode d’utilisation des pilotes, l’algorithme d’estimation du canal, et le mode de détection. Ensuite, nous allons argumenter notre choix d’estimateurs qui composent le module sensoriel de CTR, avant que nous développons, à la fin, leurs formulations mathématiques.



## Chapitre 2

# Présentation du CTR et des estimateurs de l'étalement de Doppler et du SNR

### 2.1 Introduction

Dans le chapitre précédant, nous avons introduit le principe de la radio-logicielle comme une solution de prototypage rapide des systèmes de communication sans-fil de nouvelle génération 5G. Grâce à ce concept, plusieurs technologies de pointe ont pu voir le jour, la plus importante est la radio cognitive qui permettra à ces systèmes de prendre conscience des différentes conditions de propagation et du contexte général de transmission. Pour cette raison, elles emploient des modules sensoriels semblables à nos organes de sens qui consistent généralement en estimateurs de paramètres de canal sans-fil avec lesquels ils peuvent extraire des informations pertinentes sur le contexte. Dans ce cadre, la littérature récente regorge des solutions et des techniques d'exploitation de la cognition pour concevoir des systèmes intelligents et adaptatifs.

Dans ce chapitre, nous présentons l'émetteur-récepteur cognitif sur lequel nous travaillons au sein de notre équipe de recherche Wireless Lab. Ensuite, nous décrivons l'architecture de son module sensoriel en argumentant le choix des différents algorithmes d'estimation et en développant leurs formulations mathématiques.

## 2.2 Présentation de l'émetteur-récepteur cognitif CTR

La radio cognitive, ou aussi appelée radio intelligente, est une radio-logicielle avancée dont les modules sont capables de s'adapter automatiquement et correctement à la variation des conditions de propagation, sans aucune intervention externe [31]. Ce principe, introduit par Joseph Mitola comme étant l'évolution ultime de la radio-logicielle, a été étudié dans plusieurs travaux de recherche pendant la dernière décennie [32–34]. Cependant, la majorité de ces travaux ont abordé la radio cognitive d'un point de vue de l'allocation dynamique et efficace du spectre.

Récemment, un nouveau travail a été proposé dans [5] considère l'intelligence en introduisant la cognition comme outil de basculement automatique entre plusieurs configurations internes, autres que l'allocation du spectre. Dans ce travail, l'auteur a proposé une architecture d'un nouvel émetteur-récepteur cognitif capable de faire la sélection du meilleur triplet: l'utilisation des symboles pilotes, l'algorithme d'estimation de canal, et le mode de détection, selon des règles de décision initialement établies.

Dans ce qui suit, nous présentons l'architecture du CTR, ainsi que les règles de décision qui lui permettent de sélectionner la meilleure configuration.

### 2.2.1 Architecture globale du CTR

En admettant une transmission sans-fil dans une configuration d'antenne SIMO  $1 \times 2$  (une antenne à l'émission et deux à la réception), le CTR change son architecture interne selon les conditions de propagation en choisissant le meilleur:

- **mode d'estimation du canal** entre l'estimation assistée par pilote (DA), l'estimation hybride (NDA w. pilots), et l'estimation autodidacte (NDA);
- **algorithme d'estimation du canal** entre le célèbre algorithme à moindres-carrés (LS) et celui à maximum de vraisemblance (ML);
- **mode de détection** entre le mode différentiel et le mode cohérent.

L'architecture du CTR est conçue autour d'une chaîne de transmission basée sur la technique du codage par répartition des fréquences orthogonales (OFDM) dont la disposition tempo-fréquentielle des symboles transmis est adaptée à la configuration mise par la 3GPP dans son standard LTE

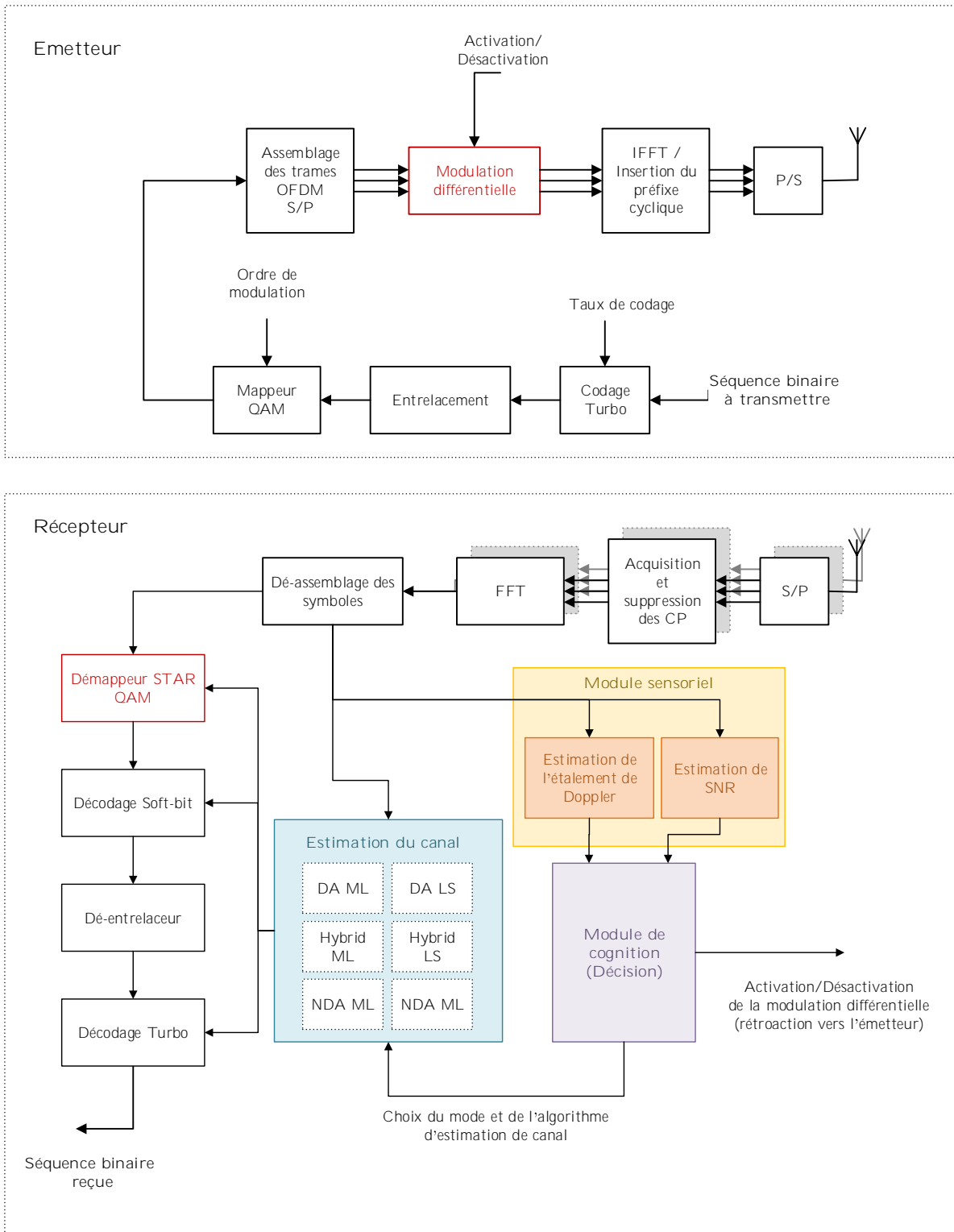
[35]. Le diagramme en bloc d'un point de vue système de l'architecture du CTR est illustré dans la figure 2.1.

L'émetteur du CTR, dont le but est de moduler et adapter l'information au canal pour assurer une performance maximale au niveau du récepteur, se compose de:

- **Codage Turbo**: il s'agit d'un codage correcteur d'erreur qui ajoute certaines redondances au message transmis afin de le rendre plus robuste aux bruits introduits par le canal. Le taux de codage est directement relié à l'information sur la qualité du canal (CQI) selon une correspondance présentée dans [35].
- **Entrelacement**: il s'agit de mélanger les bits codés pour éviter les longues séquences binaires de même valeur.
- **Mappeur QAM**: il permet d'associer à chaque code binaire un symbole complexe bien déterminé selon l'ordre de modulation. Le CTR suppose une modulation d'amplitude en quadrature d'ordre 4, 16, 64, et 256.
- **Assemblage des trames OFDM**: ce bloc permet d'assembler les symboles QAM selon la disposition tempo-fréquentielle définie par le standard LTE.
- **Modulation différentielle**: elle consiste à coder (ou moduler) la différence de phase entre les symboles successifs transmis.
- **IFFT / Insertion du préfixe cyclique**: brièvement, ce module assure la modulation multi-porteuse (MCM), grâce à l'IFFT (transformée de Fourier inverse), et l'ajout du préfixe dans le but de conserver l'orthogonalité entre les différentes sous-porteuses.

Du côté du récepteur, plusieurs modules, avec différents niveaux de complexités, ont été intégrés. Principalement, il se compose d'une chaîne de traitement qui exécute la fonction inverse de l'émetteur (comme la FFT, le dé-assemblage, la démodulation, le décodage turbo, etc.). Par ailleurs, les autres modules servent à extraire l'information sur l'environnement de propagation afin de détecter correctement les données reçues. Ces modules sont:

- **Démappeur STAR QAM**: il permet d'extraire une décision souple (soft) des bits modulés par différence d'amplitude (ou de phase) à partir de deux symboles consécutifs.
- **Module sensoriel**: ce module joue le rôle d'un organe sensoriel avec lequel le CTR prend conscience des conditions de propagation. Il est composé d'un estimateur du SNR et un



estimateur de l'étalement de Doppler. Dans ce travail, nous nous intéressons à concevoir un prototype de ce module et l'implémenter sur une plateforme radio-logicielle<sup>1</sup>.

- **Module de cognition:** il représente le cerveau du CTR qui permet de faire la sélection du meilleur triplet: l'utilisation des symboles pilotes, l'algorithme d'estimation de canal, et le mode de détection.
- **Estimation du canal:** il s'agit de l'ensemble des algorithmes d'estimation du canal, avec les différents modes d'utilisation des symboles pilotes, entre lesquels le CTR choisira le meilleur algorithme/mode dans le but de maximiser le débit.

### 2.2.2 Règles de décision

Comme dans tout système intelligent, y compris le cerveau humain, la prise de décision est basée sur des critères de sélection et sur une analyse des options qui conduisent vers un choix final. C'est pour cela que l'apprentissage est une étape nécessaire et importante pour réussir à définir les critères de sélection et les différentes options possibles. En adoptant cette théorie et après plusieurs simulations exhaustives « off-line » au niveau lien, les règles de décision du CTR ont pu être tracées en fonction du SNR et de la vitesse du mobile. La figure 2.2 illustre la grille de décision du récepteur CTR avec les gains en termes de débit par rapport à la configuration traditionnelle: estimation assistée par pilotes avec l'algorithme à moindres carrés (DA LS) [5].

## 2.3 Choix d'estimateurs de l'étalement de Doppler et du SNR

Pour sélectionner le meilleur triplet, le CTR demande la connaissance *à priori* de deux paramètres clés: le SNR et la vitesse du mobile qui est directement reliée à l'étalement du Doppler. Ainsi, leur estimation représente une étape primordiale et nécessaire pour être conscient de l'état du canal sans-fil. En effet, le SNR donne information sur la qualité du canal alors que l'étalement de Doppler permet d'avoir une idée sur sa variabilité.

---

1. Il est à noter ici que d'autres capacités sensorielles vont être intégrées dans ce module dans le futur afin de doter le CTR d'une connaissance plus fine de son environnement. En effet, pour cette raison, l'équipe Wireless Lab a développé de nombreux estimateurs pour tous les paramètres clés du canal [36–44] qui sont spécialement conçus pour les systèmes de communication futurs tels que CTR (idem., antennes multiples, modulations d'ordres supérieurs, signaux turbo-codes, etc.)

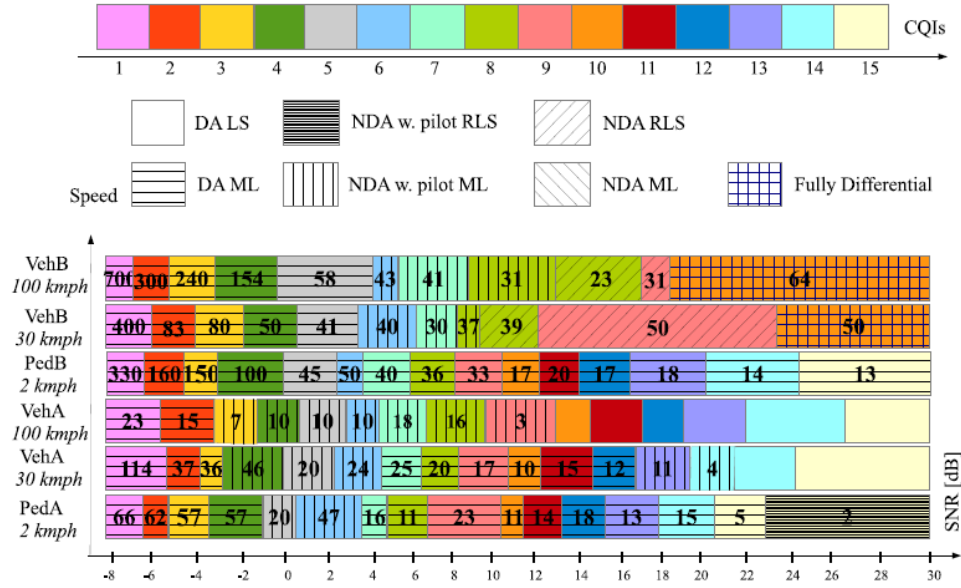


Figure 2.2 – Règles du décision du CTR basées sur l’estimation de canal ML et LS [5].

### 2.3.1 Choix de l’estimateur de l’étalement du Doppler

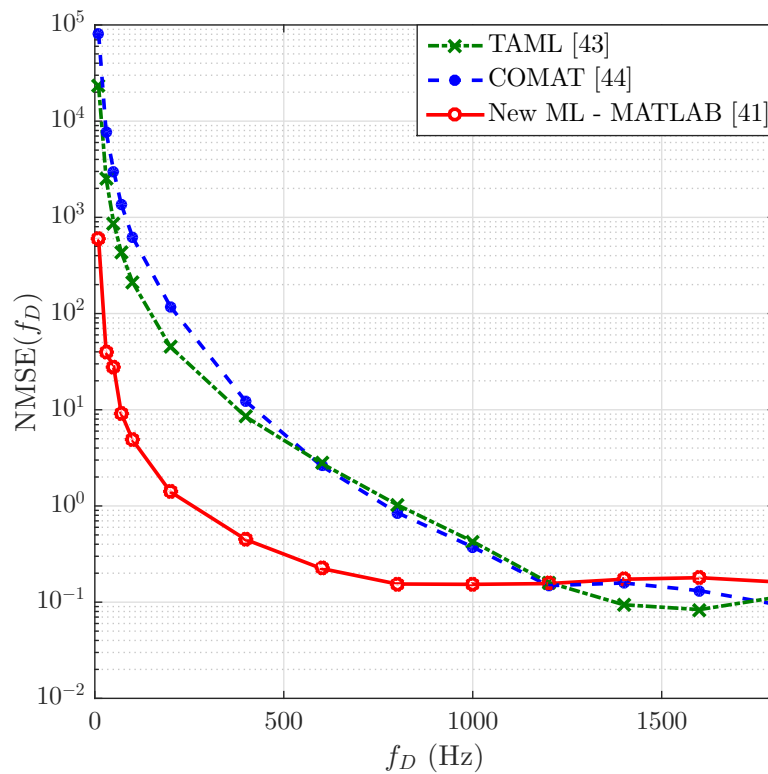
Dans la littérature ouverte, plusieurs techniques d’estimation de l’étalement de Doppler ont été proposées. Globalement, nous pouvons distinguer quatre classes d’estimateurs: level-crossing rate (LCR) [45, 46], power spectrum density (PSD) [47, 48], covariance-based [49–51], et maximum likelihood (ML) [52–54].

Les estimateurs basés sur la technique LCR exploitent la relation entre le taux d’évanouissement et l’étalement de Doppler en comptant le nombre de passages par un niveau « seuil » bien défini. L’approche basée sur la PSD extrait un estimé de l’étalement de Doppler à partir de la densité spectrale de la puissance, alors que celle basée sur la covariance trouve cet estimé à partir de l’autocorrélation du signal reçu. Autrement dit, ces trois techniques d’estimation n’extraient pas les informations sur l’étalement de Doppler à partir du signal reçu lui-même, mais plutôt de ses statistiques. Par conséquent, elles nécessitent un grand nombre d’échantillons reçus pour établir des statistiques suffisantes et pour répondre aux performances d’estimation cible (c’est-à-dire pour avoir une estimation fiable et précise). En outre, la plupart de ces techniques présente des opérations mathématiques très complexes et très lourdes telles que l’inversion ou la multiplication matricielles et, par conséquent, leur complexité de calcul est jugée très élevée d’un point de vue implémentation matérielle. Cependant, dans [53], un nouvel estimateur de l’étalement de Doppler basé sur la



technique d'estimation à maximum de vraisemblance a été proposé<sup>2</sup>. Cet estimateur, appelé « New ML », extrait l'information sur le Doppler directement du signal reçu, qui nécessite ainsi un petit nombre d'échantillons acquis au côté du récepteur. De plus, contrairement aux autres estimateurs, il ne présente aucune inversion ou multiplication matricielles grâce à un modèle d'approximation à deux raies fréquentielles pour la matrice de covariance du canal.

Motivés par ces faits, nous choisissons d'implémenter cet estimateur dans notre prototype du CTR car non seulement il présente un compromis complexité / précision attirant, mais il surpasse aussi de loin les techniques proposées dans la littérature, TAML [55] et COMAT [56], en matière de précision d'estimation, particulièrement dans le bas valeurs de Doppler<sup>3</sup> (Figure 2.3).



**Figure 2.3** – Comparative de l'estimateur choisi avec les benchmarks [55, 56] en terme de l'erreur quadratique moyenne normalisée (NMSE) de l'estimation de la fréquence de Doppler, à  $\text{SNR} = 0$  dB et avec 1000 échantillons reçus.

2. Cet estimateur est développé par des membres de notre équipe Wireless Lab au centre EMT-INRS.

3. Dans les futurs systèmes de communication 5G, la fréquence de Doppler normalisée  $f_D T_s$  sera extrêmement faible puisque ces systèmes fonctionneront à une très petite période de symbole  $T_s$  pour fournir des communications à très haut débit.

### 2.3.2 Choix de l'estimateur du SNR

En concordance avec le CTR lui-même, l'estimateur de l'étalement Doppler de [53] nécessite la connaissance au préalable du SNR instantané et de la puissance du bruit, qui sont également inconnus dans la pratique. Ainsi, plusieurs techniques ont été proposées dans le but de fournir des estimés précis de ces paramètres. Cependant, une majorité de ces techniques considère une transmission à travers des canaux sans-fil qui varient lentement ou même quasi-constantes, ce qui n'est pas le cas dans la réalité.

Pour ce but, parmi beaucoup d'autres, nous avons également développé un nouvel estimateur conjoint du SNR instantané et de la puissance du bruit basé sur la technique à maximum de vraisemblance [57]. Selon le mode d'utilisation des pilotes, cet estimateur est présenté dans ces trois configurations DA, hybride, et NDA. En raison de sa simplicité et des gains de performances énormes qu'il apporte en le comparant aux techniques proposées dans l'état de l'art, l'estimateur/démodulateur conjoint de [57] a été intégré dans le CTR. Sa version DA, en particulier, sera utilisée pour identifier le SNR instantané requis par le CTR afin de sélectionner dynamiquement le triplet de la meilleure combinaison d'utilisation pilote, d'identification de canal, et de mode de détection. Motivés par ces faits, nous décidons d'implémenter cet estimateur dans le module sensoriel du CTR.

Dans ce qui suit, nous présentons alors les formulations mathématiques de l'estimateur du SNR et de l'étalement de Doppler choisis.

## 2.4 Modèle du système et formulation mathématique des estimateurs choisis

### 2.4.1 Modèle du système

Nous considérons une transmission des symboles linéairement modulés (modulation M-aire) à travers un canal de Rayleigh à évanouissement plat,  $h(t)$ , et immergés dans un bruit blanc gaussien additif,  $w(t)$ . Le signal reçu en bande de base, échantillonné avec le taux  $T_s$  et après la synchroni-

sation temporelle et fréquentielle s'écrit:

$$y(nT_s) = h(nT_s)x(nT_s) + w(nT_s), \quad n = 1, 2, 3... \quad (2.1)$$

où  $n$  désigne l'indice temporel et  $x$  représente le signal transmis. La représentation discrète équivalente de l'observation est donnée par:

$$y[n] = h[n]x[n] + w[n], \quad n = 1, 2, 3... \quad (2.2)$$

Les estimateurs de l'étalement de Doppler et du rapport signal-à-bruit sont basés sur un mode assisté par pilote. Autrement dit, seuls les symboles reçus qui correspondent aux positions pilotes sont utilisés durant la tâche d'estimation. Selon le standard LTE [58], les symboles pilotes (ou aussi appelés symboles de référence) sont considérés constants et égaux au symbole unité (c'est-à-dire  $x(n_p T_s) = 1$ ). Dans ce cas, la séquence d'observation en bande de base s'écrit:

$$y[n] = h[n] + w[n], \quad n = 1, 2, 3... \quad (2.3)$$

En outre, les deux estimateurs sont construits sur le principe de subdiviser la fenêtre d'observation entière en plusieurs fenêtres d'approximation locale de taille  $N$ . Cette technique permet de capturer les variations temporelles du canal, qui sont généralement imprédictibles, en utilisant un nombre très petit des échantillons reçus. Soit  $M$  le nombre des fenêtres d'approximation, la  $m^{\text{ème}}$  séquence d'observation locale, dont la disposition est illustrée dans la figure 2.4, est donnée par:

$$y^{(m)}[n] = h[n] + w[n], \quad n = 1, 2, 3...N. \quad (2.4)$$

En considérant le modèle du système présenté dans (2.4), nous détaillons, dans les deux sous-sections suivantes, les formulations mathématiques de l'estimateur de l'étalement de Doppler ainsi que celui du rapport signal-à-bruit, respectivement, développé dans [53] et [57].

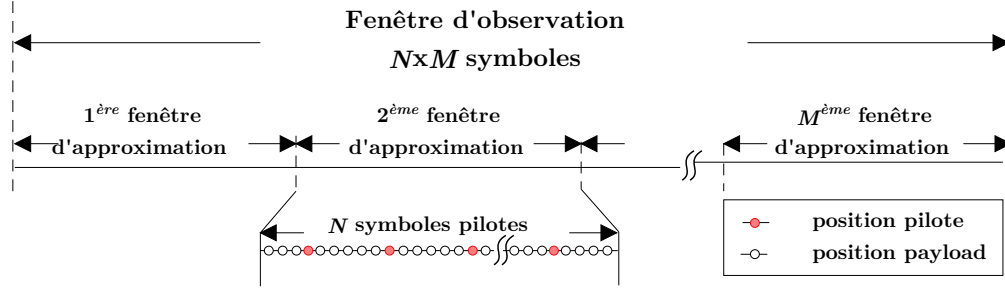


Figure 2.4 – Disposition temporelle des données et des symboles pilotes sur  $M$  fenêtres d'approximation locale.

## 2.4.2 Présentation de l'estimateur de l'étalement de Doppler

Dans le modèle du système de (2.4), l'information sur l'étalement de Doppler se trouve dans les coefficients de la matrice d'autocorrélation du canal. Comme nous l'avons mentionné précédemment, l'estimateur développé dans [53] est conçu autour de l'approximation en série de Taylor de second ordre de la matrice de covariance du canal, comme suit:

$$\mathbf{R}_h(\sigma_D) = \frac{\sigma_h^2}{2} \mathbf{A}(\sigma_D) \mathbf{A}^H(\sigma_D). \quad (2.5)$$

où  $\sigma_D$  et  $\sigma_h^2$  représente, respectivement, l'étalement de Doppler et la puissance du canal, et  $\mathbf{A}(\omega)$  est donné par:

$$\mathbf{A}(\omega) = [\mathbf{a}(-\omega) \quad \mathbf{a}(\omega)], \quad (2.6)$$

dans lequel le vecteur  $\mathbf{a}(\omega)$  est défini comme:

$$\mathbf{a}(\omega) \triangleq [1 \quad e^{j\omega T_s} \quad e^{j2\omega T_s} \quad \dots \quad e^{j(N-1)\omega T_s}]^T. \quad (2.7)$$

En se basant sur l'approximation présentée dans (2.5), et après quelques manipulations mathématiques (plus de détails en [53]), la fonction de log-vraisemblance (LLF) qui correspond à la  $m^{\text{ème}}$  séquence  $\mathbf{y}^{(m)} \triangleq [y^{(m)}[1], y^{(m)}[2], \dots, y^{(m)}[N]]^T$ , et paramétrée par le candidat de l'étalement de Doppler,  $\sigma_D$ , s'écrit:

$$L(\sigma_D) = -\text{Ln}(\psi(\sigma_D)) + \frac{1}{\hat{\sigma}_n^2(m)} \sum_{i=1}^2 \gamma_i(\sigma_D)^2 \left| \mathbf{u}_i(\sigma_D)^H \mathbf{y}^{(m)} \right|^2, \quad (2.8)$$

où:

$$\psi(\sigma_D) = \left[ 2 + \hat{\rho}^{(m)} \lambda_1(\sigma_D) \right] \left[ 2 + \hat{\rho}^{(m)} \lambda_2(\sigma_D) \right], \quad (2.9)$$

$$\gamma_i(\sigma_D) = \sqrt{\frac{\hat{\rho}^{(m)} \lambda_i(\sigma_D)}{2 + \hat{\rho}^{(m)} \lambda_i(\sigma_D)}}, \quad i = 1, 2. \quad (2.10)$$

$\hat{\rho}^{(m)}$  et  $\hat{\sigma}_n^{2(m)}$  sont respectivement les estimées du SNR instantané et de la puissance du bruit sur la  $m^{\text{ème}}$  fenêtre d'approximation. Nous notons avec  $(.)^H$  la conjuguée hermitienne (c'est-à-dire la conjugué complexe du transposé) d'un vecteur donné.

$\lambda_1$  et  $\lambda_2$  sont les deux valeurs propres de la matrice de covariance présentée en (2.5), dont leurs expressions sont données par:

$$\lambda_1 = N + \left| \frac{\sin(N\sigma_D T_s)}{\sin(\sigma_D T_s)} \right| \quad \text{et} \quad \lambda_2 = N - \left| \frac{\sin(N\sigma_D T_s)}{\sin(\sigma_D T_s)} \right|. \quad (2.11)$$

Leurs vecteurs propres associés,  $\mathbf{u}_1$  et  $\mathbf{u}_2$ , s'écrivent comme suit:

$$\mathbf{u}_1(\sigma_D) = \frac{1}{\sqrt{2\lambda_1(\sigma_D)}} \left( \mathbf{a}(-\sigma_D) + \frac{\varphi(2\sigma_D T_s)^*}{|\varphi(2\sigma_D T_s)|} \mathbf{a}(\sigma_D) \right), \quad (2.12)$$

$$\mathbf{u}_2(\sigma_D) = \frac{1}{\sqrt{2\lambda_2(\sigma_D)}} \left( \mathbf{a}(-\sigma_D) - \frac{\varphi(2\sigma_D T_s)^*}{|\varphi(2\sigma_D T_s)|} \mathbf{a}(\sigma_D) \right), \quad (2.13)$$

où

$$\varphi(x) = \frac{\sin(\frac{Nx}{2})}{\sin(\frac{x}{2})} e^{j\frac{(N-1)x}{2}}. \quad (2.14)$$

Finalement, l'estimé local de l'étalement de Doppler,  $\hat{\sigma}_D^{(m)}$  est obtenue comme suit:

$$\hat{\sigma}_D^{(m)} = \underset{\sigma_D}{\operatorname{argmax}} L(\sigma_D). \quad (2.15)$$

Afin d'améliorer la précision de l'estimation de l'étalement de Doppler, nous calculons, à la fin, la moyenne des estimés obtenus à chaque fenêtre d'approximation locale:

$$\hat{\sigma}_D = \frac{1}{M} \sum_{m=1}^M \hat{\sigma}_D^{(m)}. \quad (2.16)$$

### 2.4.3 Présentation de l'estimateur du rapport signal-à-bruit

Une caractéristique importante et attirante de l'estimateur du SNR développé en [57] est qu'au lieu d'évaluer l'expression de la LLF sur plusieurs candidats à deux dimensions,  $\boldsymbol{\theta} = [\rho, \sigma_n^2]$ , il utilise des formes analytiques simplifiées de ses estimées. Ceci lui permet d'éviter tout calcul itératif et la recherche des maxima dans un domaine à deux dimensions.

Tout d'abord, en considérant le modèle du système de (2.4), l'estimateur du SNR trouve les coefficients polynomiaux optimaux à partir de la  $m^{\text{ème}}$  séquence d'observation locale  $\mathbf{y}^{(m)}$  comme suit:

$$\hat{\mathbf{c}}^{(m)} = \left( \mathbf{B}^H \mathbf{B} \right)^{-1} \mathbf{B}^H \mathbf{y}^{(m)}, \quad (2.17)$$

où  $\mathbf{B} = \mathbf{A} \mathbf{T}$  est une matrice diagonale par blocs de dimensions  $(N \times L)$ .  $L$  représente l'ordre de l'approximation des coefficients polynomiaux du canal.  $\mathbf{A} = \text{diag}\{x(T_s), x(2T_s), \dots, x(NT_s)\}$  est une matrice diagonale qui contient les symboles pilotes de la  $m^{\text{ème}}$  fenêtre d'approximation locale, et  $\mathbf{T}$  est une matrice de *Vandermonde* dont les entrées correspondent aux instants d'échantillonnage,  $\{0, T_s, 2T_s, \dots, (N-1)T_s\}$ . Ceci permet d'écrire:

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & T_s & T_s^2 & \dots & T_s^{L-1} \\ 1 & 2T_s & (2T_s)^2 & \dots & (2T_s)^{L-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & (N-1)T_s & ((N-1)T_s)^2 & \dots & ((N-1)T_s)^{L-1} \end{bmatrix} \quad (2.18)$$

Ensuite, les estimées des coefficients polynomiaux locaux sont utilisées pour obtenir la puissance du bruit comme suit:

$$\hat{\sigma}_n^{2(m)} = \frac{1}{2N} \left\| \mathbf{y}^{(m)} - \mathbf{B}\hat{\mathbf{c}}^{(m)} \right\|^2. \quad (2.19)$$

Finalement, la  $m^{\text{ème}}$  estimée du SNR instantané est donnée par:

$$\hat{\rho}^{(m)} = \frac{\left\| \hat{\mathbf{h}}^{(m)} \right\|^2}{2N\hat{\sigma}_n^{2(m)}} \quad (2.20)$$

où,  $\hat{\mathbf{h}}^{(m)}$  représente l'estimée du canal à la  $m^{\text{ème}}$  fenêtre d'approximation. Grâce à la modélisation du canal développée dans [59], cet estimé est obtenu comme suit:

$$\hat{\mathbf{h}}^{(m)} = \mathbf{T} \hat{\mathbf{c}}^{(m)}. \quad (2.21)$$

Il est important de noter que cet estimateur permet aussi d'identifier les coefficients du canal aux positions pilotes. Il sera intégré dans l'architecture du CTR comme étant un estimateur de canal basé sur l'approche ML assistée par pilotes (DA ML).

## 2.5 Conclusion

Dans ce chapitre, nous avons présenté l'architecture interne du nouveau CTR. Ensuite, nous avons développé les formulations mathématiques de son module sensoriel que nous allons concevoir et implémenter. Dans le chapitre suivant, nous présenterons les compromis et les optimisations pris en considération lors de la conception de ce module ainsi que son architecture matérielle proposée.





## Chapitre 3

# Architecture matérielle proposée pour le module sensoriel du CTR

### 3.1 Introduction

Nous avons introduit précédemment les méthodologies et les outils de conception matérielle d'un circuit numérique et son implémentation sur des plateformes à base de FPGA. Ces outils permettent en quelque sorte de faciliter le transfert technologique de l'algorithme à l'architecture. Cependant, les concepteurs se trouvent généralement face à plusieurs contraintes de performance selon le type d'application.

Dans ce chapitre, nous présentons les contraintes de conception des systèmes de communication de future génération 5G et les stratégies d'optimisations à considérer pour réaliser le meilleur compromis entre elles. En suite, nous décrivons l'architecture matérielle proposée pour le module sensoriel du CTR en détaillant les structures internes de chaque sous-module.

## 3.2 Conception matérielle: compromis et optimisations

### 3.2.1 Contraintes de conception d'un circuit numérique

La conception d'un circuit numérique est considérée efficace et optimale si elle réussit à réaliser le meilleur compromis entre la latence de calcul, l'utilisation des ressources matérielles, la précision des résultats obtenus et l'énergie consommée au cours de l'exécution. Ces quatre paramètres constituent le défi le plus important face aux concepteurs des circuits numériques, particulièrement pour des applications de communication sans-fil à haute-vitesse. En effet, les systèmes de communication de la cinquième génération doivent assurer une latence de calcul de moins qu'une milliseconde afin de supporter des applications à temps réel telles que la communication véhicule-à-voiture (V2V) et les applications de la réalité augmentée/virtuelle (AR/VR). Par ailleurs, la nouvelle génération des systèmes de communication radio-mobiles est présentée comme une solution omniprésente qui couvre des millions d'applications et des centaines de milliards d'objets connectés. Pour atteindre cet objectif, il est primordial de minimiser le coût de fabrication et par conséquent le coût de commercialisation de ces objets par la réduction de la surface des puces embarquées, c'est-à-dire la réduction des ressources nécessaires pour faire fonctionner le circuit conçu. D'autre part, l'implémentation des algorithmes d'apprentissage automatique et de la prise de décision des radio-cognitives, ainsi que pour couvrir des applications de localisation, nécessitent une représentation binaire précise des signaux numériques et des modules de traitement de signal. Un dernier défi des circuits numériques de communication sans-fil est de réduire la consommation d'énergie afin d'augmenter l'autonomie des batteries et minimiser l'émission du dioxyde de carbone ( $CO_2$ ). Cependant, ces quatre objectifs sont liés entre eux. Par exemple, il est presque impossible de réduire la latence sans augmenter la surface utilisée ou sans dégrader la précision des signaux, et inversement.

D'autre part, la synchronisation entre les différents modules dans l'architecture interne d'un circuit numérique synchrone est très importante pour assurer sa stabilité et sa robustesse. Pour ce faire, nous intégrons dans notre design plusieurs machines séquentielles algorithmiques (MSAs). Une MSA, aussi connue sous le nom d'une machine (ou automate) à états finis, est un circuit séquentiel qui ne peut être que dans un seul état parmi un nombre fini d'états définis par le concepteur. Généralement, une MSA peut être modélisée par deux approches: celle de Moore et celle de Mealy. Dans ce travail, nous adoptons la première approche (idem., celle de Moore) pour assurer la synchronisation entre les différents modules, puisqu'avec cette approche les sorties de la MSA ne dépendent

que de l'état actuel, et leurs valeurs changent seulement au coup d'horloge (généralement au front montant). Toutefois, les machines de Mealy calculent les valeurs des sorties dès que leurs entrées se changent à cause de l'existence d'une logique combinatoire directe entre elles. Par conséquent, cette approche peut entraîner plusieurs problèmes de synchronisation et engendrer des états erronés. En outre, grâce au framework Xilinx System Generator, nous pouvons construire des MSAs avec un script MATLAB, au lieu de les coder avec des langages RTL.

### 3.2.2 Stratégies d'optimisation

#### Réduction de la latence

La latence d'un circuit numérique est définie comme le temps nécessaire pour produire un résultat final et correct. L'optimisation de ce paramètre peut se faire avec la réduction des chemins combinatoires, l'ajout des registres de pipeline, la mise en œuvre des architectures parallèles, la représentation binaire minimale des signaux, et la réduction de l'utilisation des mémoires tampons dans le circuit. Ainsi, l'utilisation d'une fréquence d'opération élevée permet de réduire la latence de calcul, mais elle peut générer plusieurs problèmes de synchronisation et induire l'apparition des glitches et du phénomène de la métastabilité.

#### Réduction de la complexité matérielle

Afin de réduire les ressources matérielles utilisées, nous adopterons une stratégie d'optimisation qui consiste à maximiser la réutilisation des modules et à réduire la taille des ressources mémoires dans le circuit.

#### Amélioration de la précision de calcul

Théoriquement, l'évaluation de performance des algorithmes d'estimation est faite en termes de l'erreur entre l'estimée et la vraie valeur. C'est pourquoi, nous devons assurer une bonne précision dans la représentation des signaux et principalement dans l'évaluation des fonctions mathématiques complexes. Nous utilisons l'outil « MATLAB Fixed Point Tool » pour déterminer la meilleure représentation en virgule fixe des différents signaux dans le design du module sensoriel du CTR.

## Minimisation de la puissance consommée

Une des techniques les plus puissantes pour réduire la puissance consommée est de maximiser l'utilisation des blocs IP disponibles et déjà optimisés par les constructeurs des FPGAs. Une autre technique consiste à simplement suspendre ou mettre en veille les parties du FPGA quand elles ne sont pas utilisées pendant l'exécution.

### 3.3 Architecture globale

En adoptant l'approche de conception descendante décrite dans 1.3.1, la première étape consiste à définir les spécifications de notre système en respectant les contraintes présentées ci-haut. Ensuite, nous identifions les différents modules du premier niveau qui constituent ce système et nous décrivons les interactions entre eux. Finalement, nous détaillons la structure matérielle interne de chaque module en le composant à des sous-modules, jusqu'à obtenir des circuits simples et basiques.

Nous illustrons dans la figure 3.1 le diagramme en bloc de l'architecture globale pour le module sensoriel du CTR. Comme tout système numérique, l'architecture proposée se compose d'un chemin de données et d'une unité de contrôle. Le chemin de données consiste en l'ensemble des fonctions qui manipulent et traitent les données. Généralement, il se compose des circuits combinatoires et séquentiels, alors que l'unité de contrôle, également connue sous le nom de contrôleur, permet de surveiller et contrôler le mouvement de ces données dans le module du chemin de données.

Bien que le nouveau CTR est conçu pour une transmission des symboles OFDM, nous intégrons l'estimateur du SNR et de l'étalement de Doppler dans une chaîne de communication simple et basée sur une transmission des symboles modulés en QAM. Cette modification de la chaîne de base est valable et n'influence pas sur la performance de l'estimateur choisi. En effet, le modèle du système de (2.1) considère une transmission à travers un canal de Rayleigh à évanouissement plat ce qui est toujours le cas avec les systèmes à base d'OFDM. Par conséquent, lors de nos expériences, nous devons considérer seulement les scénarios où le canal est plat.

Comme son nom l'indique, le « Générateur des symboles QAM » permet de générer une séquence des composants en phase et ceux en quadrature du signal à transmettre selon la disposition illustrée dans la figure 2.4. Ensuite, ces symboles, représentés par un train d'impulsions, passent par le bloc

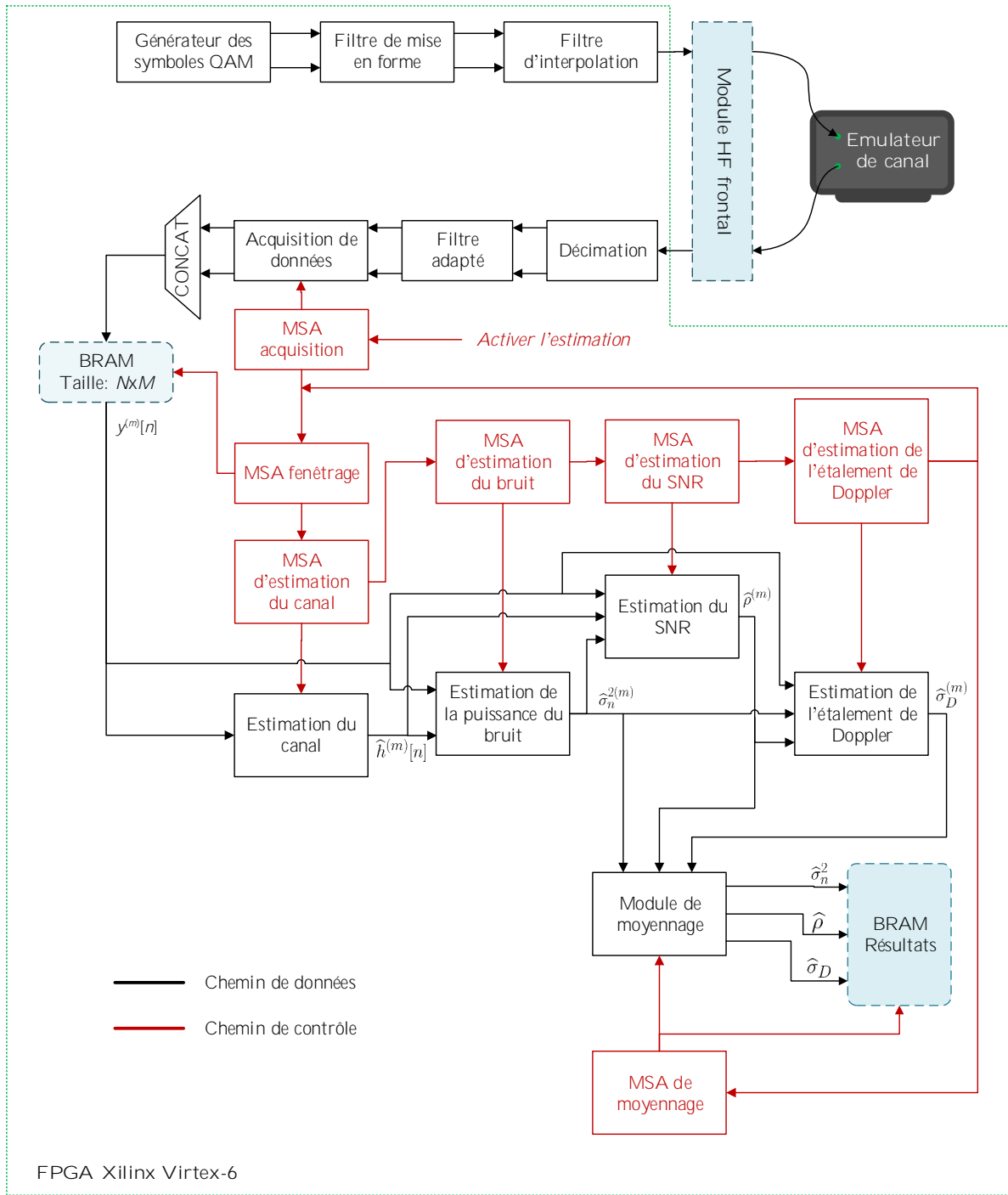


Figure 3.1 – Diagramme en bloc de l'architecture globale.

« Filtre de mise en forme » qui convertit cette séquence IQ en signal approprié pour le canal afin de minimiser l'interférence inter-symboles [60]. Nous considérons dans ce travail le bloc filtre en racine de cosinus surélevé (SRRC) proposé par la compagnie BEEcube dans [25]. Finalement, le

signal filtré passe par le module frontal d'émission de la plateforme miniBEE4, qui le convertit en signal analogique modulé en haute fréquence. Après le passage par le canal de transmission, ce signal traverse le module frontal de réception, responsable de sa conversion analogique/numérique et sa transformation en bande de base, avant qu'il soit filtré par le filtre adapté afin de maximiser le SNR au niveau du récepteur. Ensuite, le module « Acquisition de données » surveille l'extraction et l'enregistrement de  $(N \times M)$  symboles pilotes qui seront utilisés dans différents étages de calcul.

L'automate « MSA acquisition » a été conçu pour surveiller la tâche d'acquisition des symboles dans des mémoires vives partagées (BRAM) selon les paramètres  $M$ ,  $N$ , et  $T_s$ . Nous assurons toujours que ces paramètres soient définis par l'utilisateur sans avoir besoin de changer l'architecture implémentée afin de couvrir le maximum d'applications possibles. À la fin de l'acquisition de  $(N \times M)$  symboles pilotes, la « MSA acquisition » active la « MSA fenêtrage ». Cette dernière surveille et contrôle la tâche de l'estimation sur  $M$  fenêtres d'approximation locale. Son diagramme d'états est présenté dans la figure 3.2.

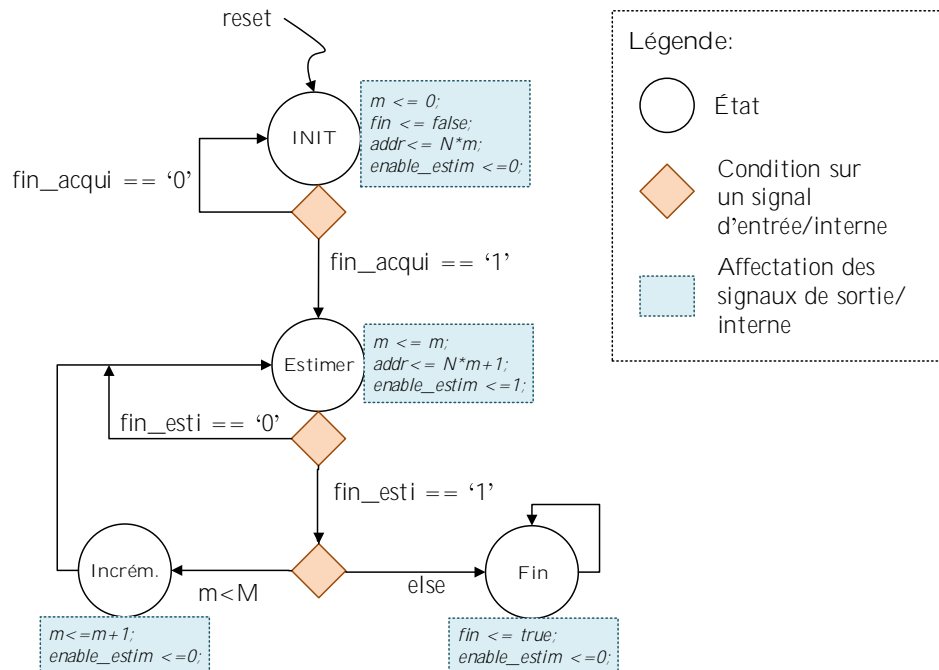


Figure 3.2 – Diagramme d'états de la « MSA fenêtrage ».

Durant l'exécution, la « MSA fenêtrage » peut être seulement dans l'un de ses quatre états (*INIT*, *Estimer*, *Incrém.*, et *Finir*). Elle prend comme entrées deux signaux d'état provenant de la « MSA acquisition » et de la « MSA d'estimation de l'étalement de Doppler » qui indiquent,

respectivement, la fin de la tâche d'acquisition de  $(N \times M)$  symboles et la tâche d'estimation locale du SNR et de l'étalement Doppler. En outre, elle produit à la sortie des signaux de contrôle qui permettent d'activer la lecture du flux de données de la  $m^{\text{ème}}$  fenêtre d'acquisition à l'adresse appropriée de la BRAM, déclencher les modules d'estimation, et réinitialiser tous les registres et les signaux intermédiaires à la fin de chaque estimation locale.

Dans ce qui suit, nous détaillons la structure interne de l'estimateur à maximum de vraisemblance du SNR et de l'étalement de Doppler.

### 3.4 Architecture proposée pour l'estimateur ML du SNR

Comme précédemment décrit dans sa formulation mathématique, l'algorithme de l'estimateur du SNR instantané s'exécute en quatre étapes:

1. Estimer les  $m^{\text{ème}}$  coefficients polynomiaux optimaux  $\hat{\mathbf{c}}^{(m)}$  en évaluant (2.17);
2. Estimer les  $m^{\text{ème}}$  coefficients du canal  $\hat{\mathbf{h}}^{(m)}$  en évaluant (2.21);
3. Estimer la puissance du bruit  $\hat{\sigma}_n^{2(m)}$  sur la  $m^{\text{ème}}$  fenêtre d'approximation en utilisant (2.19);
4. Estimer le  $m^{\text{ème}}$  SNR local instantané,  $\hat{\rho}^{(m)}$  à partir de (2.20).

En combinant les deux premières étapes dans un seul module, l'architecture proposée pour l'estimateur du SNR instantané est constituée de trois modules principaux: « Estimation du canal », « Estimation de la puissance du bruit », « Estimation du SNR ».

#### 3.4.1 Structure interne du module « Estimation du canal »

Comme son nom l'indique, ce module est responsable à l'estimation des  $N$  coefficients du  $m^{\text{ème}}$  canal. Son architecture interne est représentée à la figure 3.3. En effet, il prend comme entrée le flux des données des symboles reçus,  $\mathbf{y}^{(m)}$ , afin de produire dans sa sortie un flux des  $N$  coefficients du  $m^{\text{ème}}$  canal estimé, noté par  $\hat{\mathbf{h}}^{(m)}$ .

En supposant que la période symbole ( $T_s$ ), la taille de la fenêtre d'approximation ( $N$ ), et l'ordre du polynôme d'approximation des coefficients du canal ( $L$ ) sont déjà spécifiés par l'utilisateur et

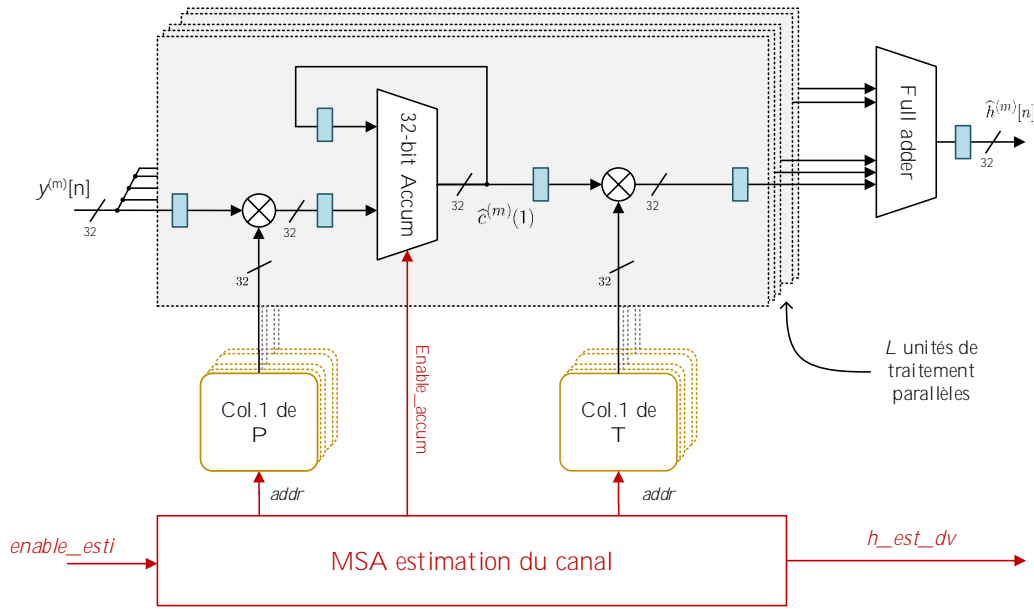


Figure 3.3 – Architecture interne du module « Estimation du canal ».

ne peuvent pas être modifiés pendant l'expérience, les  $N \times L$  éléments des matrices  $\mathbf{B}$  et  $\mathbf{T}$  sont tous constants. C'est pourquoi, et sans dégradation de performance, nous proposons de les générer « offline » à l'aide des scripts MATLAB. Par conséquent, nous pouvons éviter l'implémentation des produits et des inverses matriciels, tels que montrés dans l'expression  $(\mathbf{B}^H \mathbf{B})^{-1} \mathbf{B}^H$  de l'équation (2.17), en les calculant avec MATLAB puis en stockant leurs  $N \times L$  éléments dans des différentes unités de mémoires. Grâce à cette approche, nous réduisons le nombre de modules de multiplication et d'accumulation (MACs) dans notre architecture par un facteur de  $N(N-1)(L-1)$ , et nous évitons aussi la conception et l'implémentation des algorithmes lourds pour inverser des matrices de taille  $(L \times L)$ .

D'autre part, nous subdivisons l'architecture du module « Estimation du canal » en  $L$  unités de traitement parallèles et indépendantes. Chaque unité sert à calculer une portion du résultat final. Cette architecture est connue sous le nom d'une « architecture systolique ».

Dans la  $i^{\text{ème}}$  unité de traitement,  $i = 1, 2, \dots, L$ , le flux des symboles reçus ( $\mathbf{y}^{(m)}$ ) de la  $m^{\text{ème}}$  fenêtre locale d'approximation est multiplié et accumulé, dans une première étape, par les  $N$  éléments de la  $i^{\text{ème}}$  colonne de la matrice  $\mathbf{P} = (\mathbf{B}^H \mathbf{B})^{-1} \mathbf{B}^H$ , pour donner à sa sortie le  $i^{\text{ème}}$  coefficient polynomial  $\hat{c}^{(m)}(i)$ . Une fois que la sortie de l'accumulateur soit valide, la seconde multiplication



$\hat{c}^{(m)}(i) \times [\mathbf{T}]_i$  est déclenchée. Finalement, les sorties de toutes les unités de traitement passent par un additionneur afin d'obtenir un flux de  $N$  coefficients du  $m^{\text{ème}}$  canal estimé  $\hat{\mathbf{h}}^{(m)}$ .

Le traitement exécuté dans ce module est contrôlé par la « MSA estimation du canal ». Elle assure la lecture des coefficients des matrices  $\mathbf{P}$  et  $\mathbf{T}$ , l'activation/désactivation de l'accumulateur, et la synchronisation entre les différentes unités de traitement. Finalement, elle indique la fin de la tâche d'estimation du canal grâce au signal `h_est_dv` qui permet aussi de déclencher le module « Estimation de la puissance du bruit ».

### 3.4.2 Structure interne du module « Estimation de la puissance du bruit »

En considérant une transmission avec des symboles pilotes unitaires (idem.  $x[n] = 1$ ), la matrice  $\mathbf{A} = \text{diag}\{x(T_s), x(2T_s), \dots, x(N_T s)\}$  devient simplement égale à l'identité. Par conséquent, nous pouvons déduire que  $\mathbf{B} = \mathbf{AT} = \mathbf{T}$ . En combinant les équations (2.19) et (2.21), cette déduction mathématique permet de réécrire l'estimée de la puissance du bruit comme suit:

$$\hat{\sigma}_n^{2(m)} = \frac{1}{2N} \left\| \mathbf{y}^{(m)} - \hat{\mathbf{h}}^{(m)} \right\|^2. \quad (3.1)$$

Le module « Estimation de la puissance du bruit » exécute l'équation (3.1) afin de fournir une estimée de la puissance du bruit pendant la  $m^{\text{ème}}$  fenêtre d'approximation. La structure interne de ce module est illustrée dans la figure 3.4. Principalement, il est constitué d'un soustracteur, un sous-module « norm\_2 », et un multiplieur.

Le sous-module « norm\_2 » permet de calculer la carrée de la norme euclidienne d'un vecteur complexe de dimension  $K$ , comme suit:

$$\|\mathbf{u}\|^2 = \sum_{k=1}^K |u_k|^2 = \sum_{k=1}^K \left[ \mathcal{R}(u_k)^2 + \mathcal{I}(u_k)^2 \right], \quad (3.2)$$

où  $u_k$  représente le  $k^{\text{ème}}$  élément complexe du vecteur  $\mathbf{u}$ . Ainsi, ce sous-module sera utilisé plusieurs fois dans notre architecture.

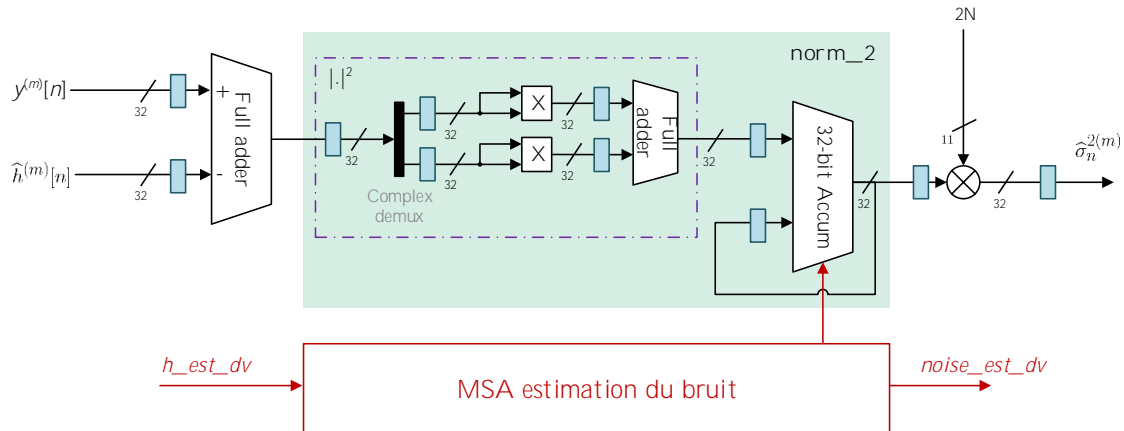


Figure 3.4 – Design interne du module « Estimation de la puissance du bruit ».

### 3.4.3 Structure interne du module « Estimation du SNR »

La dernière étape pour estimer le SNR instantané est d'évaluer l'équation (2.20). Le diagramme en bloc de la structure matérielle interne du module « Estimation du SNR », responsable à l'exécution de cette équation, est représenté dans la figure 3.5. Les signaux d'entrée de ce module sont le flux de  $N$  coefficients de la  $m^{\text{ème}}$  estimée du canal ( $\hat{\mathbf{h}}^{(m)}$ ) et la puissance du bruit ( $\hat{\sigma}_n^{2(m)}$ ) estimée sur la même fenêtre d'approximation.

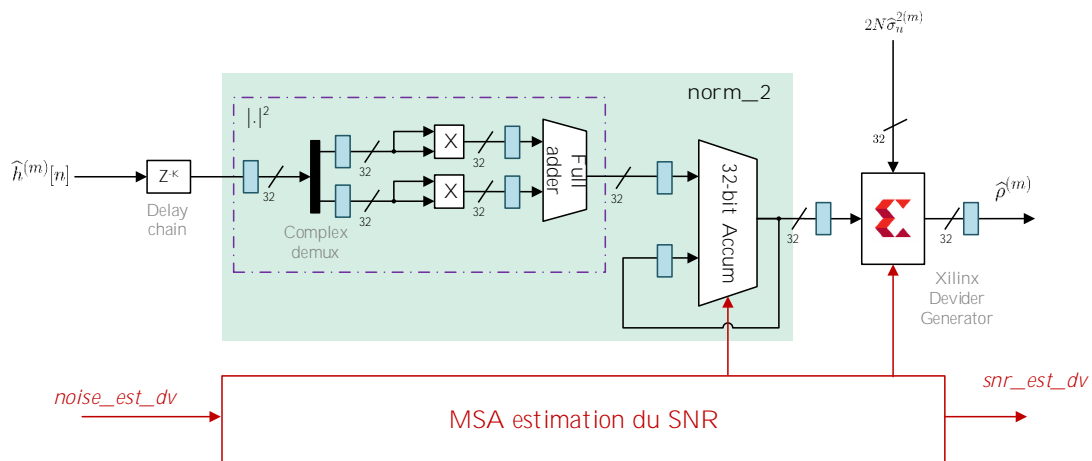


Figure 3.5 – Design interne du module « Estimation de la puissance du bruit ».

Dans une première étape, les coefficients du canal estimé sont retardés dans le temps pour synchroniser avec le résultat de l'estimation de la puissance du bruit et éviter les valeurs intermédiaires

des estimées du SNR à la sortie de ce module. La valeur de ce retard dépend principalement de la taille de l'observation locale ( $N$ ). Ensuite, nous réutilisons le sous-module « norm\_2 » pour calculer  $\|\hat{\mathbf{h}}^{(m)}\|^2$ . Dès que cette dernière est valide, la « MSA estimation du SNR » active la division par le signal  $2N\hat{\sigma}_n^{2(m)}$ . La division numérique est généralement très coûteuse en termes de latence et consommation de ressources dans le FPGA. Pour assurer une performance maximale, nous utilisons le bloc « *Xilinx Divider Generator* » de Xilinx qui fournit, en se basant sur l'approximation Radix-2, une utilisation efficace de ressources matérielles et une performance de calcul très importante. Ce bloc assure aussi la possibilité de modifier le degré du parallélisme et le niveau de pipeline, et choisir la latence appropriée dans le but de trouver le meilleur compromis entre les quatre paramètres présentés au début de ce chapitre.

### 3.5 Architecture proposée pour l'estimateur ML de l'étalement de Doppler

Dans cette section, nous présentons l'architecture proposée pour l'estimateur ML de l'étalement de Doppler. D'un point de vue système, le module « Estimation de l'étalement de Doppler » prend comme entrées les estimées du SNR instantané et de la puissance du bruit, ainsi que la séquence d'observation de la  $m^{\text{ème}}$  fenêtre d'approximation locale.

Contrairement à l'estimateur à maximum de vraisemblance du SNR, nous n'avons pas une expression analytique des estimées ML de l'étalement de Doppler. Par conséquent, ce module doit évaluer l'expression de la fonction coût (LLF) de (2.8) sur différentes valeurs candidates de l'étalement de Doppler, initialement inconnu. Ensuite, il cherche son maximum global qui correspond à l'estimée de ce paramètre sur la  $m^{\text{ème}}$  fenêtre d'approximation locale. Il est évident que cette approche itérative se traduit par une complexité matérielle et une latence de calcul relativement élevées. C'est pour cela, en adoptant l'approche de conception « Top-down », nous simplifions, tout d'abord, la formulation mathématique de l'estimateur de l'étalement de Doppler dans le but de concevoir, d'un point de vue implémentation matérielle, un modèle mathématique qui réduit toute fonction coûteuse en un ensemble d'opérations basiques. Après quelques manipulations algébriques,

l'équation (2.8) devient:

$$\begin{aligned}
L(\sigma_D) = & -\text{Ln} \left[ 2 + \hat{\rho}^{(m)} \lambda_1(\sigma_D) \right] - \text{Ln} \left[ 2 + \hat{\rho}^{(m)} \lambda_2(\sigma_D) \right] \\
& + \frac{\hat{\rho}^{(m)}}{2 + \hat{\rho}^{(m)} \lambda_1(\sigma_D)} \left| \left( \mathbf{a}(-\sigma_D) + \frac{\varphi^*(2\sigma_D T_s)}{|\varphi(2\sigma_D T_s)|} \mathbf{a}(\sigma_D) \right)^H \mathbf{y}^{(m)} \right|^2 \\
& + \frac{\hat{\rho}^{(m)}}{2 + \hat{\rho}^{(m)} \lambda_2(\sigma_D)} \left| \left( \mathbf{a}(-\sigma_D) - \frac{\varphi^*(2\sigma_D T_s)}{|\varphi(2\sigma_D T_s)|} \mathbf{a}(\sigma_D) \right)^H \mathbf{y}^{(m)} \right|^2
\end{aligned} \tag{3.3}$$

D'autre part, nous élaborons une planification temporelle des différentes opérations mathématiques afin d'identifier les prédécesseurs, les successeurs, et les dépendances entre elles. Cette étape permet de construire des MSAs robustes et de faciliter la détection des chemins critiques (idem., les opérations les plus coûteuses en termes de latence), et en conséquence, les accélérer en implémentant une ou plusieurs solutions d'optimisation présentées au début de ce chapitre.

Le diagramme en bloc de la structure interne du module « Estimation de l'étalement de Doppler » est représenté dans la figure 3.6.

Étant contrôlé par la « MSA d'estimation de l'étalement de Doppler », le chemin de données de ce module est composé de deux sous-modules: « LLF » et « Argmax ».

### 3.5.1 Conception et optimisation de l'architecture de LLF

Le sous-module « LLF » permet d'évaluer l'équation (3.3) dans des différentes valeurs candidates de l'étalement de Doppler. Pour chaque valeur de  $\sigma_D$ , un ensemble d'opérations est exécuté pour donner la valeur de  $L(\sigma_D)$ . Grâce à la planification temporelle de ces opérations, nous pouvons identifier les processus parallèles et maximiser la réutilisation des blocs afin de réduire la consommation des ressources et la latence computationnelle. D'autre part, l'expression de la LLF présente plusieurs fonctions mathématiques complexes et lourdes telles que les fonctions trigonométriques, la fonction logarithmique, la racine carrée, etc. Nous présentons dans ce qui suit, les différentes optimisations introduites dans ce module dans le but de réaliser le meilleur compromis entre la latence, la précision, l'utilisation des ressources, et la consommation d'énergie.

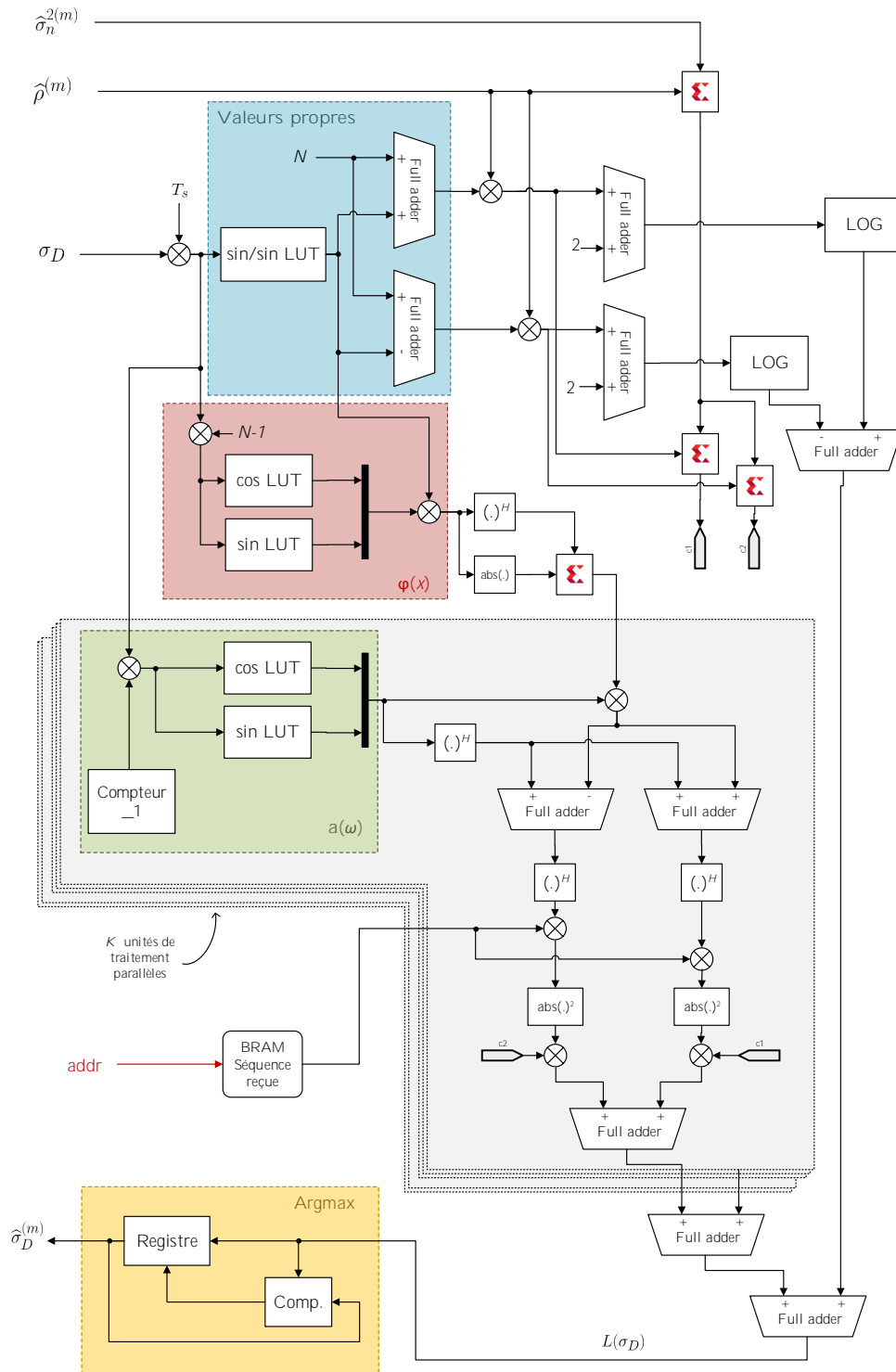


Figure 3.6 – Diagramme en bloc de la structure interne du module « Estimation de l'étalement de Doppler ».

## Optimisation de la fonction logarithme

Plusieurs méthodes et techniques de calcul du logarithme d'un nombre réel ont été proposées. En effet, on trouve des techniques qui se basent sur des approximations linéaires [61] ou sur des algorithmes rotationnels comme le CORDIC, et d'autres sur le principe de la décomposition itérative. Cependant, toutes ces techniques demandent des ressources très importantes et un temps de calcul relativement élevé pour produire une valeur approchée du logarithme d'un nombre réel. Dans ce travail, nous proposons une nouvelle architecture efficace et robuste pour le calcul du logarithme à n'importe quelle base.

Soit  $x$  un signal d'entrée représenté en virgule fixe  $\{\text{fix}(n_{\text{bit}}, b_p)\}$ , où  $n_{\text{bit}}$  et  $b_p$  représentent, respectivement, la longueur du mot binaire et la position de la virgule (idem., nombre de bits réservés pour présenter la partie fractionnelle du signal  $x$ ). Ce signal peut être représenté comme suit:

$$x = m2^e, \quad (3.4)$$

où  $m$  et  $e$  désignent, respectivement, la mantisse et l'exposant. Il est important ici de noter que  $m$  est un nombre réel signé appartenant à l'intervalle  $[1, 2[$  et  $e$  est un nombre entier signé. Par conséquent, le logarithme en base 2 du signal  $x$  est calculé comme suit:

$$\text{Log}_2(x) = \text{Log}_2(m) + e. \quad (3.5)$$

Ainsi, le calcul du logarithme de base 2 du signal  $x$  (qui est dans un intervalle inconnu) se réduit à chercher les valeurs de l'exposant  $e$  et de la mantisse  $m$ . Nous remarquons que  $e$  et  $\text{Log}_2(m)$  représentent, respectivement, la partie entière et la partie fractionnelle du résultat. Le calcul de ces deux parties est réalisé comme suit:

1. La valeur de l'exposant  $e$  correspond exactement à la position du bit le plus significatif (MSB) du signal  $x$  par rapport à la position de la virgule, obtenue avec une suite des comparaisons;
2.  $m$  est le résultat du déplacement de la virgule fixe à la position correspondante à  $e$  grâce à un registre à décalage adressable;

3.  $\text{Log}_2(m)$  est calculée avec un tableau de correspondance (LUT).

La figure 3.7 illustre la technique d'extraction de la mantisse et l'exposant d'un signal réel.

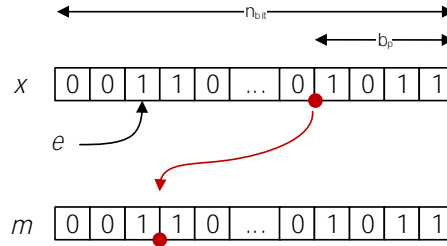


Figure 3.7 – Exemple illustratif de la technique de recherche de l'exposant et la mantisse d'un signal représenté en virgule fixe.

Nous illustrons dans la figure 3.8 l'architecture proposée pour le calcul optimal du logarithme d'un signal  $x$ , dans la base 2. Cette architecture est généralisée pour n'importe quelle base en multipliant le résultat par un facteur de  $1/\text{Log}_2(a)$ , grâce à la transformation suivante:

$$\text{Log}_a(x) = \frac{\text{Log}_2(x)}{\text{Log}_2(a)}. \quad (3.6)$$

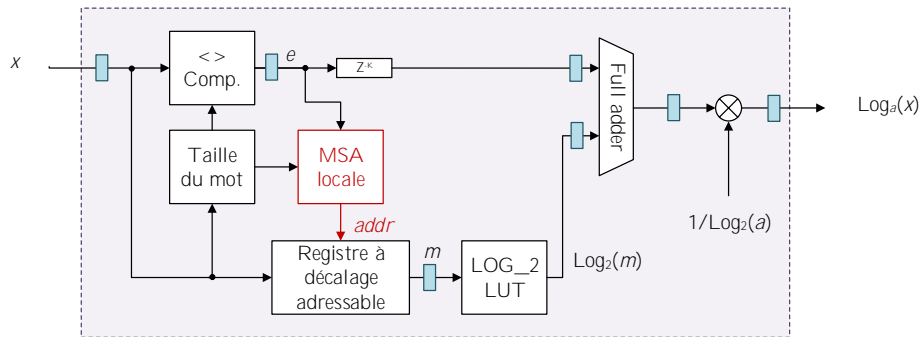


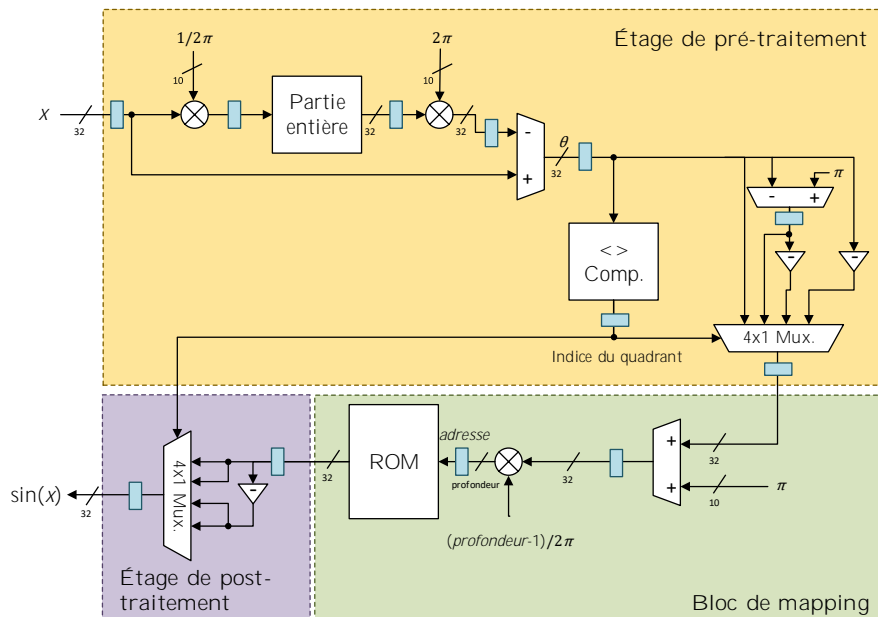
Figure 3.8 – Architecture proposée pour le calcul du logarithme d'un nombre réel.

### Optimisation des fonctions trigonométriques

Les algorithmes de traitement numérique du signal intègrent généralement un ensemble des fonctions trigonométriques telles que le sinus et le cosinus. L'utilisation des tables de correspondance

est une solution simple, facile, et rapide pour calculer ces fonctions. En effet, le fait de déduire une valeur précalculée et stockée dans une mémoire donnée est toujours plus facile et plus rapide que de la calculer en utilisant des algorithmes rotationnels ou itératifs comme l'algorithme de CORDIC. Cependant, la conception d'un LUT efficace nécessite une connaissance à priori de la nature du signal d'entrée et sa variation, ce qui n'est pas toujours le cas dans plusieurs applications. En outre, il existe un compromis entre la taille du LUT, c'est-à-dire la ressource nécessaire pour stocker la table, et la précision de la valeur de sortie. Pour faire face à ce problème, nous proposons une nouvelle technique qui permet de réduire la taille des LUTs utilisés pour des les fonctions trigonométriques sans aucune dégradation au niveau de la précision.

La solution proposée consiste à manipuler le signal sur différents étages de traitement en exploitant certaines identités mathématiques dans le but de réduire l'intervalle d'entrée. Un exemple de l'architecture matérielle de la fonction sinus est illustré dans la figure 3.9. Elle se compose d'un étage de pré-traitement, d'un bloc de mapping, et un étage de post-traitement.



**Figure 3.9 – Exemple d'architecture interne optimisée pour le calcul de la fonction sinus.**

L'étage de pré-traitement permet de traduire, dans une première étape, le signal d'entrée, qu'on note par  $x$ , dans l'intervalle  $[-\pi, \pi]$ . Autrement dit, cet étage permet de calculer la valeur principale,  $\theta$ , de ce signal. Ensuite, en exploitant les propriétés de symétrie des fonctions trigonométriques, nous



pouvons coder seulement un seul quadrant  $\left[0, \frac{\pi}{4}\right]$ . L'intégration de cet étage permet de réduire la taille de la LUT par un facteur de 4, et donc de réduire la profondeur de la mémoire à utiliser pour stocker les valeurs précalculées.

Le bloc de mapping calcule l'adresse mémoire correspondance à la valeur stockée à partir du signal  $\theta$ , en utilisant la bijection suivante:

$$adresse = \left\lfloor (\theta - min) \times \frac{profondeur - 1}{max - min} \right\rfloor, \quad (3.7)$$

où  $\lfloor \cdot \rfloor$  désigne la partie entière d'un nombre réel. *profondeur* est la taille de la LUT, et *max* et *min* représentent les bords de l'intervalle d'entrée à la LUT (idem.,  $\left[0, \frac{\pi}{4}\right]$ ).

Finalement, l'étage de post-traitement est utilisé pour réajuster le signal de sortie à son quadrant initial.

### Optimisation de la multiplication vectorielle

La multiplication vectorielle des vecteurs propres  $\mathbf{u}_1^H$  et  $\mathbf{u}_2^H$  par la séquence reçue sur la  $m^{\text{ème}}$  fenêtre d'approximation locale est relativement coûteuse en termes de latence computationnelle. Pour réduire cette latence, nous proposons une architecture systolique basée sur le principe de la décomposition du processus de la multiplication vecteur-par-vecteur en plusieurs unités de traitement parallèles et indépendantes. Chaque unité est responsable de calculer une portion du résultat final. Un exemple d'une décomposition systolique d'un ordre  $K$ , est donné par:

$$\mathbf{u} \cdot \mathbf{v} = \sum_{n=1}^N u[n]v[n] = \sum_{k=0}^{K-1} \sum_{n=1}^{N/K} u \left[ n + k \frac{N}{K} \right] v \left[ n + k \frac{N}{K} \right]. \quad (3.8)$$

Avec cette approche et en assurant une synchronisation parfaite entre les unités de traitement, il est possible de réduire la latence par un facteur de  $N/K$ .

### 3.5.2 Structure interne du sous-module Argmax

Ce bloc, illustré dans la figure 3.6, est responsable de déterminer l'argmax de la fonction LLF, en cherchant la candidate  $\sigma_D$  à laquelle la fonction LLF atteint son maximum. Une comparaison "online" permet d'éviter l'enregistrement des toutes les valeurs obtenues de la LLF dans une BRAM. Cette méthode permet de réduire les ressources mémoires nécessaires pour faire fonctionner l'architecture proposée.

### 3.5.3 Optimisation de la technique de recherche sur le maximum de la LLF

Comme nous avons précédemment présenté, l'estimateur de l'étalement de Doppler doit évaluer la fonction LLF dans plusieurs valeurs candidates de  $\sigma_D$ . Bien évidemment, la latence du processus de recherche de la valeur maximale de la fonction LLF est directement liée au nombre de valeurs candidates auxquelles la LLF est évaluée. L'intervalle de ces valeurs varie selon le type d'application. En supposant une situation où un mobile roule avec une vitesse maximale  $v = 500 \text{ km/h}$  et transmet un signal modulé sur une fréquence porteuse  $f_c = 2.45 \text{ GHz}$ , la fréquence maximale de Doppler dans ce cas est  $f_D \simeq 1200 \text{ Hz}$ . Il est important de noter que la fréquence maximale de Doppler est liée à l'étalement de Doppler  $\sigma_D$  par une relation qui dépend de la nature du modèle spectral du canal. Dans ce travail, nous considérons le modèle de Jake's où la fréquence de Doppler est donnée par:

$$f_D = \frac{\sqrt{2}}{2\pi} \sigma_D. \quad (3.9)$$

Dans ce cas, il est nécessaire d'évaluer la fonction LLF sur un intervalle  $\sigma_D \in [0, 10000]$  afin de couvrir plusieurs applications et respecter l'exigence de la haute mobilité envisagée par les systèmes 5G. Ainsi, pour réduire la latence de la recherche de la valeur maximale sur cet intervalle, nous proposons une nouvelle technique basée sur le principe de la recherche « Coarse-To-Fine » (CTF).

Nous admettons que la fonction LLF présente un et un seul pic sur l'intervalle  $[0, 10000]$ , ce qui est toujours vrai selon les résultats obtenus dans [53]. Dans ce cas, la recherche du maximum se fait sur deux étapes. Premièrement, la fonction LLF est évaluée dans un ensemble des valeurs candidates incrémentées par un « pas grossier » dans le but de trouver un premier maximum global. Ensuite, une recherche « fine » autour de ce dernier est exécutée afin de trouver le maximum local

à partir duquel l'estimée  $\hat{\sigma}_D^{(m)}$  est obtenue. La tâche de recherche du maximum est contrôlée par la « MSA estimation de l'étalement de Doppler » dont le diagramme d'état est représenté dans la figure 3.10.

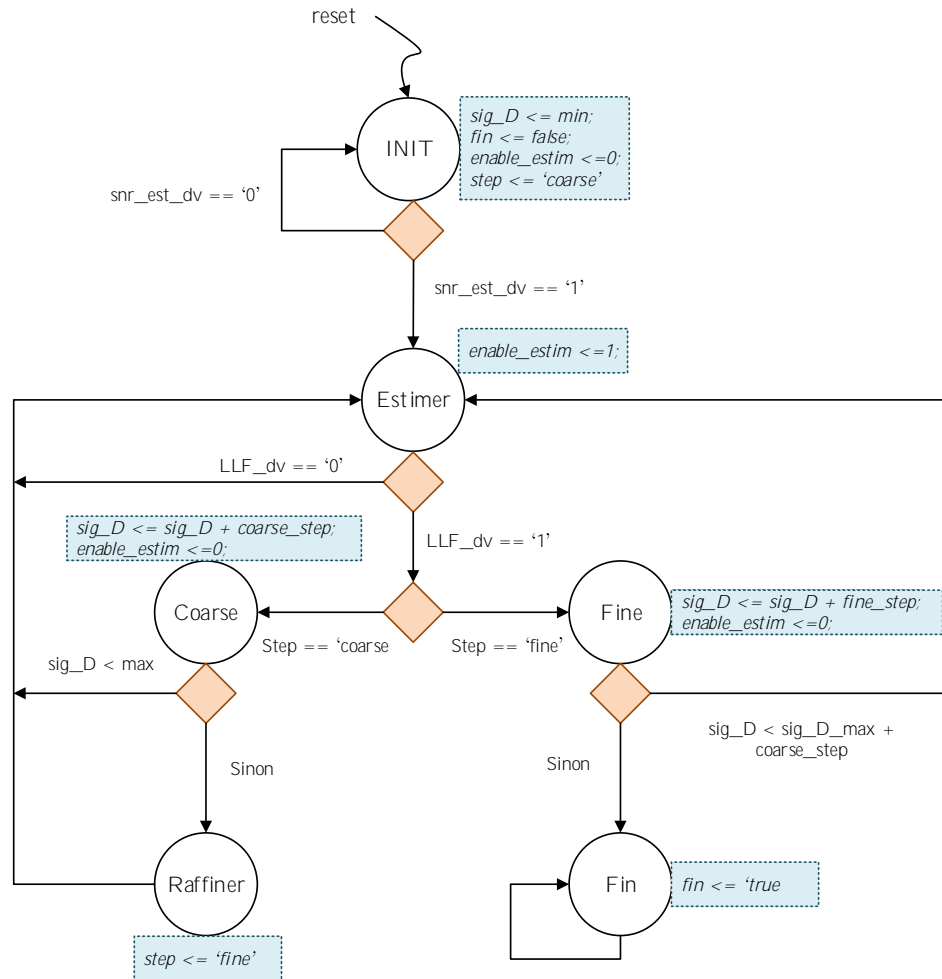


Figure 3.10 – Diagramme d'état de la « MSA estimation de l'étalement de Doppler ».

Nous assurons que les paramètres  $min$ ,  $max$ ,  $coarse\_step$ ,  $fine\_step$  sont définis par l'utilisateur afin de donner le maximum de flexibilité à l'architecture proposée. À la fin de l'estimation locale du SNR instantané et de l'étalement de Doppler, la « MSA estimation de l'étalement de Doppler » transmet un signal d'état à la « MSA fenêtrage » et à la « MSA moyennage ».

### **3.6 Conclusion**

Dans ce chapitre, nous avons détaillé la phase de conception du module sensoriel du nouveau CTR et les différentes stratégies d'optimisations effectuées pour réduire la complexité matérielle de l'architecture proposée. Dans le chapitre suivant, nous allons présenter les résultats expérimentaux de l'implémentation de l'architecture proposée dans la plateforme SDR.

# Chapitre 4

## Implémentation et évaluation de performances

### 4.1 Introduction

La dernière étape dans le processus de prototypage rapide des systèmes numériques est l'étape de test et de validation. Pour ce faire, l'architecture conçue est implémentée dans la plateforme cible et intégrée dans un banc d'essai logiciel ou matériel afin d'évaluer sa performance sous plusieurs scénarios de propagation. Cette évaluation est faite généralement selon le type d'application. Dans notre cas, nous évaluons l'architecture proposée sur quatre axes: complexité matérielle, efficacité énergiquement, latence computationnelle, et précision d'estimation.

Dans ce chapitre, nous allons procéder à l'étape de l'implémentation et de l'évaluation de performance du design proposé.

### 4.2 Implémentation du module sensoriel

L'architecture matérielle proposée pour le module sensoriel a été développée à l'aide de l'outil Xilinx System Generator. Cet outil offre aussi la possibilité de simuler le design dans un contexte « cycle/bit-accurate » grâce au transfert des données entre les blocs de Xilinx et l'environnement de

travail de MATLAB/Simulink. Par ailleurs, nous avons pu concevoir des bancs d'essai (en anglais, *testbenches*) afin de tester et valider séparément les différentes unités de traitement et vérifier les interactions entre elles lors de l'étape de l'intégration. Ces bancs d'essai sont composés des signaux de stimulation, de l'unité mise à l'essai (*unit under test*), et des outils d'affichage.

Les signaux de stimulation sont utilisés pour stimuler les entrées de l'unité mise à l'essai. Ils peuvent être des signaux de contrôle, tels qu'un signal d'activation ou de remise à zéro (*reset*), comme ils peuvent être des signaux de données générés « offline ». L'intégration de ces signaux est réalisée avec des blocs génériques de Simulink comme le bloc *From\_Workspace* et le bloc *step*. Ensuite, les signaux générés sont transférés à l'unité mise à l'essai grâce à l'interface *Gateway\_In*.

L'unité mise à l'essai peut être une MSA, un chemin de données, ou les deux combinés ensemble. Finalement, les outils d'affichage permettent de représenter les signaux de sortie de cette unité sur différents supports d'affichage. Ces derniers peuvent être des blocs de visualisation temporelle (*scope*) ou des variables exportées vers l'environnement MATLAB pour une éventuelle manipulation « offline ».

Après avoir effectué les tests unitaires et d'intégration en MATLAB Simulink, nous avons ajouté au design proposé un ensemble des registres personnalisés fournis par BPS et accessibles à partir du logiciel lors de l'exécution matérielle afin de modifier les paramètres de simulation sans avoir besoin de régénérer le bitstream. Pour assurer le maximum de flexibilité, nous avons intégré des registres pour contrôler le signal *System\_Enable* et *System\_Reset*, et modifier les valeurs de la taille de la fenêtre d'approximation locale ( $N$ ), le nombre des fenêtres d'approximation ( $M$ ), la période symbole ( $T_s$ ), les bords de l'intervalle de recherche de l'argument qui maximise la LLF (argmax) ( $[min, max]$ ), le pas grossier (*coarse\_step*), et le pas fin (*fine\_step*). D'autre part, nous avons ajouté aussi des mémoires « Shared BRAM » dans lesquelles nous sauvegardons les résultats de l'exécution matérielle du design afin de les rendre accessibles à partir du logiciel. Finalement, le design complet est synthétisé, mappé, placé, et routé avec l'outil BPS pour un FPGA cible de type « Virtex-6 XC6V5X475T FF1759-2 ».

Les paramètres de la co-simulation HIL sont présentés dans le tableau 4.1.

Tableau 4.1 – Configuration des paramètres de la co-simulation HIL <sup>1</sup>.

	Paramètre	Description	Valeur
<b>Configuration en bande de base</b>	$N$	Taille de la fenêtre locale	128
	$M$	Nombre des fenêtres locales	8
	$T_s$	Période symbole	10 $\mu$ sec
	$L$	Ordre du polynôme optimal	3
<b>Configuration FPGA/FMC111 front-end</b>	$f_c$	Fréquence porteuse	2.4576 GHz
	$f_{IF}$	Fréquence intermédiaire	32.720 MHz
	vga	Gain d'amplification variable	8 dB
	rx_atten	Atténuation au RX	8 dB
	DAC/ADC freq	Fréquence d'échantillonnage DAC/ADC	61.440 MHz
	Clk	Horloge du FPGA	61.440 MHz

### 4.3 Évaluation de la complexité matérielle

Nous évaluons la complexité matérielle de l'architecture proposée pour le module sensoriel du nouveau CTR en termes de la surface consommée (idem., ressources consommées du FPGA). Comme il n'y a pas des travaux antérieurs d'implémentation d'un tel module auxquels nous pouvons comparer la complexité matérielle du design proposé, nous avons décidé de concevoir et implémenter une autre architecture « non optimisée » et la considérer comme une référence pour évaluer la complexité matérielle de notre design proposé. En effet, cette architecture, appelée « Architecture I », n'intègre pas les différentes optimisations décrites dans le chapitre précédent, telles que la recherche CTF, les LUTs optimisés, etc. Nous comparons l'utilisation des ressources matérielle de la solution optimisée à l'architecture non optimisée et évaluons le gain fourni par les différentes optimisations que nous avons réalisées lors de la conception de l'ensemble du système.

1. Notez que tous les paramètres de la co-simulation présentés dans le tableau 4.1 sont contrôlables et peuvent être modifiés par l'utilisateur, grâce à l'interface NectarOs, sans avoir besoin de régénérer le fichier de programmation du FPGA. Cependant, il existe une contrainte de choix de la fréquence d'opération (idem., l'horloge du FPGA) et les fréquences du DAC et du ADC. En effet, selon [25], le choix de ces fréquences doit respecter la relation suivante:  $f_{DAC} = 2f_{ADC} = 2f_{FPGA}$ .

Grâce à BPS, nous générons un rapport de synthèse contenant un résumé sur les différentes ressources consommées du FPGA. Le tableau 4.2 représente un aperçu comparatif entre les deux solutions (idem., optimisée et non-optimisée). Comme il est montré dans le tableau 4.2, l'architecture optimisée consomme moins de 2% des registres disponibles sur le FPGA, tandis que l'« Architecture I » demande environ 10% de la même ressource. En outre, l'architecture optimisée utilise 9837 sur 297600 des LUTs, ce qui représente seulement 3.31% de tous les LUTs fournis par le FPGA Virtex-6. Étant donné qu'un bloc logique configurable (CLB) de la gamme Virtex-6 est constitué de quatre LUTs et huit registres, l'architecture proposée (idem., optimisée) exige seulement 1480/74400 CLBs, tandis que l'« Architecture I » nécessite 7380/74400 CLBs pour faire exécuter le même algorithme dans les mêmes conditions.

**Tableau 4.2 – Complexité matérielle à  $N = 128$ ,  $M = 8$ , et  $L = 3$ .**

	Éléments logiques		Ressources spécifiques	
	Regitres	LUTs	BRAM/FIFO	DSP48
<b>Disponible sur Xilinx Virtex-6</b>	595200	297600	1064	2016
<b>Architecture I</b>	58789 (9.88%)	36117 (12.14%)	563 (52.91%)	318 (15.77%)
<b>Proposée Architecture</b>	11854 (1.99%)	9837 (3.31%)	369 (34.68%)	355 (17.61%)
<b>Gain</b>	79.83%	72.76%	34.45%	-11.64%

Les ressources spécifiques, BRAM/FIFO et DSP48, sont des ressources disponibles seulement sur la FPGA Virtex-6 et sont les plus limitées pour faire fonctionner un design matériel. Comme le montre le tableau 4.2, l'architecture optimisée utilise respectivement 34.68% et 17.61% des BRAM/FIFO et DSP48 disponibles. En fait, un DSP48 est une structure matérielle composée d'un multiplieur, un additionneur, et un accumulateur. La taille de la mémoire BRAM est principalement de 36 Kbits, et celle-ci peut également être utilisée comme deux blocs indépendants de 18 Kbits. L'architecture optimisée présente un gain de 34.45% en termes des ressources mémoires utilisées. Ce gain est expliqué par le fait que cette architecture n'intègre pas des mémoires tampons (buffers). Bien qu'elle demande seulement 17.61% des ressources DSP48, l'architecture proposée montre une perte de 11.64% au niveau de cette ressource par rapport à celle non optimisée. Cette perte est principalement due à l'utilisation massive des structures MACs pour évaluer la multiplication vecteur-par-vecteur sur plusieurs unités de traitement parallèles.



En conclusion, les différentes optimisations effectuées montrent un gain important en termes de complexité matérielle par rapport à la solution non optimisée. Ainsi, les résultats du tableau 4.2 confirment l'efficacité de l'architecture proposée pour être intégrée dans un émetteur-récepteur cognitif de futures générations 5G.

#### 4.4 Évaluation de la consommation d'énergie

L'efficacité énergétique est considérée une des principales exigences des futurs systèmes de communication 5G. C'est pour cela que nous évaluons la performance de l'architecture proposée en termes de l'énergie consommée, et la comparons à l'« Architecture I ». À l'aide de l'outil Xilinx PlanAhead Power Estimator, nous générons un rapport sur la puissance consommée par les deux solutions. Le principe de cet outil est de trouver, à une température et une fréquence d'opération données, une estimée de la puissance consommée à partir des ressources utilisées. Un aperçu est présenté au tableau 4.3.

**Tableau 4.3 – Résultats de l'estimation de la puissance consommée et la fréquence maximale d'opération à la température 25°C.**

	<b>Architecture I</b>	<b>Architecture proposée</b>
<b>Puissance dynamique (mW) à 61.44 MHz</b>	1191.43	741.96
<b>Puissance statique (mW) at 61.44 MHz</b>	7489.74	
<b>Énergie consommée pour 1000 cycles d'horloge (<math>\mu</math>J)</b>	141.33	133.98
<b>Fréquence max. d'opération (MHz)</b>	125.33	149.12

La puissance consommée dans un FPGA est divisée en deux parties: une puissance statique et une puissance dynamique. La première représente la puissance requise pour faire fonctionner le FPGA et le programmer. Une grande partie est due au phénomène de « leakage » dans les transistors utilisés pour maintenir la configuration du FPGA. D'autre part, la puissance dynamique représente une quantité supplémentaire due à l'exécution du design. Elle dépend principalement de l'alimentation, les ressources matérielles consommées, et le routage utilisé dans le FPGA.

Comme le montre le tableau 4.3, l'architecture proposée (idem., optimisée) présente un gain de puissance dynamique de 38% par rapport à celle non-optimisée. En outre, ce gain se traduit

à une économie de 6% de l'énergie consommée pendant 1000 cycles d'horloge pour un design qui fonctionne sur une fréquence d'opération de 61.44 MHz. Par conséquent, cette économie confirme l'efficacité des différentes optimisations effectuées pour réduire l'énergie consommée par le module sensoriel du CTR et montre la validité de l'architecture proposée pour être intégrée dans un système de communication 5G cognitif.

## 4.5 Évaluation de la latence computationnelle

La latence de calcul est définie comme étant le temps demandé par un circuit de traitement numérique du signal pour fournir un résultat valide. Plusieurs applications exigent une latence computationnelle très faible afin d'assurer une meilleure qualité d'expérience (QoE) pour l'utilisateur, telles que les véhicules autonomes, communication machine-à-machine, les applications de réalité virtuelle/augmentée, etc. C'est pour cela que dans cette partie, nous évaluons la performance de notre architecture proposée en termes de latence de calcul. Les résultats de cette évaluation sont présentés dans le tableau 4.4.

**Tableau 4.4** – Latence computationnelle de l'architecture proposée à une fréquence d'opération de 61.44 MHz,  $N = 128$ ,  $M = 8$ ,  $L = 3$ ,  $min = 0$ ,  $max = 10000$ ,  $coarse\_step = 400$ , et  $fine\_step = 40$ .

Module	Nb. des cycles d'horloge	Latence ( $\mu$ sec)
Estimation du canal	148	2.40796
Estimation de la puissance du bruit	135	2.19645
Estimation du SNR	36	0.586
Estimation de l'étalement de Doppler	6643	108.27202
Sous-module LLF	138	2.2460
<b>Total d'une estimation locale</b>	<b>6962</b>	<b>113.46198</b>
Moyennage	9	0.14643
<b>Total</b>	<b>55705</b>	<b>906.6522</b>

Comme le montre le tableau 4.4, les modules « Estimation du canal » et « Estimation de la puissance du bruit » demandent seulement 148 et 135 cycles d'horloge, respectivement, tandis que le module « Estimation du SNR » prend 36 cycles pour fournir des estimées valides. Cette faible latence de calcul s'explique par le fait que ces modules intègrent des structures matérielles parallèles, profondément pipelinées, et qui ne comportent aucune mémoire tampon. Le nombre de cycles nécessaires est principalement lié à la taille de la fenêtre d'approximation locale ( $N$ ), puisque la projection vecteur-par-vecteur représente l'opération la plus coûteuse en termes de temps d'exécution.

D'autre part, le sous-module « LLF » prend seulement 138 cycles d'horloge pour calculer la fonction LLF dans une seule valeur candidate  $\sigma_D$ . Cependant, la latence du module « Estimation de l'étalement de Doppler » représente 95% de tout le temps de traitement. Ceci est expliqué par le fait que nous n'avons pas une expression explicite de l'estimée de l'étalement de Doppler, et qu'il est nécessaire d'évaluer la LLF sur un large intervalle des valeurs de  $\sigma_D$ . Nous considérons un facteur de latence  $Q$  indiquant le nombre de fois que la LLF est exécutée. Ce facteur s'écrit comme suit :

$$Q = \left\lfloor \frac{max - min}{coarse\_step} + 2 \times \frac{coarse\_step}{fine\_step} \right\rfloor. \quad (4.1)$$

Nous constatons à partir de l'équation (4.1) que la latence de tout le design est fortement liée au choix des paramètres de la recherche sur l'argmax, c'est-à-dire l'intervalle  $[min, max]$  et les pas  $coarse\_step$  et  $fine\_step$ .

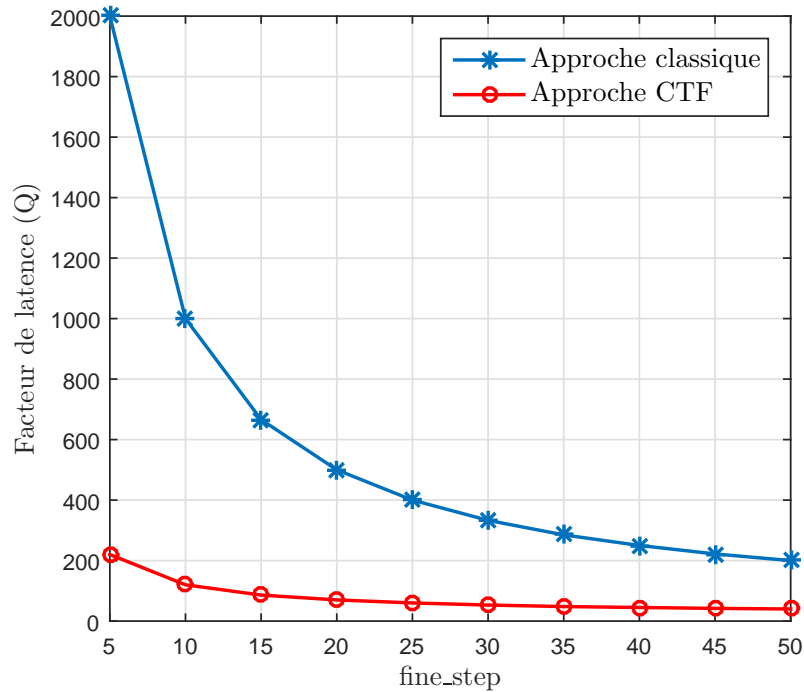


Figure 4.1 – Variation du facteur de latence  $Q$  en fonction du pas  $fine\_step$  pour les deux approches de recherche sur le maximum, avec  $coarse\_step = 10 \times fine\_step$ ,  $max = 10000$ , et  $min = 0$ .

L'avantage de l'approche de recherche CTF est de réduire la valeur de ce facteur  $Q$ . La figure 4.1 représente la variation de  $Q$  en fonction du pas  $fine\_step$  pour l'approche classique (idem., recherche avec un pas fixe) et l'approche CTF où nous considérons toujours  $coarse\_step = 10 \times fine\_step$ .

Nous remarquons que grâce à l’approche CTF et sans dégradation de la précision de l’estimée, nous réduisons la latence du module « Estimation de l’étalement de Doppler » d’un facteur de 10. Ceci permet d’avoir un temps d’exécution d’une estimation locale de l’ordre de  $113 \mu\text{sec}$ . En gros, le design proposé demande seulement  $0.9 \text{ msec}$  pour donner une estimation du canal, du SNR, et de l’étalement de Doppler sur  $M = 8$  fenêtres d’approximation de taille  $N = 128$ .

## 4.6 Évaluation de la précision de l’estimation

Pour évaluer la performance de notre prototype en termes de précision d’estimation, nous effectuons une étude comparative, en utilisant l’erreur quadratique moyenne normalisée (NMSE) comme mesure de performance, entre ces deux configurations:

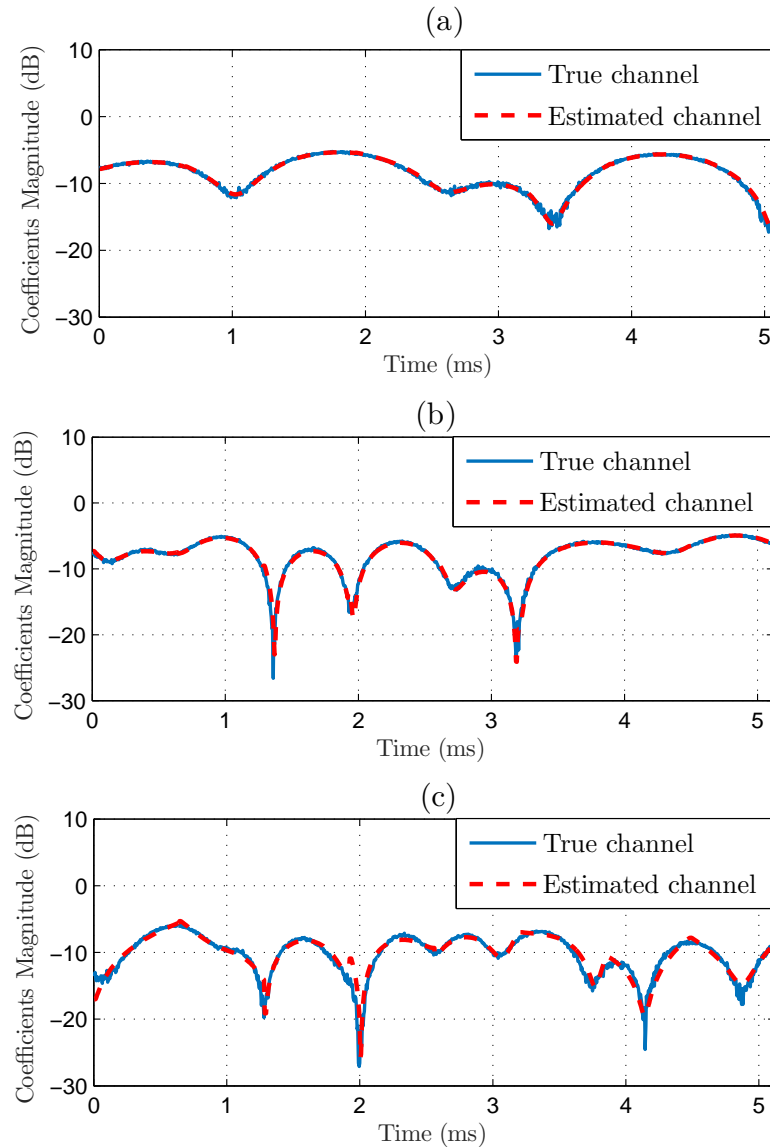
- **Simulations MATLAB:** les résultats d’estimation sont obtenus à l’aide des simulations MATLAB à virgule flottante. Le canal sans fil est généré à l’aide des scripts MATLAB;
- **Co-simulation mBEE4/EB Prosim:** dans cette configuration, le canal est émulé à l’aide de l’émulateur EB Prosim F8, tandis que la tâche d’estimation est effectuée au sein de la carte FPGA de la plateforme mBEE4 SDR.

Le modèle de canal considéré est le même pour chacune des versions MATLAB et mBEE4/EB Prosim. La NMSE est calculée sur  $M_c = 10000$  boucles de simulation Monte-Carlo. Nous vérifions la performance du prototype proposé pour:

- Identifier le canal  $\mathbf{h}$ ;
- Estimer le SNR instantané,  $\hat{\rho}$ ;
- Estimer l’étalement de Doppler,  $\hat{\sigma}_D$ .

### 4.6.1 Résultats de l’identification du canal de transmission

Nous étudions, dans cette partie, l’effet de la fréquence maximale de Doppler ( $f_D$ ) sur la performance de l’identification du canal de transmission. Les résultats de la co-simulation mBEE4/EB Prosim sont illustrés dans la figure 4.2, où nous traçons le gain du canal de Rayleigh réel généré par l’émulateur EB Prosim contre le canal estimé. Nous remarquons que pour différentes valeurs de ( $f_D$ ), le design proposé permet de bien identifier le canal de transmission aux positions pilotes.



**Figure 4.2 – Canal estimé vs. canal vrai pour (a)  $f_D = 400$  Hz, (b)  $f_D = 800$  Hz, and (c)  $f_D = 1200$  Hz.**

Les autres coefficients du canal correspondant aux positions non-pilotes peuvent être obtenus par interpolation.

Bien que l'estimation des canaux sans-fil devient un problème difficile dans une situation de mobilité élevée, la figure 4.2.(c) confirme que la conception proposée est capable d'identifier correctement les canaux très variables dans le temps, c'est-à-dire lorsque la mobilité de l'utilisateur peut atteindre 500 Km/h.

#### 4.6.2 Résultats de l'estimation du SNR instantanée

Dans la figure 4.3, nous étudions l'impact du SNR moyen (c'est-à-dire le SNR à long terme) sur l'estimation du SNR instantané pour les deux configurations de simulation. Le SNR moyen est défini comme suit:

$$\text{SNR} = \frac{E \left\{ |x(n)|^2 \right\}}{2\sigma_n^2}, \quad (4.2)$$

où  $E \{.\}$  indique l'espérance sur tous les symboles transmis. En considérant une constellation à énergie normalisée (idem.,  $E \left\{ |x(n)|^2 \right\} = 1$ ), le SNR moyen s'écrit comme suit:

$$\text{SNR} = \frac{1}{2\sigma_n^2}. \quad (4.3)$$

Nous remarquons que les deux courbes de MATLAB et mBEE4/EB Propsim sont pratiquement confondues et suivent les mêmes variations sur un large intervalle du SNR moyen (idem., entre -10 dB et 20 dB). Ceci confirme la validité de l'architecture matérielle proposée pour l'estimateur DA ML SNR choisi et prouve sa validité dans des conditions de propagation réelles.

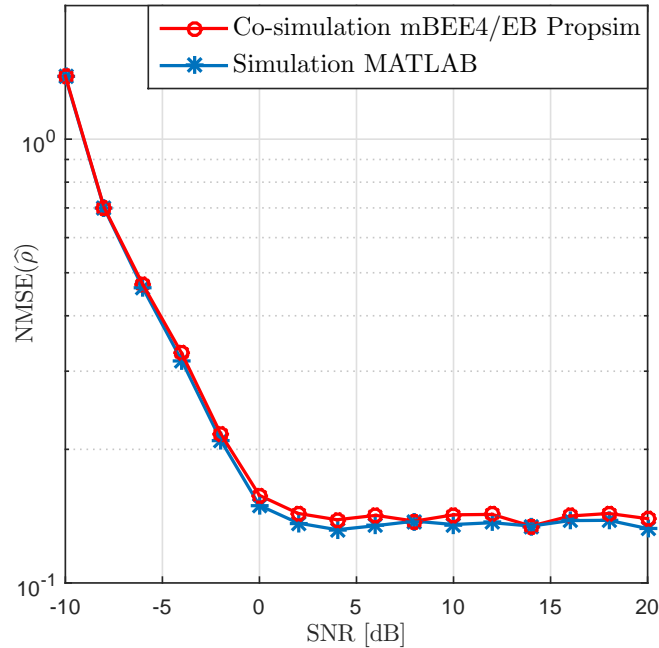


Figure 4.3 – NMSE du SNR instantané obtenue par la co-simulation mBEE4/EB Propsim contre les simulations MATLAB en fonction du SNR moyen, avec  $f_D = 200$  Hz and  $L = 3$ .

Dans la figure 4.4, nous traçons l’histogramme normalisé illustrant la distribution des estimées du SNR instantané (en dB) pour les deux configurations de simulation, à  $f_D = 200$  Hz et à SNR = 0 dB. À première vue, les résultats obtenus par co-simulation matérielle se coïncident avec ceux obtenus par simulation MATLAB à virgule flottante. Les réalisations de chaque configuration présentent des statistiques quasi égales en termes de moyenne et d’écart-type. D’autre part, comme il est montré dans la figure 4.4, les réalisations de la co-simulation mBEE4/EB Prosim suivent une distribution Gaussienne autour de la valeur moyenne, ce qui confirme la décorrélation entre les différentes réalisations (théorème central limite).

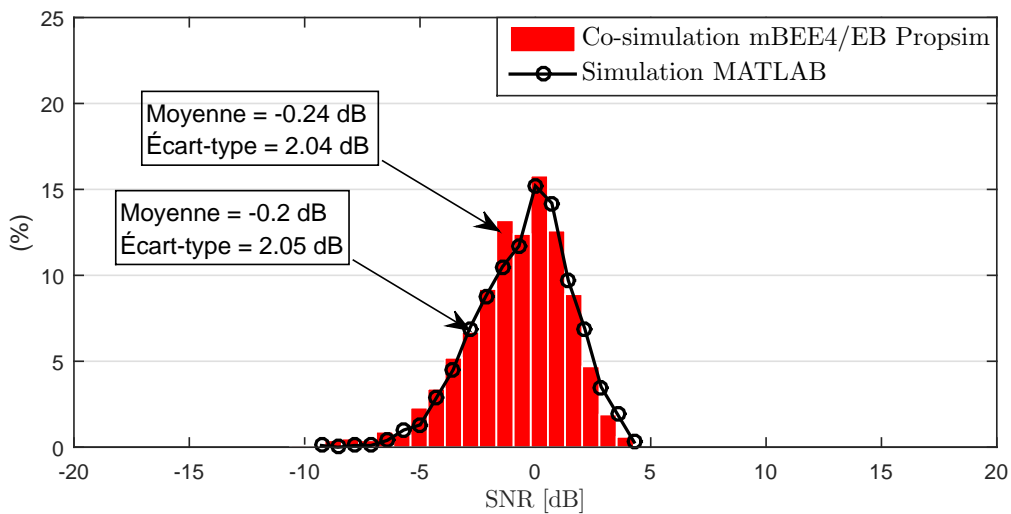


Figure 4.4 – Histogramme de l’estimée du SNR instantané à  $f_D = 200$  Hz, SNR = 0 dB, et  $M_c = 10000$ .

En outre, nous traçons dans la figure 4.5 la fonction de répartition cumulative (CDF) de la NMSE de l’estimée du SNR instantané à la fréquence maximale de Doppler  $f_D = 200$  Hz et à un niveau de SNR moyen SNR = 0 dB. En fait, la CDF d’une variable aléatoire  $X$  est définie comme suit:

$$F_X(x) = \text{Prob}(X \leq x). \quad (4.4)$$

Nous constatons que les deux CDFs sont presque confondues et que 90% des estimées présentent une erreur NMSE de moins de 0.5, ce qui se traduit par un écart maximal de  $\pm 2$  dB autour de la vraie valeur du SNR moyen (c’est-à-dire autour de 0 dB). Ce résultat est considéré largement précis

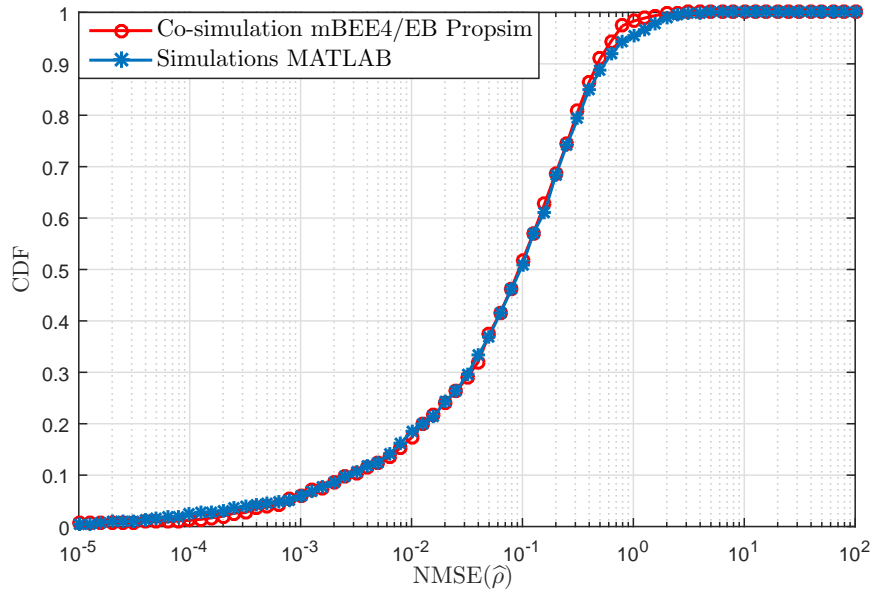


Figure 4.5 – CDF de la NMSE du SNR instantané à  $f_D = 200$  Hz et SNR = 0 dB.

pour fournir au module de décision du CTR une information correcte sur le niveau du SNR (voir les règles de décision à la figure 2.2). Par conséquent, nous confirmons le choix de cet estimateur et validons sa performance pour être intégré dans le prototype final du CTR.

### 4.6.3 Résultats de l'estimation de l'étalement de Doppler

La figure 4.6 représente la NMSE de la fréquence de Doppler estimée en fonction du SNR moyen pour la simulation MATLAB et la co-simulation basée sur le matériel. Nous constatons que les deux courbes présentent un léger écart dans la performance d'estimation pour les valeurs élevées du SNR moyen, mais elles suivent globalement les mêmes variations. Cet écart est principalement dû aux imperfections matérielles introduites par la plateforme SDR, l'émulateur de canal, et les interconnexions entre eux.



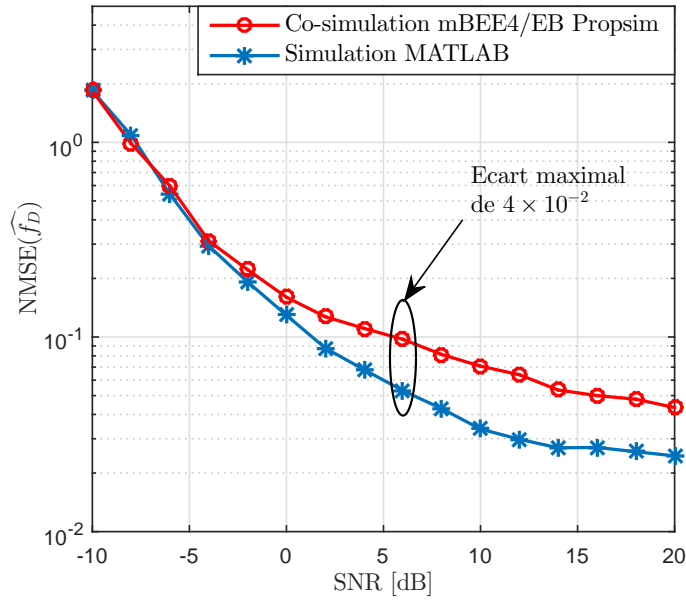


Figure 4.6 – NMSE de la fréquence maximale de Doppler obtenue par la co-simulation mBEE4/EB Propsim contre les simulations MATLAB en fonction du SNR moyen, avec  $f_D = 200$  Hz and  $L = 3$ .

Dans la figure 4.7, nous représentons la NMSE de la fréquence maximale de Doppler estimée contre la fréquence réelle à  $\text{SNR} = 0$  dB. La comparaison entre les deux configurations de simulation montre que même dans des mauvaises conditions de transmission (idem.,  $\text{SNR} = 0$  dB) et à différentes valeurs de la fréquence de Doppler, les deux courbes se coïncident et suivent le même comportement. Encore, le léger écart entre elles est dû aux imperfections matérielles introduites par le FPGA, les parties frontales de la plateforme et l’émulateur de canal. De plus, comme la figure 4.7 le montre, l’effet de ces imperfections est beaucoup plus important dans des situations de faible mobilité.

Nous traçons l’histogramme normalisé des estimées de la fréquence maximale de Doppler, à  $\text{SNR} = 0$  dB, pour une vraie valeur  $f_D = 100$  Hz et  $f_D = 400$  Hz, respectivement, dans les figures 4.8.(a) et 4.8.(b). Nous constatons que les deux simulations (matérielle et logicielle) présentent pratiquement les mêmes statistiques, pour les deux cas. De plus, la figure 4.8 montre que les estimées de la fréquence maximale de Doppler suivent une distribution Gaussienne autour de la vraie valeur, ce qui confirme encore la décorrélation entre les différentes réalisations.

Cependant, un léger écart est toujours présent entre les résultats de la co-simulation mBEE4/EB Propsim et ceux obtenus par simulation MATLAB. Pour mieux voir l’effet de cet écart sur la

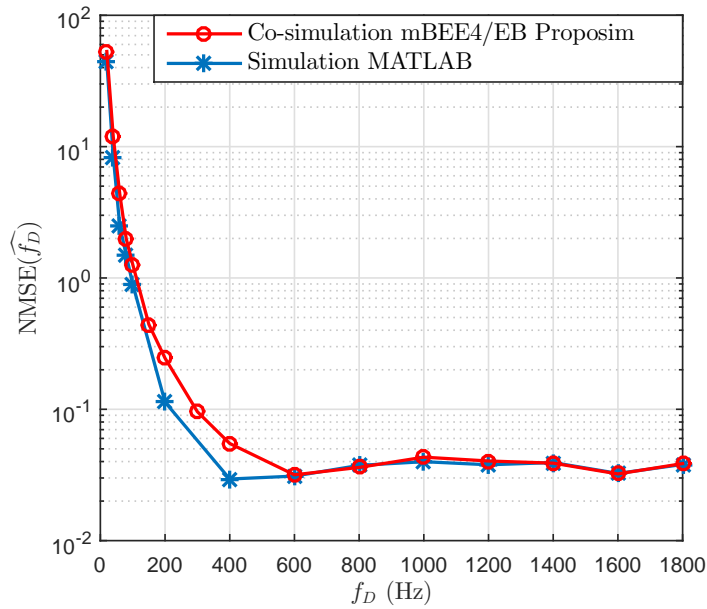


Figure 4.7 – NMSE de la fréquence maximale de Doppler obtenue par la co-simulation mBEE4/EB Proposim contre les simulations MATLAB en fonction de la fréquence de Doppler réelle, à SNR = 0 dB.

performance du module sensoriel de CTR, nous traçons, dans la figure 4.9 la CDF des estimées de la fréquence maximale de Doppler dans le cas où cet écart est maximal (idem., à  $f_D = 200$  Hz). Comme il est montré dans la figure 4.9, 83% des réalisations présentent une erreur NMSE de moins de 0.1, à SNR = 0 dB. Ceci se traduit par un écart d'estimation maximale de  $\pm 60$  Hz par rapport à la vraie valeur de 200 Hz. Autrement dit, au pire des cas, le module sensoriel fournit un estimé de la vitesse de mobile avec une erreur d'environ 26.42 Km/h. Par conséquent, nous pouvons négliger cet écart conformément aux règles de décision de CTR présentée dans la figure 2.2.

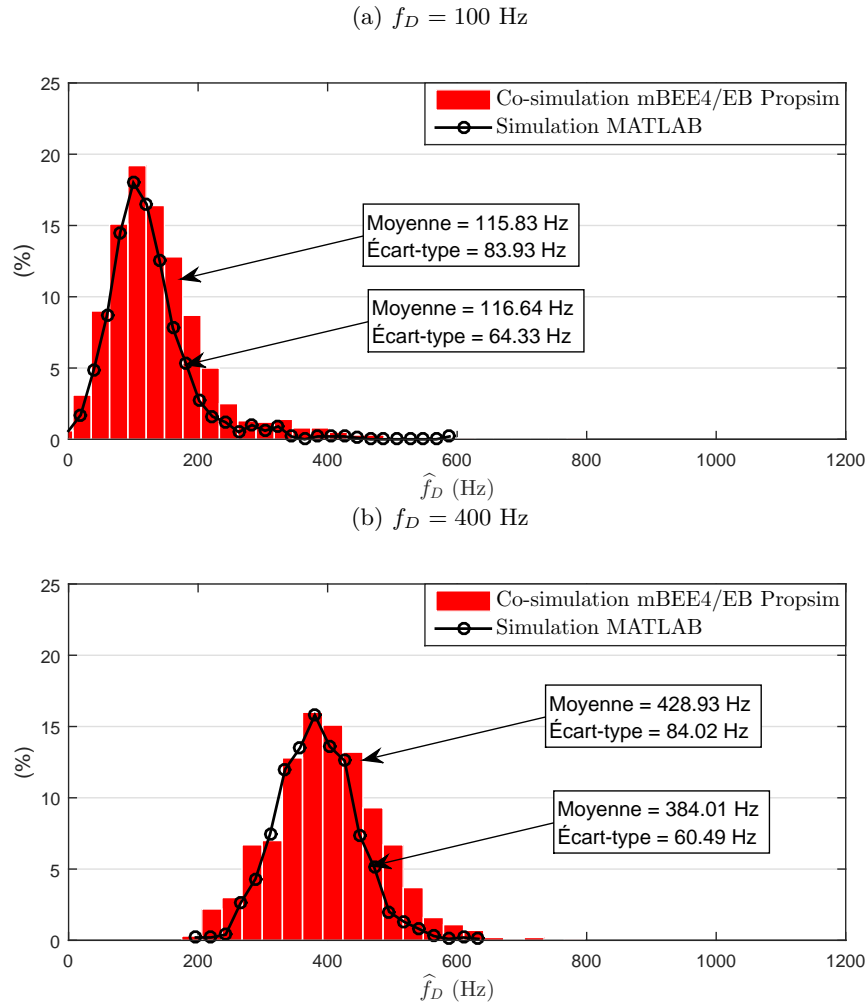


Figure 4.8 – Histogramme de l’estimée de la fréquence maximale de Doppler à SNR = 0 dB et  $M_c = 10000$ .

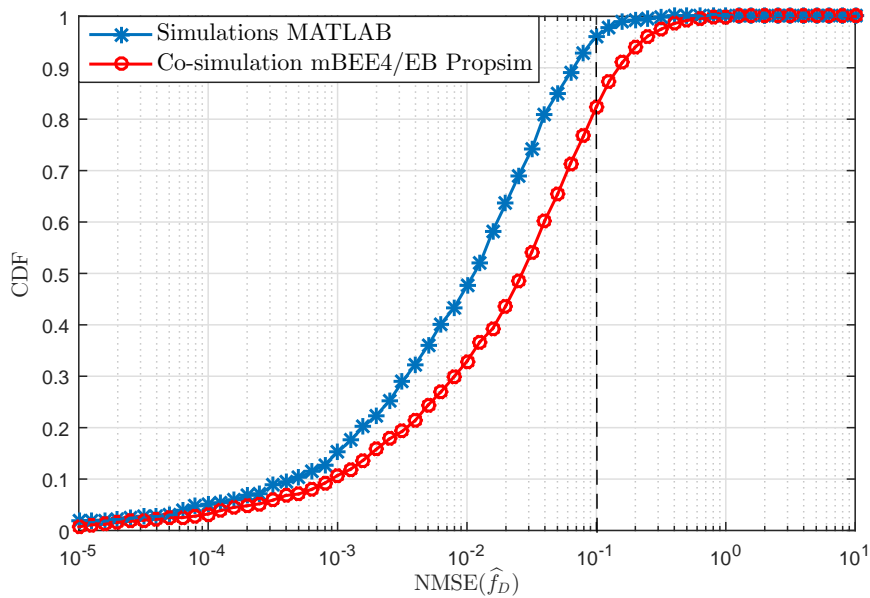


Figure 4.9 – CDF de la NMSE de la fréquence maximale de Doppler à  $f_D = 200$  Hz et SNR = 0 dB.

En résumant, les résultats de l'estimation du canal, du SNR instantané, et de l'étalement de Doppler valident l'architecture proposée pour l'intégrer dans le module sensoriel de CTR, et confirment aussi la robustesse des estimateurs choisis face à des conditions de propagation réelles.

## 4.7 Conclusion

Dans ce chapitre, nous avons présenté, dans une première étape, la phase d'implémentation de l'architecture proposée dans la plateforme SDR mBEE4 de BEEcube. Ensuite, nous avons procédé à l'évaluation de performance de cette architecture en termes de la complexité matérielle, la consommation énergétique, la latence computationnelle, et la précision d'estimation. Les résultats obtenus ont montré que l'architecture proposée présente un compromis attractif pour qu'elle soit implémentée dans le CTR.

# Conclusion générale

La future génération 5G des réseaux mobiles, définie comme le réseau « of everything », est non seulement une évolution des générations actuelles, mais aussi une révolution dans les domaines des TICs dans le sens où elle offrira des services de communication ultra fiables, sécurisés, à faible latence et à très large bande. Par conséquent, les fournisseurs de service se trouvent face à un ensemble de contraintes sévères telles que le temps de mise en marché, la flexibilité des solutions déployées, l'efficacité énergétique, etc. Ainsi, le prototypage rapide sur des plateformes définies par logiciel est considéré comme une solution efficace et à faible coût pour mettre en place des solutions de pointe telles que les réseaux cognitifs conscients du contexte (« context-aware cognitive networking »).

C'est dans ce contexte que le travail présenté dans ce mémoire s'inscrit. Il consiste à concevoir un prototype matériel du module sensoriel d'un nouvel émetteur-récepteur cognitif sur une plateforme radio-logicielle à base de FPGA. Ce module est composé de deux estimateurs pour le rapport signal-à-bruit et l'étalement de Doppler. Les deux estimateurs choisis sont basés sur la technique d'estimation à maximum de vraisemblance et ont montré un compromis attractif en termes de précision et complexité de calcul.

La conception matérielle de ces deux estimateurs a été réalisée avec une approche dite « basée sur modèle » (MBD), en utilisant l'outil Xilinx System Generator, ce qui nous a permis de concevoir et implémenter l'architecture proposée sans avoir recours aux méthodes classiques, coûteuses en termes de temps de développement, comme les langages HDLs. Nous avons implémenté cette architecture sur la plateforme radio-logicielle miniBEE4 de BEEcube Inc., et testé son fonctionnement en temps réel et sous des conditions de propagation réalistes grâce à l'émulateur de canal EB Proposim FS8.

Nous avons évalué la performance de notre design en termes d'utilisation des ressources matérielles, de consommation énergétique, de latence computationnelle, et de précision des estimées.

Les résultats obtenus ont montré que les différentes optimisations effectuées durant la phase de conception ont fourni un gain d'environ 80 % et de 34%, respectivement, en termes de complexité matérielle et de puissance dynamique consommée. En outre, l'évaluation de la latence de calcul de notre architecture a montré que le module sensoriel conçu fournit une information sur le SNR et l'étalement de Doppler en moins d'une milliseconde, un résultat très intéressant pour une éventuelle intégration de ce module dans un émetteur-récepteur de la future génération 5G. Finalement, nous avons comparé les résultats d'estimation obtenus par la co-simulation hardware-in-the-loop à ceux de MATLAB à virgule flottante. Cette comparaison a montré que le design matériel proposé estime les deux paramètres sous-jacents avec la même précision que MATLAB. Bien que les imperfections matérielles ont provoqué une légère perte au niveau de la précision d'estimation de l'étalement de Doppler, les résultats expérimentaux obtenus sont acceptables par l'application en question, c'est-à-dire par le module de décision, validant ainsi l'architecture proposée pour qu'elle soit intégrée dans le prototype final du nouvel émetteur-récepteur cognitif.

Plusieurs travaux de recherche et développement peuvent être considérés comme extensions de ce travail. Une première extension consiste à concevoir les autres modules du nouveau CTR et les interfacer avec le module sensoriel présenté dans ce mémoire. Nous avons déjà commencé à travailler sur la conception de la chaîne de transmission OFDM et de l'estimateur de canal aveugle à maximum de vraisemblance (ML NDA). Nous pourrions aussi dériver la formulation mathématique des deux estimateurs choisis en exploitant la diversité spatiale et fréquentielle fournie par les configurations OFDM-MIMO pour une future extension de l'architecture du nouveau CTR.

# Références

- [1] C. LIANG et F. R. YU : Wireless Network Virtualization: A Survey, Some Research Issues and Challenges. *IEEE Communications Surveys Tutorials*, 17(1):358–380, Mar 2015.
- [2] M. E. HOQUE : *Advanced applications of rapid prototyping technology in modern engineering*. InTech, 2011.
- [3] NOKIA : 5G use case and requirements - White Paper, 2014.
- [4] S. E. ELAYOUBI, M. FALLGREN, P. SPAPIS, G. ZIMMERMANN, D. MARTIN-SACRISTAN, C. YANG, S. JEUX, P. AGYAPONG, L. CAMPOY, Y. QI et S. SINGH : 5G service requirements and operational use cases: Analysis and METIS II vision. *In 2016 European Conference on Networks and Communications (EuCNC)*, pages 158–162, Juin 2016.
- [5] I. MARISSA, F. BELLILI, S. AFFES et A. STEPHENNE : Context-Aware Cognitive SIMO Transceiver for Increased LTE-Downlink Link-Level Throughput. *In 2015 IEEE International Conference on Ubiquitous Wireless Broadband (ICUWB)*, pages 1–5, Oct 2015.
- [6] A. ATI, F. BELLILI, H. HAGGAI, A. SAMET et S. AFFES : Implementation of a Maximum Likelihood Doppler Spread Estimator on a Model-Based Design Platform. *In 2015 IEEE International Conference on Ubiquitous Wireless Broadband (ICUWB)*, pages 1–5, Oct 2015.
- [7] Z. Ben GAMRA, F. BELLILI, A. SAMET et S. AFFES : Rapid prototyping and validation on an SDR platform of a low-cost hybrid ML SNR estimator over time-varying SIMO channels. *In 2016 IEEE International Conference on Ubiquitous Wireless Broadband (ICUWB)*, pages 1–4, Oct 2016.
- [8] J. MITOLA : The software radio architecture. *IEEE Communications Magazine*, 33(5):26–38, Mai 1995.
- [9] J. H. REED : *Software radio: a modern approach to radio engineering*. Prentice Hall Professional, 2002.
- [10] A. S. RODRIGUEZ, M. C. MENSINGER, I. S. AHN et Y. LU : Model-based software-defined radio(SDR) design using FPGA. *In 2011 IEEE International Conference on Electro/Information Technology*, pages 1–6, Mai 2011.
- [11] V. BHATNAGAR, G. S. OUEDRAOGO, M. GAUTIER, A. CARER et O. SENTIEYS : An FPGA Software Defined Radio Platform with a High-Level Synthesis Design Flow. *In 2013 IEEE 77th Vehicular Technology Conference (VTC Spring)*, pages 1–5, Juin 2013.
- [12] E. PAONE : Open-Source SCA Implementation-Embedded and Software Communication Architecture: OSSIE and SCA Waveform Development, 2010.

- [13] J. MITOLA : Software radios: Survey, critical evaluation and future directions. *IEEE Aerospace and Electronic Systems Magazine*, 8(4):25–36, Avril 1993.
- [14] O. ANJUM, T. AHONEN, F. GARZIA, J. NURMI, C. BRUNELLI et H. BERG : State of the art baseband DSP platforms for Software Defined Radio: A survey. *EURASIP Journal on wireless communications and networking*, 2011(1):5, 2011.
- [15] J. DELORME, J. MARTIN, A. NAFKHA, C. MOY, F. CLERMIDY, P. LERAY et J. PALICOT : A FPGA partial reconfiguration design approach for cognitive radio based on NoC architecture. *In 2008 Joint 6th International IEEE Northeast Workshop on Circuits and Systems and TAISA Conference*, pages 355–358, Juin 2008.
- [16] H. HARADA : Software defined radio prototype toward cognitive radio communication systems. *In First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005.*, pages 539–547, Nov 2005.
- [17] H. HARADA : A Software Defined Cognitive Radio Prototype. *In 2007 IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 1–5, Sept 2007.
- [18] H. KAESLIN : *Top-down Digital VLSI Design: From Architectures to Gate-level Circuits and FPGAs*. Morgan Kaufmann, 2014.
- [19] B. A. JONES : *Rapid prototyping of wireless communications systems*. Thèse de doctorat, Rice University, 2002.
- [20] P. A. SABATIER : Top-down and bottom-up approaches to implementation research: a critical analysis and suggested synthesis. *Journal of public policy*, 6(01):21–48, 1986.
- [21] P. COUSSY et A. MORAWIEC : *High-level synthesis: from algorithm to digital circuit*. Sépringer Science & Business Media, 2008.
- [22] F. GHENASSIA *et al.* : *Transaction-level modeling with SystemC*. Springer, 2005.
- [23] D. GAJSKI, J. ZHU, R. DÖMER, A. GERSTLAUER et S. ZHAO : *SpecC: Specification language and methodology*. Springer Science & Business Media, 2012.
- [24] Agility Design SOLUTIONS : *Handel-C Language Reference Manual*. Agility Design Solutions, 2007.
- [25] BEEcube INC. : *miniBEE SDR platform*, 2015.
- [26] J. KOLU, T. JAMSA et A. HULKKONEN : Real time simulation of measured radio channels. *In 2003 IEEE 58th Vehicular Technology Conference. VTC 2003-Fall (IEEE Cat. No.03CH37484)*, volume 1, pages 183–187 Vol.1, Oct 2003.
- [27] A. ALCOCER-OCHOA, V. Y. KONTOROVITCH et R. PARRA-MICHEL : A MIMO Channel Simulator Applying the Universal Basis. *In IEEE Vehicular Technology Conference*, pages 1–5, Sept 2006.
- [28] M. PATZOLD et B. O. HOGSTAD : A Wideband Space-Time MIMO Channel Simulator Based on the Geometrical One-Ring Model. *In IEEE Vehicular Technology Conference*, pages 1–6, Sept 2006.



- [29] J. M. CONRAT et P. PAJUSCO : A Versatile Propagation Channel Simulator for MIMO Link Level Simulation. *EURASIP J. Wirel. Commun. Netw.*, 2007(2):19–19, janvier 2007.
- [30] ANITE : User-friendly Emulator for MIMO Performance Testing, 2013.
- [31] J. MITOLA : *Cognitive Radio Architecture*. Springer Netherlands, Dordrecht, 2006.
- [32] W. KRENIK et A. BATRA : Cognitive radio techniques for wide area networks. *In Proceedings. 42nd Design Automation Conference, 2005.*, pages 409–412, Juin 2005.
- [33] J. W. HUANG et V. KRISHNAMURTHY : Cognitive Base Stations in LTE/3GPP Femtocells: A Correlated Equilibrium Game-Theoretic Approach. *IEEE Transactions on Communications*, 59(12):3485–3493, December 2011.
- [34] A. ATTAR, V. KRISHNAMURTHY et O. N. GHAREHSHIRAN : Interference management using cognitive base-stations for UMTS LTE. *IEEE Communications Magazine*, 49(8):152–159, Août 2011.
- [35] J. ZYREN et W. MCCOY : Overview of the 3GPP long term evolution physical layer. *Freescale Semiconductor, Inc., white paper*, 2007.
- [36] F. BELLILI, C. ELGUET, S. Ben AMOR et S. AFFES : Code-Aided DOA Estimation from Turbo-Coded QAM Transmissions: Analytical CRLBs and Maximum Likelihood Estimator. *IEEE Transactions on Wireless Communications*, accepted for publication, Jan 2017. to appear.
- [37] F. BELLILI, Y. SELMI, S. AFFES et A. GHAYEB : A Low-Cost and Robust Maximum Likelihood Joint Estimator for the Doppler Spread and CFO Parameters Over Flat-Fading Rayleigh Channels. *IEEE Transactions on Communications*, Decision: Major revision, Mar 2017. under review.
- [38] F. BELLILI, S. Ben AMOR, S. AFFES et A. SAMET : A new importance-sampling ML estimator of time delays and angles of arrival in multipath environments. *In 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4219–4223, Mai 2014.
- [39] A. MASMOUDI, F. BELLILI, S. AFFES et A. STEPHENNE : A Non-Data-Aided Maximum Likelihood Time Delay Estimator Using Importance Sampling. *IEEE Transactions on Signal Processing*, 59(10):4505–4515, Oct 2011.
- [40] M. A. BOUJELBEN, F. BELLILI, S. AFFES et A. STEPHENNE : SNR Estimation Over SIMO Channels From Linearly Modulated Signals. *IEEE Transactions on Signal Processing*, 58(12): 6017–6028, Déc 2010.
- [41] S. B. HASSEN, F. BELLILI, A. SAMET et S. AFFES : DOA Estimation of Temporally and Spatially Correlated Narrowband Noncircular Sources in Spatially Correlated White Noise. *IEEE Transactions on Signal Processing*, 59(9):4108–4121, Sept 2011.
- [42] F. BELLILI, S. AFFES et A. STEPHENNE : DOA Estimation for ULA Systems from Short Data Snapshots: An Annihilating Filter Approach. *In 2011 IEEE Global Telecommunications Conference - GLOBECOM 2011*, pages 1–5, Déc 2011.
- [43] A. STEPHENNE, F. BELLILI et S. AFFES : Moment-based SNR estimation over linearly-modulated wireless SIMO channels. *IEEE Transactions on Wireless Communications*, 9(2): 714–722, Fév 2010.

- [44] F. BELLILI, A. STEPHENNE et S. AFFES : SNR estimation of QAM-modulated transmissions over time-varying SIMO channels. *In 2008 IEEE International Symposium on Wireless Communication Systems*, pages 199–203, Oct 2008.
- [45] M. D. AUSTIN et G. L. STÜBER : Velocity adaptive handoff algorithms for microcellular systems. *IEEE Transactions on Vehicular Technology*, 43(3):549–561, 1994.
- [46] C. TEPEDLENLIOĞLU, A. ABDI, G. GIANNAKIS et M. KAVEH : Estimation of Doppler spread and signal strength in mobile communications with applications to handoff and adaptive transmission. *Wireless Communications and Mobile Computing*, 1(2):221–242, 2001.
- [47] K. BADDOUR et N. BEAULIEU : Robust Doppler spread estimation in nonisotropic fading channels. *Wireless Communications, IEEE Transactions on*, 4(6):2677–2682, 2005.
- [48] H. ZHANG et A. ABDI : A robust mobile speed estimator in fading channels: Performance analysis and experimental results. *In Global Telecommunications Conference, 2005. GLOBECOM'05. IEEE*, volume 5, pages 5–pp. IEEE, 2005.
- [49] K. ANIM-APPIAH : On generalized covariance-based velocity estimation. *Vehicular Technology, IEEE Transactions on*, 48(5):1546–1557, 1999.
- [50] W. SHENG et S. BLOSTEIN : SNR-independent velocity estimation for mobile cellular communications systems. *In Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, volume 3, pages III–2469. IEEE, 2002.
- [51] J. CAI, W. SONG et Z. LI : Doppler spread estimation for mobile OFDM systems in Rayleigh fading channels. *Consumer Electronics, IEEE Transactions on*, 49(4):973–977, 2003.
- [52] H. HANSEN, S. AFFES et P. MERMELSTEIN : A Rayleigh Doppler frequency estimator derived from maximum likelihood theory. *In Signal Processing Advances in Wireless Communications, 1999. SPAWC'99. 1999 2nd IEEE Workshop on*, pages 382–386. IEEE, 1999.
- [53] F. BELLILI et S. AFFES : A low-cost and robust maximum likelihood Doppler spread estimator. *In 2013 IEEE Global Communications Conference (GLOBECOM)*, pages 4325–4330, Déc 2013.
- [54] Y. R. TSAI, K. J. YANG, C. H. TSAI et C.L. WANG : Low-complexity ML Doppler spread estimation for OFDM systems. *In Vehicular Technology Conference (VTC Fall), 2011 IEEE*, pages 1–5. IEEE, 2011.
- [55] Yuh-Ren TSAI et Kai-Jie YANG : Approximate ML Doppler spread estimation over flat Rayleigh fading channels. *Signal Processing Letters, IEEE*, 16(11):1007–1010, 2009.
- [56] M. SOUDEN, S. AFFES, J. BENESTY et R. BAHROUN : Robust Doppler Spread Estimation in the Presence of a Residual Carrier Frequency Offset. *IEEE Transactions on Signal Processing*, 57(10):4148–4153, Oct 2009.
- [57] F. BELLILI, R. MEFTEHI, S. AFFES et A. STÉPHENNE : Maximum Likelihood SNR Estimation of Linearly-Modulated Signals Over Time-Varying Flat-Fading SIMO Channels. *IEEE Transactions on Signal Processing*, 63(2):441–456, Jan 2015.
- [58] Evolved Universal Terrestrial Radio Access NETWORK : 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access Network, 2011.

- [59] P. BELLO : Characterization of Randomly Time-Variant Linear Channels. *IEEE Transactions on Communications Systems*, 11(4):360–393, Déc 1963.
- [60] D. TSE et P. VISWANATH : *Fundamentals of wireless communication*. Cambridge university press, 2005.
- [61] S. PAUL, N. JAYAKUMAR et S. P. KHATRI : A Fast Hardware Approach for Approximate, Efficient Logarithm and Antilogarithm Computations. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 17(2):269–277, Fév 2009.