

Université du Québec  
Institut national de la recherche scientifique  
Centre Énergie Matériaux Télécommunications

**Prototypage rapide à base de FPGA et validation en temps réel et sur les ondes d'un estimateur à maximum de vraisemblance de l'étalement Doppler**

Par  
Adel Ati

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

**Jury d'évaluation**

Président du jury et examinateur interne	Monsieur Serge Tatu INRS Énergie Matériaux Télécommunications
Examineur externe	Monsieur Georges Kaddoum ÉTS
Directeur de recherche	Monsieur Sofiène Affes INRS Énergie Matériaux Télécommunications



# Remerciements

Ce travail a été élaboré dans le cadre d'un projet ENGAGE en collaboration avec l'entreprise Nutaq dans le but d'explorer et de mettre l'accent sur les possibilités offertes par le « Nutaq's model-based design kit (MBDK) » pour le prototypage rapide.

Je tiens à exprimer ma gratitude et mes respects les plus sincères à mon directeur de recherche, le **Professeur Sofiène Affes** pour l'aide qu'il a bien voulu m'accorder tout au long des différentes étapes de mes travaux de recherche, pour ses critiques constructives et ses suggestions pertinentes. Je m'incline par-devant ses qualités humaines et morales dont j'étais toujours impressionné, sa volonté permanente d'aider et ses idées innovantes qui m'ont permis d'élaborer mes travaux de recherches.

Je remercie également toute son équipe, et spécialement celle du prototypage, qui n'a pas cessé de m'encourager à bien avancer tout au long mes études. Je tiens à leur exprimer toute ma reconnaissance et gratitude.

J'exprime également ma gratitude la plus profonde pour **Mr Abdelaziz Samet** pour son soutien et son aide. Sa disponibilité, ses suggestions et ses conseils judicieux m'ont aidé à réussir ce mémoire.

Mes remerciements vont également aux membres de jury pour l'attention qu'ils voudraient porter à ce mémoire.

Je remercie mes parents, mes frères, et ma sœur pour leur soutien tout au long de mes études, j'espère qu'ils seront toujours fiers de moi.

Finalement, je voudrais dédier ce mémoire pour ma fiancée qui a été ma plus grande source d'inspiration tout au long de mes études.

# Résumé

Les nouvelles technologies sans fil sont maintenant en plein essor et jouent un rôle de plus en plus important dans notre mode de vie. L'intégration des systèmes embarqués dans le domaine des communications sans fil modernes a permis de répondre rapidement aux demandes toujours croissantes. L'invention de matrices pré-diffusées programmables par l'utilisateur (FPGAs) a rendu possible le concept du matériel reconfigurable pour la radio logicielle (SDR). En effet, les méthodes traditionnelles de conception des radios architecturées autour de FPGA ne permettent pas une productivité élevée. Cette dernière peut être améliorée en utilisant la technique de conception basée sur un modèle (MBD). Des outils basés sur cette technique sont ciblés pour le développement des systèmes embarqués, le développement des algorithmes de traitement du signal, l'intégration rapide et l'analyse du comportement des systèmes numériques complexes pour une grande variété de cas d'utilisation. En conséquence, de nombreuses plates-formes matérielles et d'outils logiciels ont été développés pour tirer profit de cette technique afin de réaliser un prototypage rapide.

Nous avons adopté cette approche de conception basée sur un modèle dans ce mémoire pour implémenter et valider un estimateur de l'étalement Doppler au maximum de vraisemblance développé au sein du Wirelesslab Group fondé au niveau de l'INRS et dirigé par M. Sofiène Affes. Le prototypage rapide de l'ensemble du design est fait en tant qu'un modèle Simulink haut niveau en utilisant des blocs de propriété intellectuelle propres à Xilinx offerts par l'outil Xilinx system generator (XSG). L'étage frontal aux radio-fréquences (RF) a été implémenté en utilisant le « Nutaq's model-based design kit (MBDK) ».

Durant la phase d'évaluation, nous avons développé un modèle de test sous Simulink afin de pouvoir préciser les paramètres globaux de notre design. Ce dernier a été testé pour une fréquence d'échantillonnage de 80 Msps avec un canal RF réaliste généré par l'émulateur de canal "EB Prosim FS8" et les résultats mesurés ont été comparés à ceux obtenus avec MATLAB.

**Mots-clés :** Systèmes embarqués, communications sans fil, matrices pré-diffusées programmables par l'utilisateur, radio logicielle, technique de conception basée sur un modèle, prototypage rapide, estimateur de l'étalement Doppler au maximum de vraisemblance, Simulink, Xilinx system generator, Nutaq's model-based design kit, émulateur de canal, MATLAB.

# Abstract

New wireless technologies are now booming and are playing an increasingly important role in our lifestyle. The integration of embedded systems in modern wireless communication systems has made it possible to respond quickly to its ever increasing demands. The invention of field programmable gate arrays (FPGAs) has made possible the concept of re-configurable hardware for software defined radio (SDR). Indeed, traditional methods of designing FPGA based radios do not allow high productivity. The latter can be improved using Model based design (MBD) technique. Tools based on this technique are targeted for embedded systems development, signal processing algorithm development, rapid integration and behavior analysis of complex digital systems for a wide variety of use cases. As a result, Many hardware platforms and Software tools have been developed to take benefit from this technique in order to achieve rapid prototyping.

We adopted this model-based design approach in this master's thesis to implement and validate a maximum likelihood (ML) Doppler spread estimator developed by the Wirelesslab Group based in the INRS and directed by Mr.Sofiène Affes. Rapid prototyping of the entire design is done as a high-level Simulink model using Xilinx System Generator IP blocks. The RF front-end of the design has been implemented using the Nutaq's model-based design kit (MBDK).

In the evaluation phase, we have developed a test model in Simulink to be able to specify the global parameters of our design. The latter has been tested at a sampling rate of 80 Msps with a realistic RF channel generated by the "EB Propsim FS8" channel emulator and the measured results have been compared to those obtained with MATLAB.

**Key-words :** Embedded systems, wireless communication, field programmable gate arrays, software defined radio, Model based design technique, rapid prototyping, maximum likelihood Doppler spread estimator, Simulink, Xilinx system generator, Nutaq's model-based design kit, channel emulator, MATLAB.

# Table des matières

<b>Avant-propos</b>	<b>iii</b>
<b>Résumé</b>	<b>iv</b>
<b>Abstract</b>	<b>vi</b>
<b>Table des matières</b>	<b>vii</b>
<b>Liste des figures</b>	<b>ix</b>
<b>Liste des abréviations</b>	<b>xi</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Prototypage rapide des systèmes de communication sans fil embarqués: Technique de « la conception basée sur un modèle » (MBD)</b>	<b>4</b>
2.1 Introduction . . . . .	4
2.2 La radio logicielle et la technique MBD . . . . .	5
2.2.1 La radio logicielle . . . . .	5
2.2.2 La technique MBD . . . . .	6
2.2.3 Les outils de la conception basée sur un modèle . . . . .	7
2.3 Le flot de conception de systèmes basée sur FPGA . . . . .	9
2.3.1 Le flot de conception traditionnel . . . . .	9
2.3.2 Le flot de conception MBD . . . . .	10
2.4 Conclusion . . . . .	13
<b>3 Description de l'estimateur de l'étalement Doppler et présentation de notre station du travail</b>	<b>14</b>
3.1 Introduction . . . . .	14
3.2 Description de l'estimateur de l'étalement Doppler choisi . . . . .	15
3.2.1 Modèle du système . . . . .	16
3.2.2 Formulation de l'estimateur de l'étalement Doppler . . . . .	17
3.3 Présentation de notre station du travail . . . . .	19
3.3.1 La plateforme de la radio-logicielle: PicoSDR 2x2 . . . . .	19
3.3.2 L'émulateur de canal radio . . . . .	24
3.4 Conclusion . . . . .	26
<b>4 Architecture globale et implémentation de l'estimateur de l'étalement Doppler</b>	<b>27</b>
4.1 Introduction . . . . .	27
4.2 Architecture globale du design . . . . .	27

4.3	Implémentation de l'estimateur de l'étalement Doppler . . . . .	29
4.3.1	Chemin des données . . . . .	29
4.3.2	Chemin de contrôle . . . . .	35
4.4	Génération du fichier « bitstream » et problèmes rencontrés . . . . .	36
4.5	Conclusion . . . . .	37
<b>5</b>	<b>Évaluation des performances</b>	<b>38</b>
5.1	Introduction . . . . .	38
5.2	Modèle Simulink de test . . . . .	38
5.3	Résultats de la co-simulation . . . . .	40
5.3.1	NMSE en fonction de $FdT_s$ . . . . .	40
5.3.2	NMSE en fonction du SNR . . . . .	44
5.3.3	Histogramme de l'estimée de la fréquence Doppler maximale normalisée . . . . .	46
5.3.4	Histogramme de l'erreur quadratique normalisée (NSE) de l'estimée de la fréquence maximale de Doppler . . . . .	50
5.3.5	La fonction de répartition (CDF) de NSE . . . . .	54
5.4	Conclusion . . . . .	55
<b>6</b>	<b>Conclusion générale</b>	<b>57</b>
	<b>Références</b>	<b>59</b>



# Liste des figures

2.1	Organigramme de la conception d'un système DSP générique. . . . .	7
2.2	Flot de conception traditionnel. . . . .	10
2.3	Flot de conception MBD pour FPGA. . . . .	11
3.1	NMSE vs. $F_d T_s$ avec $T_s = 10 \mu s$ , $SNR = 0 dB$ , et $N = 1000$ . . . . .	16
3.2	La plateforme PicoSDR 2x2 [32]. . . . .	19
3.3	Diagramme de l'architecture matérielle de la plateforme PicoSDR 2x2 [32]. . . . .	20
3.4	Diagramme de BPS du Nutaq [36]. . . . .	22
3.5	Nutaq MBDK & BSDK flows [39]. . . . .	23
3.6	Les émulateurs EB Prosim F32, EB Prosim FS8, et EB Prosim F8 [40]. . . . .	25
3.7	Schéma de remplacement du canal radio par "EB Prosim" [40]. . . . .	25
4.1	Diagramme en bloc du design. . . . .	28
4.2	Diagramme en bloc du module d'estimation de l'étalement Doppler. . . . .	29
4.3	Module d'estimation de l'étalement Doppler. . . . .	30
4.4	Diagramme Xilinx du sous-module EIGENVALUES. . . . .	31
4.5	Diagramme Xilinx de sous-module EIGENVECTORS. . . . .	32
4.6	Diagramme Xilinx du sous-module LLF. . . . .	33
4.7	Diagramme Xilinx du sous-module ARGMAX. . . . .	34
4.8	Diagramme Xilinx du sous-module AVERAGING. . . . .	34
4.9	Diagrammes d'états des différentes FSMs. . . . .	36
4.10	Diagramme Xilinx du design. . . . .	37
5.1	Modèle Simulink de test . . . . .	39
5.2	NMSE vs. $F_d T_s$ avec $T_s = 10 \mu s$ , $SNR = -10 dB$ et $N = 1000$ . . . . .	41
5.3	NMSE vs. $F_d T_s$ avec $T_s = 10 \mu s$ , $SNR = -5 dB$ et $N = 1000$ . . . . .	41
5.4	NMSE vs. $F_d T_s$ avec $T_s = 10 \mu s$ , $SNR = 0 dB$ et $N = 1000$ . . . . .	42
5.5	NMSE vs. $F_d T_s$ avec $T_s = 10 \mu s$ , $SNR = 5 dB$ et $N = 1000$ . . . . .	42
5.6	NMSE vs. $F_d T_s$ avec $T_s = 10 \mu s$ , $SNR = 10 dB$ et $N = 1000$ . . . . .	43
5.7	NMSE vs. $F_d T_s$ avec $T_s = 10 \mu s$ , $SNR = 20 dB$ et $N = 1000$ . . . . .	43
5.8	NMSE vs. SNR avec $F_d T_s = 0.006$ et $N = 1000$ . . . . .	44
5.9	NMSE vs. SNR avec $F_d T_s = 0.008$ et $N = 1000$ . . . . .	45
5.10	NMSE vs. SNR avec $F_d T_s = 0.01$ et $N = 1000$ . . . . .	45
5.11	Histogramme de l'estimée de la fréquence Doppler maximale normalisée: $F_d T_s = 0.006$ , $SNR = 0 dB$ , $N = 1000$ et $MC = 10000$ . . . . .	47
5.12	Histogramme de l'estimée de la fréquence Doppler maximale normalisée: $F_d T_s = 0.006$ , $SNR = 10 dB$ , $N = 1000$ et $MC = 10000$ . . . . .	48

5.13	Histogramme de l'estimée de la fréquence Doppler maximale normalisée: $F_d T_S = 0.006$ , $SNR = 20 \text{ dB}$ , $N = 1000$ et $MC = 10000$ . . . . .	49
5.14	Histogramme de NSE de l'estimée de la fréquence maximale de Doppler: $F_d T_S = 0.006$ , $SNR = 0 \text{ dB}$ , $N = 1000$ et $MC = 10000$ . . . . .	51
5.15	Histogramme de NSE de l'estimée de la fréquence maximale de Doppler: $F_d T_S = 0.006$ , $SNR = 10 \text{ dB}$ , $N = 1000$ et $MC = 10000$ . . . . .	52
5.16	Histogramme de NSE de l'estimée de la fréquence maximale de Doppler: $F_d T_S = 0.006$ , $SNR = 20 \text{ dB}$ , $N = 1000$ et $MC = 10000$ . . . . .	53
5.17	CDF vs. NSE avec $F_d T_s = 0.006$ , $SNR = 0 \text{ dB}$ et $N = 1000$ . . . . .	54
5.18	CDF vs. NSE avec $F_d T_s = 0.006$ , $SNR = 10 \text{ dB}$ et $N = 1000$ . . . . .	55
5.19	CDF vs. NSE avec $F_d T_s = 0.006$ , $SNR = 20 \text{ dB}$ et $N = 1000$ . . . . .	55

# Liste des abréviations

ADC	Analog-to-Digital Converter
AMC	Advanced Mezzanine Card
API	Application Programming Interface
BSDK	Board Software Development Kit
BSP	Board Support Package
CCE	Central Communication Engine
CDMA	Code Division Multiple Access
DAC	Digital-to-Analog Converter
DSP	Digital Signal Processing
FDD	Frequency Division Duplex
FMC	FPGA Mezzanine Card
FPGA	Field Programmable Gate Array
FSM	Finite State Machine
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
GUI	Graphical User Interface
HDL	Hardware Description Language
IP	Intellectual Property
JTAG	Joint Test Action Group
JTRS	Joint Tactical Radio System
LCR	Level-Crossing Rate
LLF	Log-Likelihood Function
LNA	Low Noise Amplifier
LTE	Long Term Evolution
MBDK	BModel Based Design Kit
MDB	Model-Based Design

MIMO	Multiple-Input Multiple-Output
ML	Maximum Likelihood
MV	Maximum de vraisemblance
NMSE	Normalized Mean Square Error
NSE	Normalized Square Error
OTA	Over-The-Air
PA	Power Amplifier
PIM	Platform Independent Model
PSD	Power Spectral Density
PSM	Platform Specific Model
RF	Radio Frequency
RTDEx	Real-Time Data Exchange
SDK	Software Development Kit
SDR	Software Defined Radio
SNR	Signal-to-Noise Ratio
TDD	Time Division Duplex
UML	Unified Modeling Language
UMTS	Universal Mobile Telecommunications System
USB	Universal Serial Bus
VHDL	VHSIC (Very High Speed Integrated Circuit) Hardware Description Language
WiFi	Wireless Fidelity
WiMAX	Worldwide interoperability for Microwave Access
XSG	Xilinx System Generator

# Chapitre 1

## Introduction

L'estimation des paramètres du canal sans fil est un facteur clé pour l'optimisation de l'émetteur-récepteur dans les systèmes de communication mobiles. L'environnement de propagation sans fil est généralement caractérisé par un canal multi-trajets variant dans le temps dont le taux d'évanouissement est essentiellement liée à l'étalement Doppler. La connaissance ou encore l'estimation à priori de l'étalement Doppler est donc une exigence pour de nombreuses applications telles que l'estimation de canal adaptative [1], les systèmes du contrôle de puissance et de commutation [2, 3]; pour n'en nommer que quelques-unes. Plusieurs estimateurs de l'étalement Doppler ont été proposés dans la littérature (voir [4, 5] et les autres références citées) chacun ayant ses propres avantages et inconvénients. Cependant, c'est l'estimateur qui montre le compromis le plus attractif entre la complexité de calcul et la précision de l'estimation qui est habituellement considéré comme le meilleur candidat pour l'implémentation matérielle dans la pratique. Motivé par ce fait, nous avons choisi d'implémenter le très récent estimateur de l'étalement Doppler proposé dans [6]. En fait, ce dernier est dérivé de la théorie du maximum de vraisemblance (MV) et présente de bonnes performances sur une large plage de valeurs Doppler pratiques. L'aspect le plus intéressant se manifeste par le fait que cet estimateur n'implique aucune inversion de matrices, contrairement à tous les estimateurs de type MV existants. Mis à part ses avantageux compromis en termes de précision/complexité, c'est aussi le premier de son genre en termes de robustesse envers le modèle de mobilité du canal puisqu'il ne nécessite pas une connaissance à priori de ce modèle offrant ainsi un degré de liberté très précieux dans la pratique.

Récemment, les radios logicielles (SDRs) ont été largement adoptées pour le prototypage rapide des applications de communication sans fil. En fait, l'immense progrès technologique dans les domaines de dispositifs re-configurables et de traitement du signal numérique (DSP) a ouvert de nouvelles possibilités au niveau de la conception et l'implémentation des systèmes de communication sans fil [7, 8]. La matrice pré-diffusée programmable par l'utilisateur (FPGA) est considérée comme le dispositif le plus efficace pour les applications radios logicielles en raison de sa performance, sa faible consommation d'énergie, et sa re-configurabilité [9]. Pourtant, les approches classiques de l'implémentation des circuits basés sur FPGA reposent sur des langages de bas niveau (VHDL, Verilog) qui souffrent d'une mauvaise flexibilité et d'une faible productivité. Récemment, une nouvelle approche du flot de conception descendant a été proposée afin de surmonter ce problème. Cette approche repose sur l'environnement Simulink de haut niveau qui offre une conception basée sur un modèle et exploite un ensemble de blocs existants de traitement du signal numérique. Par conséquent, l'implémentation matérielle de circuits plus sophistiqués peut désormais être facilement conçue par une procédure "drag-and-drop" en utilisant ces blocs déjà existants. Les codes correspondants en langage de description du matériel (HDL) peuvent alors être générés automatiquement par des outils puissants tels que Xilinx System Generator (XSG) [10].

Dans le présent mémoire, nous adoptons cette nouvelle approche de conception basée sur un modèle pour implémenter et valider un estimateur de l'étalement Doppler au maximum de vraisemblance développé au sein du Wireless Lab sur une plateforme radio logicielle basée sur FPGA. Au meilleur de notre connaissance, c'est la toute première implémentation matérielle d'un estimateur de l'étalement Doppler dans la littérature. En outre, nous avons utilisé, pendant la phase d'évaluation, un émulateur de canal qui imite les canaux radios du monde réel et permet de faire des tests de performances over-the-air (OTA).

Afin de présenter clairement le travail accompli ainsi que les contributions de ce mémoire, nous avons structuré ce document comme suit:

- Le chapitre 2 décrit les techniques de conception de systèmes de communication sans fil embarqués. nous donnons un aperçu à propos des plates-formes SDR et nous présentons la technique de conception basée sur un modèle ainsi que ses outils. Nous mettons l'accent sur les avantages de cette technique par rapport aux méthodes de conception traditionnelles.

- Le chapitre 3 détaille l'estimateur de l'étalement Doppler choisi ainsi que sa théorie. Nous présentons également notre station du travail.
- Le chapitre 4 commence par une présentation de l'architecture globale du design matériel de l'estimateur de l'étalement Doppler. Nous détaillons ensuite l'implémentation des différents blocs du design en se focalisant sur la technique de conception basée sur un modèle.
- Le chapitre 5 présente les différents résultats obtenus. Nous commençons par présenter le modèle de test que nous avons préparé au niveau de l'environnement Simulink. Ensuite, nous détaillons les résultats de co-simulation.
- Le chapitre 6 conclut ce document avec un résumé des travaux effectués ainsi que les résultats obtenus. Nous identifions également les travaux futurs afin d'améliorer la technique de design proposée.

## **Chapitre 2**

# **Prototypage rapide des systèmes de communication sans fil embarqués:**

# **Technique de « la conception basée sur un modèle » (MBD)**

### **2.1 Introduction**

Les nouvelles technologies sans fil sont maintenant en plein essor et jouent un rôle de plus en plus important dans notre mode de vie. L'intégration des systèmes embarqués dans le domaine de communications sans fil modernes a permis de répondre rapidement à ses demandes toujours croissantes. L'invention de FPGA a rendu possible le concept du matériel re-configurable pour la radio logicielle. En effet, les méthodes traditionnelles de conception de radios architecturées autour de la FPGA ne permettent pas une productivité élevée. Cette dernière peut être améliorée en utilisant une nouvelle technique de conception basée sur un modèle (MBD). Des outils basés sur cette technique sont ciblés pour le développement des systèmes embarqués, le développement des algorithmes de traitement du signal, l'intégration rapide du système et



l'analyse du comportement des systèmes numériques complexes pour une grande variété de cas d'utilisation. En conséquence, de nombreuses plates-formes matérielles et d'outils logiciels ont été développés pour tirer profit de cette technique afin de réaliser un prototypage rapide.

Dans ce chapitre, nous donnons un aperçu sur les plates-formes SDR et nous présentons la technique MBD ainsi que ses outils. Nous mettons l'accent sur les avantages de cette technique par rapport aux méthodes de conception traditionnelles pour le prototypage rapide de systèmes de communication sans fil embarqués.

## **2.2 La radio logicielle et la technique MBD**

### **2.2.1 La radio logicielle**

Le terme radio logicielle désigne les systèmes radios dont les composantes ont été réalisées par un logiciel [11]. Ces systèmes sont caractérisés par leur faisabilité technique, leur flexibilité et offrent plusieurs avantages sur le plan commercial. Généralement, la radio logicielle consiste en un système de communication sans fil dans lequel nous utilisons un logiciel spécialisé pour réaliser ses caractéristiques de transmission et de communication. Ce système est flexible puisque nous pouvons utiliser la même plateforme pour de multiples systèmes de communication. En outre, nous pouvons facilement et instantanément la re-configurer. Étant donné que ces radios n'ont pas les mêmes caractéristiques que les radios conventionnelles, elles doivent être construites différemment afin d'offrir une variété de fonctions telles que les multiples formes d'ondes dans le même matériel, la re-configurabilité en temps réel, etc. L'existence de telles fonctions sépare les fournisseurs de forme d'onde des fournisseurs de plates-formes [12].

La croissance rapide du marché des systèmes radios et l'existence de plusieurs normes ont forcé les fabricants à trouver une méthode rentable pour le déploiement de nouvelles normes sans fil dans les radios. Le concept de la radio logicielle est apparu pour la première fois au niveau du marché militaire, aux États-Unis, à travers la spécification Joint Tactical Radio System (JTRS) qui offre l'interopérabilité entre les radios utilisées par les soldats. Par la suite, ce concept a été adopté par le marché civil des systèmes radio ayant

connu une évolution des technologies telles que le GSM, GPRS, CDMA, UMTS, WiMAX, LTE, WiFi, et Bluetooth. Un seul appareil doit supporter toutes ces technologies simultanément qui ont leurs propres caractéristiques [12]. l'aspect re-configurable des radios logicielles peut faciliter la tâche d'amélioration des technologies sans fil ainsi que des techniques de correction de bugs. En raison de la flexibilité de ces systèmes, les fonctions de sécurité peuvent être améliorées en permanence pour lutter contre les menaces en constante évolution et garder la fonctionnalité radio sans risque.

### **2.2.2 La technique MBD**

La technique MBD pour les FPGAs résulte de la nécessité de concevoir des systèmes DSP complexes qui nécessitent des unités arithmétiques spécifiques comme l'unité addition-comparaison-sélection pour le décodeur de Viterbi. Ces unités de calcul spécifiques exigent un niveau plus fin d'optimisation des circuits basés sur FPGA. Ce niveau d'optimisation est généralement associée à la conception numérique traditionnelle des systèmes de communication. La conception basée sur un modèle est essentiellement une façon de décrire comment un système va interagir avec le monde analogique en temps réel.

Cette technique de conception consiste à convertir le modèle du système en question à partir de sa formulation mathématique à une spécification exécutable. Celle-ci contribue à la réalisation d'un modèle indépendant de plateforme (PIM) . Ce PIM est ensuite utilisé pour créer un modèle élaboré pour des matériels et logiciels spécifiques dans le but d'achever la conception du système à l'aide d'un outil de génération automatique de code. Ainsi, le PIM est converti en un modèle spécifique de plateforme (PSM) [13].

La technique MBD fournit un cadre commun qui implique différentes phases du processus de développement. Cela réduit les délais nécessaires pour créer un modèle complet. Les phases de conception associées à cette technique permettent au concepteur de localiser et de corriger les erreurs avant le prototypage du système.

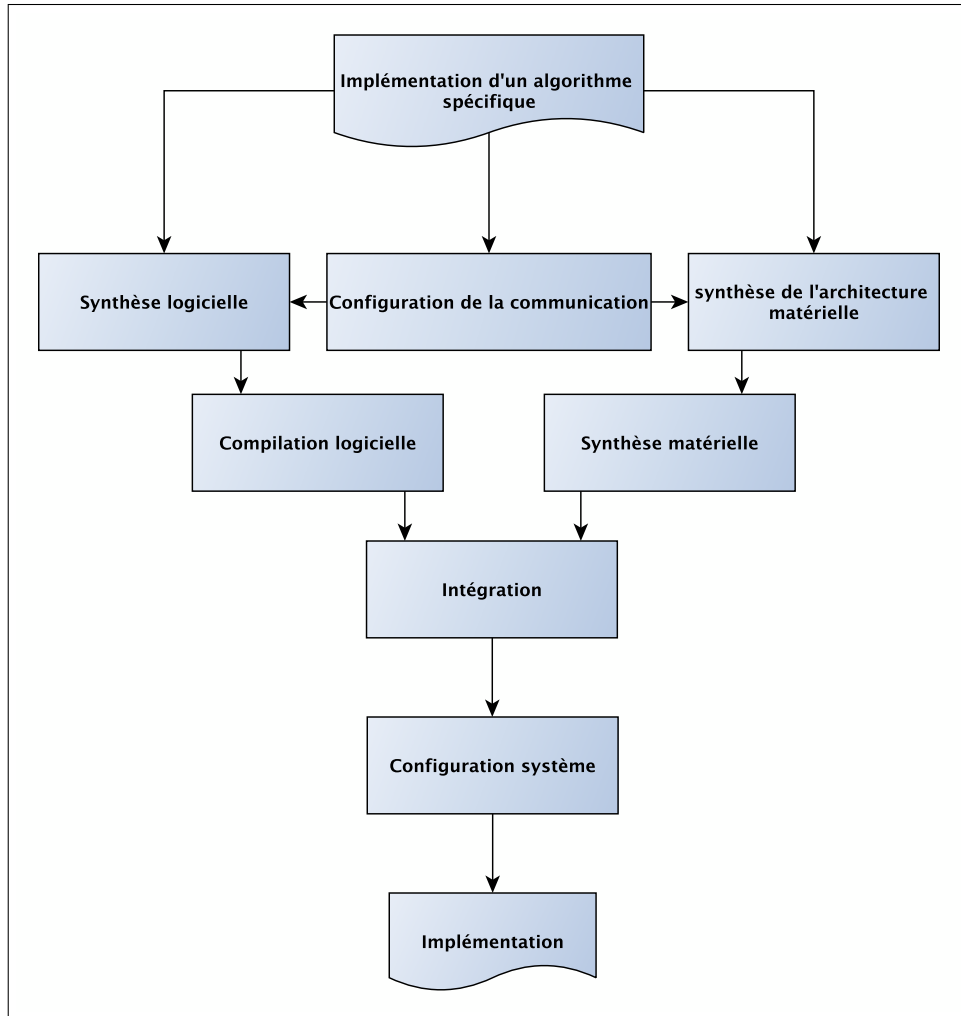


FIGURE 2.1 – Organigramme de la conception d'un système DSP générique.

### 2.2.3 Les outils de la conception basée sur un modèle

Il existe de nombreux outils disponibles pour la conception des systèmes FPGA en utilisant la technique MBD. La plupart de ces outils tirent profit du langage de modélisation unifié (UML) standardisé . Ces outils diffèrent dans leur façon de décrire un système et définir ses caractéristiques. Certaines techniques d'implémentation utilisées par ces outils peuvent être moins efficaces que d'autres. Cependant, elles garantissent le prototypage rapide du système assurant l'efficacité en termes de temps. Le choix d'un outil dépend de nombreux facteurs tels que le niveau de flexibilité, la disponibilité des bibliothèques/blocs pré-construits, et la compréhension globale de ces blocs. Parmi les outils basés sur UML, on trouve Real time Studio d'Arti-

san [14], Rhapsody d'I-logix [15], MATLAB et Simulink Realtime workshop [16]. Ces outils sont utilisés pour la conception d'un environnement multiprocesseur embarqué.

Les outils qui permettent la génération du code HDL pour le FPGA peuvent être classés selon deux catégories. En effet, on perçoit les outils à base de blocs et ceux basés sur le langage C. Les outils à base de blocs génèrent le code HDL à partir du diagramme en bloc, qui est ensuite utilisé par l'outil de synthèse matérielle pour implémenter le design du système sur FPGA. La plupart de ces outils sont basées sur les deux environnements Simulink et MATLAB tels que Synplify DSP [17], Xilinx System Generator, DSP Builder d'Altera [18], et Simulink HDL Coder [19]. Ces outils garantissent un environnement de modélisation de haut niveau des algorithmes de traitement du signal. Les blocs de la bibliothèque Simulink sont utilisés avec les IP cores des fournisseurs de FPGA pour créer un code HDL spécifique à la plateforme en question. Les outils tels que Simulink HDL Coder et Synplify DSP permettent plus de flexibilité pour le concepteur, car ils intègrent des fonctions MATLAB et des fichiers m-block. Avec ces outils, le concepteur développe un modèle Simulink, puis le traduit sous l'environnement de FPGA dans une séquence d'étapes. La deuxième catégorie des outils MBD exploite le langage de programmation C pour créer une abstraction pour la conception des systèmes avec FPGA. Parmi ces outils, on trouve Mentor Graphics Catapult C [20] et Celoxica's Handel-C [21]. La principale motivation derrière ces outils est que le langage C, étant couramment utilisé pour l'implémentation des algorithmes DSP, offre un gain de productivité par rapport aux codes Verilog/VHDL.

Maintenant, nous allons décrire quelques outils MBD qui sont basés sur les environnements Simulink et MATLAB.

### **Simulink HDL coder**

Simulink est un outil MBD qui permet la modélisation, l'analyse et la simulation des systèmes. Il fournit un environnement graphique bien structuré pour le concepteur qui lui permet de créer des designs de haut niveau des systèmes complexes en utilisant des blocs. En outre, cet outil permet à l'utilisateur de créer des blocs personnalisés flexibles à partir des fonctions MATLAB.

Simulink HDL coder est une fonctionnalité de Simulink qui permet au concepteur de créer un code HDL précis au bit et synthétisable à partir du modèle développé en utilisant les blocs Simulink. Le code HDL obtenu peut être synthétisé et mappé sur la carte FPGA cible en utilisant des outils tels que Altera Quartus II, Synplify, et Xilinx ISE.

Simulink HDL coder a de nombreuses bibliothèques construites pour assurer la génération du code HDL. Certaines de ces bibliothèques prédéfinies comprennent des additionneurs, des multiplicateurs, des accumulateurs, des intégrateurs, des commutateurs multi-ports, des lookup tables, etc.

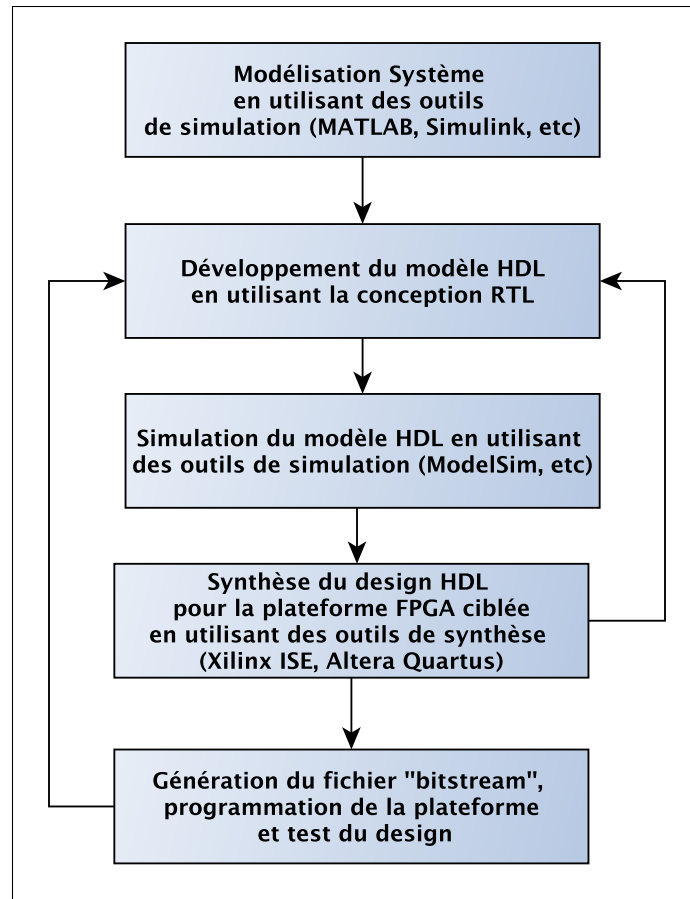
### **Xilinx System Generator for DSP**

XSG est un outil de modélisation au niveau dédié pour les architectures Xilinx de FPGA. Cet outil exploite les capacités offertes par l'environnement Simulink pour fournir un plus haut niveau d'abstraction du système. Il permet de créer automatiquement un code FPGA de bas niveau pour les plateformes Xilinx. Cet outil utilise des DSP IP cores de Xilinx ainsi que des blocs des fonctionnalités communes. Sa différence majeure de l'outil Simulink HDL coder est la génération du code de bas niveau qui cible les plateformes Xilinx telle que Virtex. Cet outil permet également la co-simulation matérielle [10] utilisée pour tester les cores créés par l'utilisateur au niveau du matériel ciblé simultanément avec le modèle présent au niveau de l'environnement Simulink .

## **2.3 Le flot de conception de systèmes basée sur FPGA**

### **2.3.1 Le flot de conception traditionnel**

La figure 2.2 montre que la méthode traditionnelle de la conception d'un système basé sur FPGA commence par la phase de spécifications suivie par celle de la simulation/modélisation de ce système selon ces spécifications. Le design obtenu du système est ensuite traduit en une description hardware de haut niveau en utilisant un langage HDL tel que VHDL. Malgré l'importance de cette phase qui permet la description du comportement réel du système à concevoir, elle nécessite beaucoup de temps. Finalement, le code HDL



**FIGURE 2.2 – Flot de conception traditionnel.**

obtenu est synthétisé, implémenté, validé et testé. L'absence de la validation durant la phase de simulation/modélisation résulte en un temps de développement plus élevé. La détermination des erreurs, en termes de temps et de fonctionnalité pendant ou après l'étape d'implémentation, résulte en un effort énorme pour réitérer l'ensemble du processus de conception.

Le flot de conception MBD vient comme une solution pour faire face aux inconvénients de la méthode traditionnelle. Ce flot de conception est de plus en plus adopté car il intègre un système de simulation/modélisation et de validation en une seule étape.

### **2.3.2 Le flot de conception MBD**

Un flot de conception MBD typique pour l'implémentation des systèmes sans fil sur FPGA est représenté dans la figure 2.3

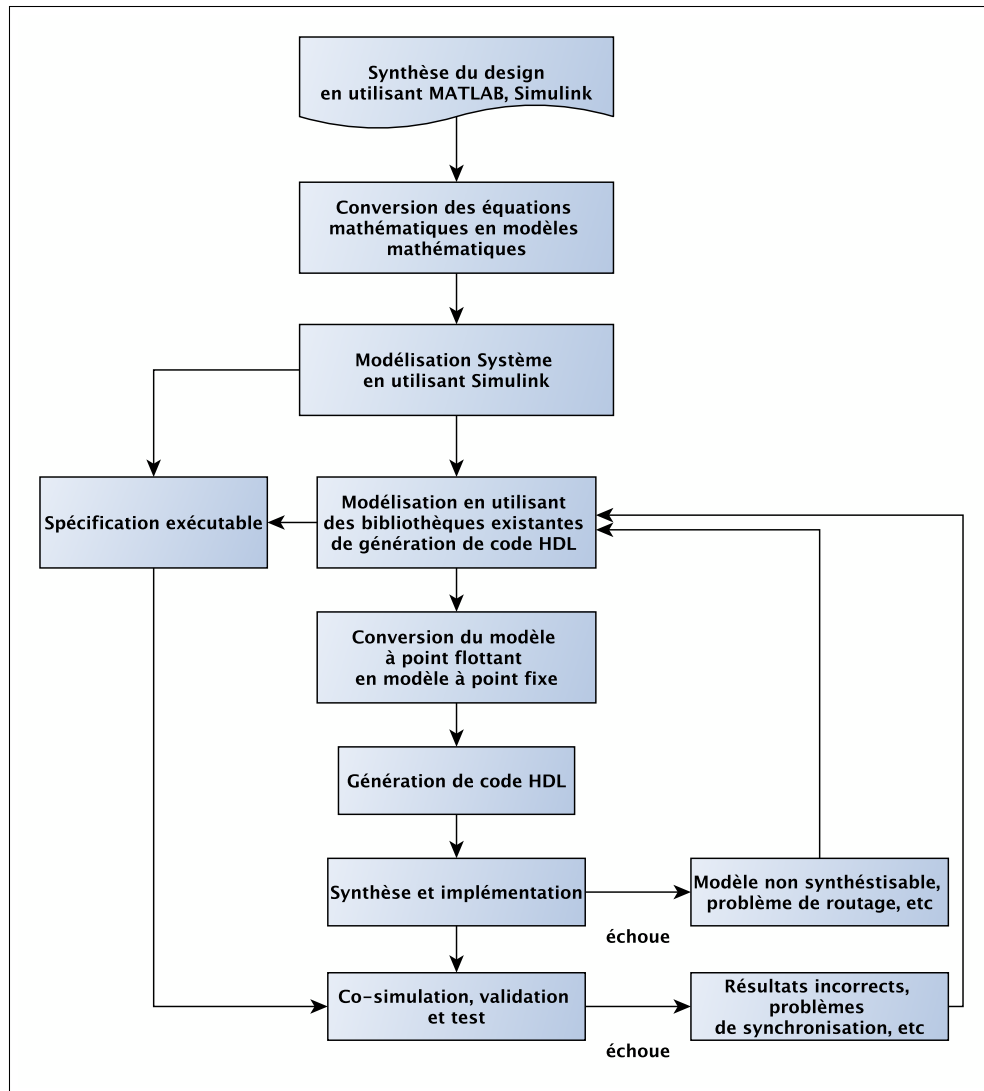


FIGURE 2.3 – Flot de conception MBD pour FPGA.

### Les spécifications de conception

La conception de tout bloc DSP commence toujours par définir les spécifications de conception. Cette définition représente une manière abstraite pour décrire la fonction principale du bloc. À la fin de cette étape, on obtient différents paramètres tels que le type d'entrée/sortie et la fonction mathématique de base du bloc. Nous pouvons toujours déterminer si la conception du bloc peut être réutilisée ou non.

## **L'analyse des besoins de conception**

Comme mentionné, la fonction mathématique de base du bloc est déterminée dans la phase des spécifications. La prochaine étape dans le processus de conception est d'identifier l'algorithme requis pour l'implémentation pratique. Cet algorithme est choisi en fonction de nombreux facteurs tels que la complexité, la durée du calcul, et les ressources utilisées. Cet algorithme est ensuite subdivisé en fonctions.

Une fois que nous identifions les algorithmes des différentes fonctionnalités utilisées dans le design, nous analysons l'effet des opérations en virgule fixe sur l'algorithme choisi. Cette analyse permet au concepteur de déterminer la longueur optimale du mot binaire, le format en virgule fixe optimal, et la mise à l'échelle nécessaire à chaque étape de l'algorithme.

## **Simulation et modélisation**

Dans cette étape, nous convertissons Les spécifications de conception dans un modèle de haut niveau en utilisant les outils MBD tels que Simulink HDL Coder et XSG. Cela consiste à représenter les équations mathématiques sous la forme de modèles mathématiques. Les bibliothèques et les blocs fournis par l'outil facilitent cette représentation. Le modèle identifié dans cette étape peut être utilisé comme un modèle PIM, car il peut être ciblé par des plateformes multiples.

Le modèle obtenu est ensuite testé en observant la sortie pour les différents signaux d'entrée de test. En dépit de la grande variété de fonctionnalités et des blocs fournis par Simulink, seulement certains blocs peuvent être convertis à l'aide de son environnement de conversion HDL ce qui signifie que les autres blocs doivent être remplacés ou construits en utilisant les bibliothèques primitives de l'outil de conversion HDL tel que VHDL.

## **L'implémentation du système**

Cette étape est la dernière aussi la plus importante. Elle consiste à convertir le modèle PIM en un modèle PSM afin d'obtenir une implémentation du modèle exécutable. Un mécanisme de génération automatique



du code est nécessaire dans cette étape. Ce mécanisme est fourni par les outils MBD. L'outil flow advisor fourni par Simulink HDL Coder, aide à vérifier la compatibilité des modèles pour la génération du code, la conversion du modèle en virgule flottante en un modèle en point fixe, réglage de l'horloge, les types de données d'entrée/sortie dans le modèle en point fixe, et la mise à l'échelle de données.

Lorsque le code HDL est généré, il peut être synthétisé au niveau de l'outil Simulink HDL Coder. Ce dernier prend en charge les outils de simulation comme Modelsim [22] et Cadence Incisive [23]. Il prend également en charge les outils de synthèse de Xilinx et Altera.

Pour les FPGAs fournis par Xilinx, nous pouvons utiliser l'outil XSG qui génère des IP cores ciblant cette famille de FPGAs. Il exploite à la fois les modèles Simulink de haut niveau et l'ensemble des blocs Xilinx qui fournissent des fonctions de traitement du signal, des mémoires, et de la logique numérique. Le type de données nécessaire pour le modèle développé doit être identifié durant l'utilisation de l'outil XSG.

## **2.4 Conclusion**

Dans ce chapitre, nous avons donné un aperçu à propos de la radio logicielle ainsi qu'à propos de la technique MBD et ses outils. Nous avons également souligné les avantages de cette technique par rapport aux méthodes traditionnelles de conception des systèmes radio basés sur FPGA. Dans le chapitre suivant, nous allons présenter l'estimateur de l'étalement Doppler à implémenter ainsi que notre station du travail.

## **Chapitre 3**

# **Description de l'estimateur de l'étalement Doppler et présentation de notre station du travail**

### **3.1 Introduction**

Dans les systèmes de communication sans fil, le canal est caractérisé par son aspect d'évanouissement variant dans le temps et en fréquence qui fait que le signal reçu ainsi que sa phase varient dans le temps d'une manière aléatoire. Étant donné que cette variation est liée à l'étalement Doppler, la connaissance de ce dernier représente un paramètre clé pour de nombreuses applications de communication sans fil. Dans ce travail, on considère un canal à évanouissement plat (non sélectif en fréquence) et on s'intéresse à l'effet Doppler. En effet, Plusieurs estimateurs de l'étalement Doppler ont été proposés dans la littérature. Cependant, c'est l'estimateur qui montre le compromis le plus attractif entre la complexité du calcul et la précision de l'estimation qui est habituellement considéré comme le meilleur candidat pour l'implémentation matérielle dans la pratique.

Les nouvelles technologies sans fil sont maintenant en plein essor. L'intégration des systèmes embarqués dans le domaine des communications sans fil modernes a permis de répondre rapidement à ses demandes toujours croissantes. En conséquence, de nombreuses plates-formes matérielles et des outils logiciels ont été développés.

Dans ce chapitre, nous allons commencer par décrire l'estimateur de l'étalement Doppler choisi ainsi que sa théorie. Ensuite, nous allons présenter la station de travail utilisée dans la phase d'implémentation.

## 3.2 Description de l'estimateur de l'étalement Doppler choisi

Dans la pratique, les estimés de l'étalement Doppler sont obtenus à partir des estimés des coefficients du canal. Ainsi, quatre classes d'estimateurs Doppler ont été proposées dans la littérature selon la méthode d'estimation des coefficients du canal utilisée:

- Level-crossing rate (LCR)-based [24, 25].
- Covariance-based [26, 27].
- Spectrum-based [28].
- Maximum likelihood (ML) techniques [29].

L'estimateur Doppler que nous avons choisi pour l'implémentation est basé sur la dernière méthode d'estimation. Nos motivations derrière le choix de l'estimateur proposé dans [6] sont justifiées par son compromis attractif entre la complexité du calcul (nombre d'opérations) et la précision de l'estimation. En fait, étant dérivé de la théorie ML, il a été prouvé dans [6] que cet estimateur surpasse de loin les différentes techniques proposées dans la littérature en termes de précision de l'estimation comme le montre la figure 3.1

En outre, il est basé sur le modèle d'approximation de deux raies fréquentielles pour la matrice de covariance du canal. Par conséquent, l'inverse de la matrice de covariance du signal reçu est déterminé analytiquement. Le plus intéressant est que ce modèle d'approximation est valable pour différents types de la densité spectrale de puissance du canal (PSD) dont la connaissance au préalable n'est plus nécessaire pour cet

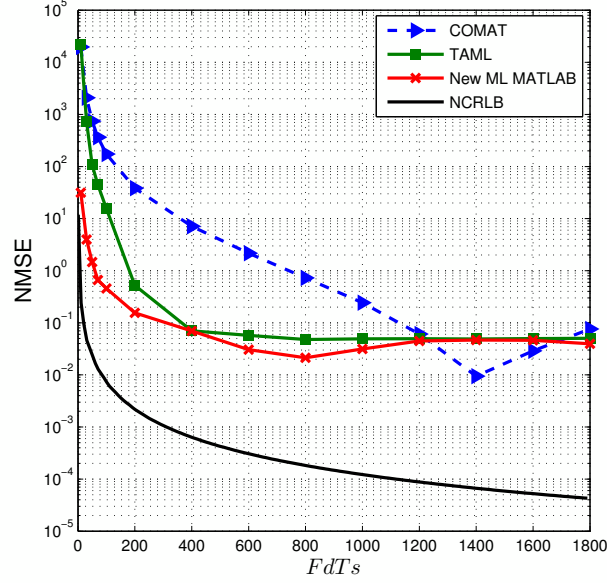


FIGURE 3.1 – NMSE vs.  $F_d T_s$  avec  $T_s = 10 \mu s$ ,  $SNR = 0 \text{ dB}$ , et  $N = 1000$ .

estimateur contrairement à toutes les techniques existantes. Ce modèle rend cet estimateur un très bon candidat pour l'implémentation matérielle puisque la PSD du canal est généralement inconnue dans la pratique et change souvent d'un environnement à un autre.

### 3.2.1 Modèle du système

Nous considérons un signal  $x(t)$  qui se propage à travers un canal de Rayleigh à évanouissement plat  $h(t)$ . Ce signal est immergé dans un bruit additif  $w(t)$ , qui peut également représenter n'importe quel signal d'interférence. L'expression du signal reçu est donnée par:

$$y(t) = h(t).x(t) + w(t) \quad (3.1)$$

Soit  $n_p T_s$  les positions des symboles pilotes, avec  $T_s$  la période symbole. Le signal reçu en ces positions représente les estimés brutes des coefficients du canal. De même, le signal reçu peut être échantillonné à la cadence  $\frac{1}{T_s}$  pour obtenir  $N$  échantillons permettant le suivi et l'estimation du canal pour les deux cas des symboles pilotes et non pilotes  $nT_s$ .

Nous supposons que  $N$  coefficients équadistants du canal sont fournis au récepteur par n'importe quelle technique d'estimation du canal comme suit:

$$\hat{h}(nT_s) = h(nT_s) + w(nT_s), n = 0..N - 1 \quad (3.2)$$

avec  $w(nT_s)$  désigne l'erreur d'estimation modélisée par un bruit blanc gaussien circulaire centré ayant une variance  $\sigma_n^2$ . Les statistiques des coefficients du canal sont régies par l'étalement Doppler inconnu obtenu à partir des estimés  $\{\hat{h}(nT_s)\}_{n=0}^{N-1}$ .

Nous optons pour une représentation vectorielle de l'équation ci-dessus comme suit:

$$\hat{\mathbf{h}} = \mathbf{h} + \mathbf{w} \quad (3.3)$$

avec

$$\begin{cases} \hat{\mathbf{h}} &= [\hat{h}(0), \dots, \hat{h}((N-1)T_s)]^T \\ \mathbf{h} &= [h(0), \dots, h((N-1)T_s)]^T \\ \mathbf{w} &= [w(0), \dots, w((N-1)T_s)]^T \end{cases} \quad (3.4)$$

### 3.2.2 Formulation de l'estimateur de l'étalement Doppler

Selon [6], L'étalement Doppler estimé localement  $\hat{\sigma}_D(m)$  obtenu à partir de la  $m^{\text{ème}}$  fenêtre d'approximation est donné par:

$$\hat{\sigma}_D(m) = \underset{\sigma_D}{\operatorname{argmax}} \mathcal{L}(\hat{\mathbf{h}}(m), \sigma_D) \quad (3.5)$$

avec  $\mathcal{L}(\hat{\mathbf{h}}(m), \sigma_D)$  la fonction de logvraisemblance (LLF) de  $\hat{\mathbf{h}}(m)$  paramétrée par  $\sigma_D$ . Tous les détails relatifs à la formulation de l'estimateur conduisant à l'expression finale du LLF sont donnés dans [6]:

$$\mathcal{L}(\hat{\mathbf{h}}(m), \sigma_D) = \log([2 + \hat{\rho}\sigma_1][2 + \hat{\rho}\sigma_2]) + \frac{1}{\hat{\sigma}_n^2} \sum_{i=1}^2 \sqrt{\frac{\hat{\rho}\sigma_i}{2 + \hat{\rho}\sigma_i}} \left| \mathbf{u}_i^H \hat{\mathbf{h}}(m) \right|^2 \quad (3.6)$$

avec  $\{\cdot\}^H$  est l'opérateur de trans-conjugaison. En outre,  $\hat{\sigma}_n^2$  et  $\hat{\rho}$  sont, respectivement, l'estimé de la variance du bruit et le  $SNR$  qui sont obtenus en utilisant la technique proposée dans [30]. En utilisant des fenêtres d'approximation de même taille,  $N$ , les expressions de deux valeurs propres  $\sigma_1$  and  $\sigma_2$  sont données par:

$$\sigma_1 = N + \frac{\sin(N\sigma_D T_s)}{\sin(\sigma_D T_s)} \quad \text{and} \quad \sigma_2 = N - \frac{\sin(N\sigma_D T_s)}{\sin(\sigma_D T_s)} \quad (3.7)$$

Leurs vecteurs propres associés  $\mathbf{u}_1$  and  $\mathbf{u}_2$  sont obtenus à partir des expressions suivantes:

$$\begin{cases} \mathbf{u}_1 &= \frac{\sqrt{2}}{\sqrt{\sigma_1}} e^{-j\frac{N-1}{2}\sigma_D T_s} \tilde{\mathbf{u}}_1 \\ \mathbf{u}_2 &= j\frac{\sqrt{2}}{\sqrt{\sigma_2}} e^{-j\frac{N-1}{2}\sigma_D T_s} \tilde{\mathbf{u}}_2 \end{cases} \quad (3.8)$$

avec

$$\tilde{\mathbf{u}}_1 = \begin{bmatrix} \cos(\frac{N-1}{2}\sigma_D T_s) \\ \cos(\frac{N-3}{2}\sigma_D T_s) \\ \vdots \\ \cos(\frac{N-(2k+1)}{2}\sigma_D T_s) \\ \vdots \\ \vdots \\ \cos(\frac{N-(2k+1)}{2}\sigma_D T_s) \\ \vdots \\ \cos(\frac{N-3}{2}\sigma_D T_s) \\ \cos(\frac{N-1}{2}\sigma_D T_s) \end{bmatrix} \quad \tilde{\mathbf{u}}_2 = \begin{bmatrix} \sin(\frac{N-1}{2}\sigma_D T_s) \\ \sin(\frac{N-3}{2}\sigma_D T_s) \\ \vdots \\ \sin(\frac{N-(2k+1)}{2}\sigma_D T_s) \\ \vdots \\ \vdots \\ -\sin(\frac{N-(2k+1)}{2}\sigma_D T_s) \\ \vdots \\ -\sin(\frac{N-3}{2}\sigma_D T_s) \\ -\sin(\frac{N-1}{2}\sigma_D T_s) \end{bmatrix}. \quad (3.9)$$

Afin d'améliorer la précision de l'estimation, nous calculons la moyenne des estimées de l'étalement Doppler obtenues pour chaque fenêtre d'approximation locale (de taille  $N$ ) comme suit:

$$\tilde{\sigma}_D = \frac{1}{M} \sum_{m=1}^M \hat{\sigma}_D(m) \quad (3.10)$$

avec  $M \times N$  est la taille de la fenêtre d'approximation globale.

### 3.3 Présentation de notre station du travail

#### 3.3.1 La plateforme de la radio-logicielle: PicoSDR 2x2

La plateforme PicoSDR 2x2 [31, 32], fournie par la compagnie Nutaq, est une plateforme de radio-logicielle qui intègre deux modules émetteur-récepteur RF multimode, une carte FPGA (Virtex-6) puissante, et un module de mémoire. Cette plateforme est une solution pour toute la chaîne de transmission du traitement numérique en bande de base jusqu'à l'interface radio. Combinée avec le framework logiciel de Nutaq, la PicoSDR permet d'optimiser le temps de mise sur le marché d'un produit, tout en offrant les performances requises à une variété d'applications sans fil telles que les radios cognitives, la radio-logicielle multimode, les systèmes MIMO , etc.



FIGURE 3.2 – La plateforme PicoSDR 2x2 [32].

#### Architecture matérielle de la plateforme PicoSDR 2x2

La plateforme PicoSDR 2x2 est composée d'une carte porteuse "Perseus601X" [32, 33] équipée d'une carte FPGA Virtex-6, d'une carte mezzanine de FPGA (FMC) "Radio420m" [32, 34], et d'une carte mère " $\mu$ TCA" [32].

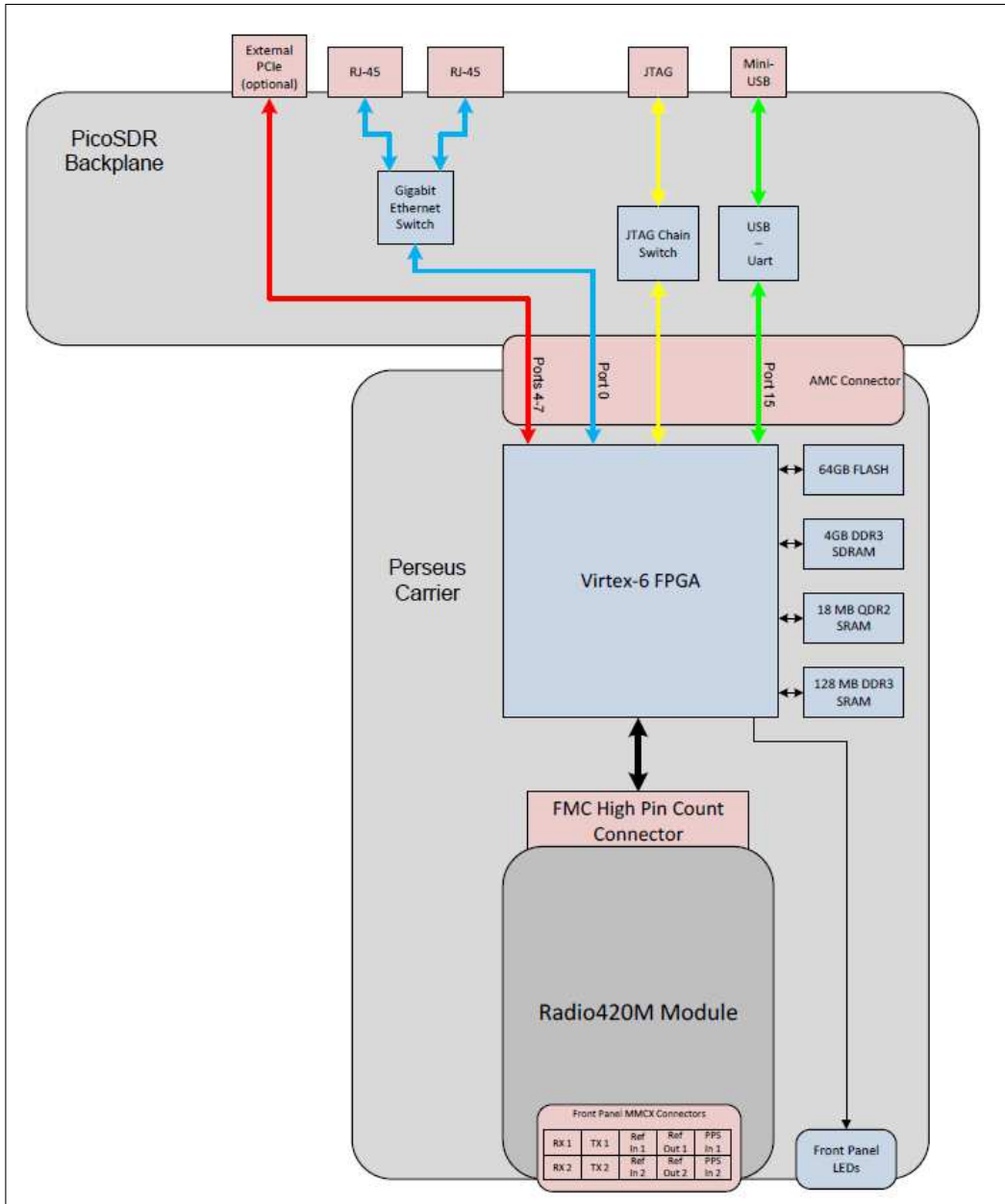


FIGURE 3.3 – Diagramme de l’architecture matérielle de la plateforme PicoSDR 2x2 [32].

- La carte mère “ $\mu$ TCA”:

Elle fournit l’alimentation électrique nécessaire au bon fonctionnement de la plateforme. Elle contient également les éléments suivants:

- Un commutateur Ethernet: qui représente l’interface de communication entre la “Perseus” et les connecteurs RJ-45.



- Un connecteur JTAG: qui permet à l'utilisateur d'assurer la programmation de la carte FPGA intégrée dans la "Perseus".
  - Un port USB: qui permet à l'utilisateur de se connecter soit au menu console de la palteforme, soit à la console Linux de la "Perseus".
  - Un connecteur AMC: qui contrôle tous les liens entre la "Perseus" et la carte mère " $\mu$ TCA".
- La carte porteuse "Perseus601X":

La carte mezzanine avancée "Perseus601X" est conçue autour de la carte FPGA Virtex-6. Elle est conçue également pour les applications de hautes performances qui nécessitent une large bande passante à faible latence de traitement [32, 33].

- La FMC "Radio420m":

Cette carte mezzanine de FPGA est un module émetteur-récepteur RF radio multimode très puissant. Elle est conçue autour du circuit intégré d'émetteur-recepteur RF programmable "Lime Microsystems LMS6002D" [35]. Cette carte supporte une couverture large bande. Elle supporte également le mode duplex par séparation temporelle (TDD) ainsi que le mode duplex par séparation fréquentielle (FDD). Cette carte offre une largeur de bande sélectionnable (1.5 jusqu'à 28 MHz) afin de garantir une meilleure sélectivité du canal pour plusieurs applications large bande et à bande étroite [32, 34].

## Architecture logicielle de la plateforme PicoSDR 2x2

- a. Board Support Package (BSP):

le "Board Support Package" fourni par Nutaq sert comme un design de référence qui inclut des IP cores, des interfaces VHDL d'entrée/sortie, et des APIs permettant aux concepteurs de développer leurs applications avec les matériels du Nutaq. Ce BSP réduit le temps d'exécution de certaines tâches telle que l'assignement des pins d'entrée/sortie du FPGA [36].

Le BSP permet également de contrôler les interfaces d'entrée/sortie de la carte FMC en utilisant des bibliothèques de fonction conçues pour ce type de tâches. Ces bibliothèques sont développées spécialement pour le processeur logique MicroBlaze de Xilinx. Ainsi, le kit de développement logiciel (SDK) de Xilinx permet à l'utilisateur de créer ses propres applications [37].

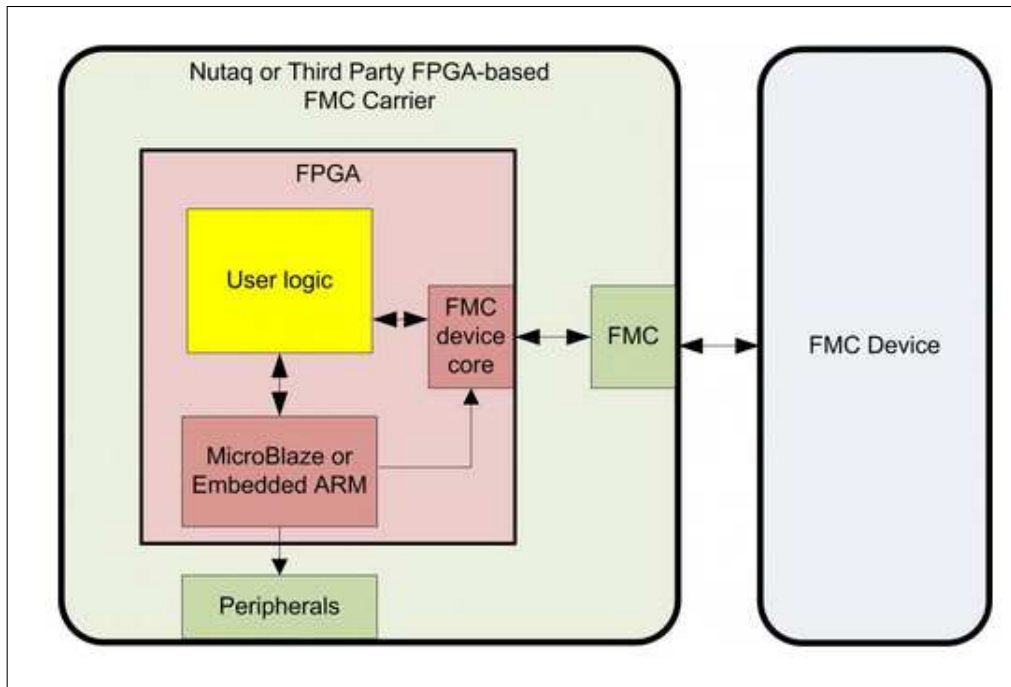


FIGURE 3.4 – Diagramme de BPS du Nutaq [36].

**b. Board Software Development Kit (BSDK):**

Le “Board Software Development Kit”, fourni par Nutaq, inclut des IP cores, des APIs, et des interfaces graphiques (GUI) qui réduisent le temps d’exécution des tâches à faible valeur ajoutée telles que la programmation des interfaces et la configurations des contraintes liées au FPGA. The BSDK inclut une couche transport (CCE) qui permet le contrôle à distance ainsi que l’échange des données entre le FPGA et un ordinateur distant [38].

**c. Model-Based Design Kit (MBDK):**

Le “Model-Based Design Kit” est un ensemble d’outils développés au-dessus de BSDK. Ces outils permettent aux concepteurs de modéliser, simuler, tester, débogger et déployer leurs applications à partir de l’environnement graphique de Simulink [39].

Le MBDK tire profit également de l’outil XSG permettant de développer des systèmes basés sur FPGA sans avoir besoin de coder en VHDL ce qui réduit d’une manière significative le délai de commercialisation. En outre, le MBDK offre la possibilité d’exécuter une co-simulation sur un ordinateur hôte assurant un échange des données en temps réel entre Simulink et la plateforme PicoSDR.

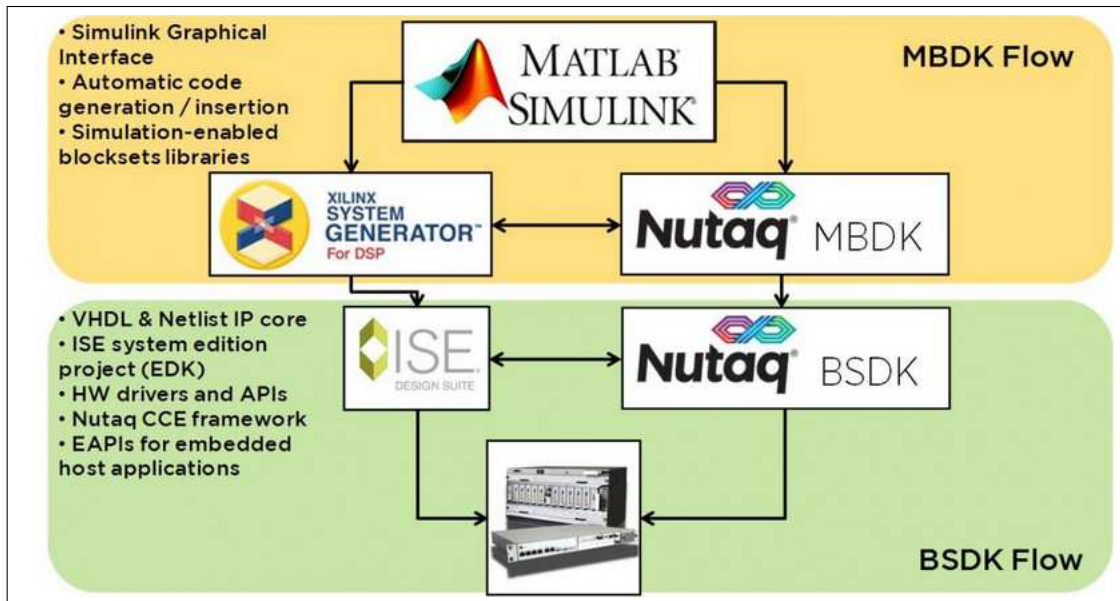


FIGURE 3.5 – Nutaq MBDK & BSDK flows [39].

- Les outils de développement FPGA: Ces outils fournis par le MBDK exploitent les différentes bibliothèques offertes par XSG pour permettre aux concepteurs de développer leurs propres algorithmes sur FPGA. Combinés avec la bibliothèque du Nutaq, les concepteurs peuvent mapper leurs algorithmes aux interfaces d'entrée/sortie tels que les convertisseurs analogique-numérique (ADC) et numérique-analogique (DAC), les modules de mémoire, ou les interfaces PCIe/GigE. La bibliothèque des blocs Nutaq permet la synthèse du design entier et offre la possibilité d'interagir avec les blocs Simulink en phase de simulation.
- Les outils de co-simulation:
 

Ces outils assurent un échange de données en temps réel entre Simulink et la plateforme PicoSDR en utilisant l'interface RTDEx (Real-Time Data Exchange). ils incluent:

  - L'ensemble des blocs RTDEx.
  - L'ensemble des blocs du contrôle des modules d'entrée/sortie de la carte FMC.
  - L'ensemble des blocs des registres du contrôle personnalisés.

### **3.3.2 L'émulateur de canal radio**

#### **L'émulateur "EB Prosim"**

L'environnement sans-fil impose plusieurs contraintes et limitations affectant les performances des systèmes de communications sans-fil. Les caractéristiques de propagation du canal radio tels que l'atténuation, l'effet de masque, l'évanouissement, l'effet Doppler, le bruit et les interférences provoquent la dégradation du lien de transmission. Ainsi, le déploiement des systèmes de communication sans-fil exige l'utilisation d'une méthodologie de test rigoureuse afin d'assurer le bon fonctionnement des différents produits de communication sans-fil même dans les environnements de propagation les plus exigeants. La technique traditionnelle d'expérimentation sur terrain est prenante, coûteuse, et nécessite un travail intensif. Ainsi, une nouvelle méthodologie de test est nécessaire tels que les émulateurs du canal radio parmi lesquels on trouve l'émulateur "EB Prosim". Ce dernier permet de recréer les effets de la propagation d'un canal radio dans un environnement de laboratoire contrôlé. Il permet d'émuler d'une manière réaliste, et avec une grande précision, les effets typiques de propagation radio tels que la propagation multi-trajets, l'évanouissement rapide, l'atténuation, le bruit, les interférences ou encore l'effet de masque. L'émulateur "EB Prosim" supporte une largeur de bande très importante et une plage dynamique élevée. En outre, les caractéristiques physiques du canal radio peuvent être émulées d'une manière indépendante. L'émulation peut co-exister avec la technique d'expérimentation sur terrain ou bien la remplacer. La figure 3.6 montre les trois modèles de l'émulateur "EB Prosim" fournis par la compagnie Anite telecoms. Dans le présent travail, nous avons utilisé le modèle "EB Prosim FS8" [40].

#### **Le concept d'émulation du canal radio**

"EB Prosim" est un émulateur générique du canal radio qui permet de recréer l'environnement de propagation indépendamment de la technologie du système de communication sans-fil en question. Cet émulateur supporte également presque tous les standards de communication sans-fil ainsi que tout type de signal pour une large gamme de fréquences. Pour un scénario de test typique, l'émetteur et le récepteur sont connectés



FIGURE 3.6 – Les émulateurs EB PropSim F32, EB PropSim FS8, et EB PropSim F8 [40].

à l'émulateur "EB PropSim" tel qu'il est illustré dans la figure 3.7. L'émulateur utilise les signaux physiques générés par l'équipement sous-test comme un signal d'entrée.

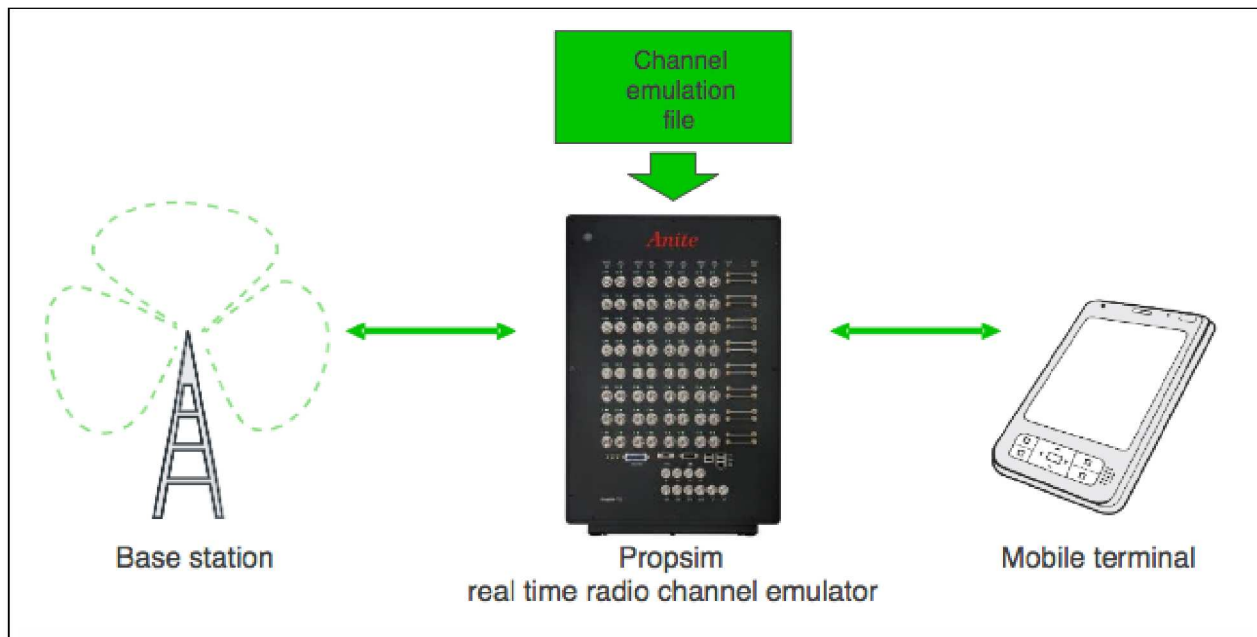


FIGURE 3.7 – Schéma de remplacement du canal radio par "EB PropSim" [40].

L'émulateur "EB Prosim" permet d'effectuer plusieurs types de tests et d'émulations grâce à sa flexibilité ainsi qu'à la multitude des options de configuration qu'il fournit. Les émulations sont basées sur des fichiers d'émulation pré-configurée stockés dans la mémoire de l'émulateur. De nouveaux fichiers d'émulation peuvent être ajoutés par l'utilisateur. Cette approche d'émulation basée sur un fichier permet d'assurer la répétabilité et la contrôlabilité des émulations. Pour une utilisation basique, l'utilisateur a seulement besoin de charger l'émulation et d'ajuster les niveaux de puissance de l'émulateur. Dans le cas d'une utilisation avancée, l'utilisateur peut créer des nouveaux modèles du canal ou bien modifier les modèles déjà existants.

L'émulateur "EB Prosim" vient avec une interface graphique facile à utiliser qui permet une configuration rapide et facile de l'émulation en question.

### **3.4 Conclusion**

Dans ce chapitre, nous avons décrit l'estimateur de l'étalement Doppler choisi ainsi que sa théorie. Nous avons également présenté une description de la station du travail que nous utiliserons dans la phase d'implémentation composée essentiellement de la plateforme PicoSDR et de l'émulateur de canal "EB Prosim FS8".

## **Chapitre 4**

# **Architecture globale et implémentation de l'estimateur de l'étalement Doppler**

### **4.1 Introduction**

La technique MBD pour les FPGAs résulte de la nécessité de concevoir des systèmes DSP complexes qui nécessitent des unités arithmétiques spécifiques. Cette technique de conception consiste à convertir le modèle du système en question à partir de sa formulation mathématique en une spécification exécutable.

Dans ce chapitre, nous allons présenter en premier temps l'architecture globale du design matériel de l'estimateur de l'étalement Doppler. Nous détaillerons par la suite l'implémentation des différents blocs du design en se basant sur la technique MBD.

### **4.2 Architecture globale du design**

La figure 4.1 montre un diagramme en bloc de haut-niveau de l'architecture globale de notre design. En effet, un système SISO simple est déployé. La chaîne d'émission est constituée d'un générateur des données IQ qui génère le composant en phase et celui en quadrature du signal à transmettre. Ce générateur

fournit, en sortie, les données IQ entrelacées ainsi qu'un signal de sélection. L'émetteur "Radio420m" effectue la conversion numérique-analogique ainsi que la transposition de la fréquence du signal en bande de base afin d'obtenir un signal RF analogique. L'élément suivant dans notre diagramme est l'émulateur du canal radio "EB Prosim F8". Ce dernier permet d'émuler un environnement de propagation avec une grande précision en prenant comme entrée le signal émis. Le récepteur "Radio420m" permet de récupérer la version numérique en bande de base du signal reçu. Ensuite, on sépare les deux composants du signal reçu à l'aide du bloc démultiplexeur IQ. Ces deux composants représentent les entrées du bloc d'estimation de l'étalement Doppler qui constitue le cœur de notre présent travail.

Des registres de contrôle sont utilisés pour la saisie des paramètres globaux du design. En outre, les différents blocs sont implémentés en utilisant les bibliothèques de l'outil XSG et le MBDK du Nutaq.

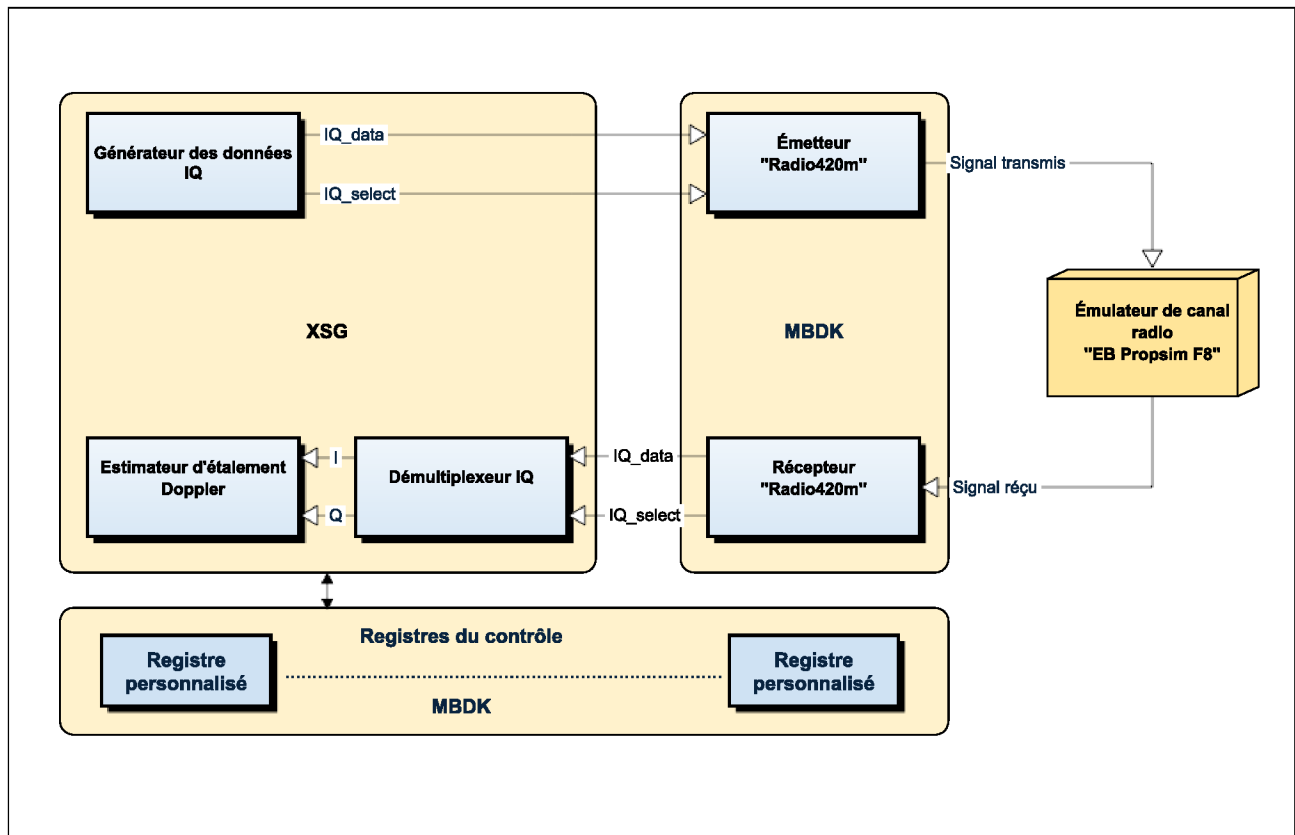


FIGURE 4.1 – Diagramme en bloc du design.



### 4.3 Implémentation de l'estimateur de l'étalement Doppler

La figure 4.2 montre un diagramme en bloc de haut-niveau du design de l'estimateur de l'étalement Doppler. En effet, ce module est constitué de deux chemins différents: un chemin de données qui englobe les unités du calcul et du stockage des données et un chemin de contrôle qui commande ces unités et synchronise entre elles.

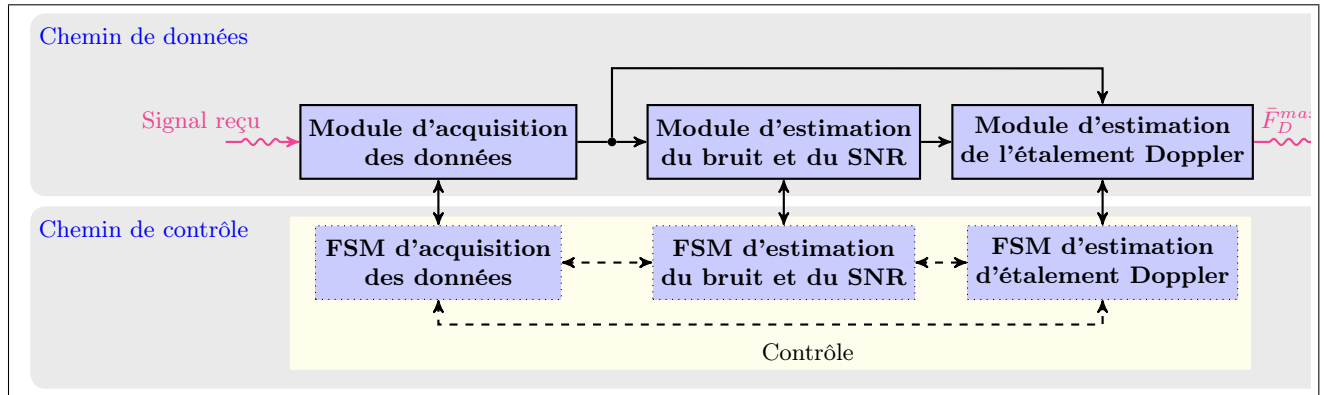


FIGURE 4.2 – Diagramme en bloc du module d'estimation de l'étalement Doppler.

#### 4.3.1 Chemin des données

##### Module d'acquisition des données

Ce module permet de stocker 100 échantillons du signal reçu dans des modules de mémoire. la cadence d'échantillonnage  $T_s$  est spécifiée par l'utilisateur et peut être modifiée selon le standard de communication sans-fil utilisé. Ces échantillons sont utilisés dans différents étages de calcul. Ainsi, nous avons procédé à leur copie dans différents modules de mémoire afin de pouvoir paralléliser l'exécution de certaines tâches .

##### Module d'estimation du bruit et du SNR

Ce module permet d'estimer la valeur de la puissance du bruit ainsi que celle du SNR qui sont nécessaires pour l'évaluation de la fonction de coût représentée par l'équation 3.6. L'estimation de ces deux paramètres est effectuée en utilisant l'algorithme proposé dans [30].

## Module d'estimation de l'étalement Doppler

Ce module représente le cœur de notre travail. Il permet d'estimer la valeur de l'étalement Doppler en utilisant l'algorithme décrit dans le chapitre 3. En effet, le  $m^{ème}$  estimé local de l'étalement Doppler est obtenu numériquement en déterminant le maximum de la fonction de coût. L'évaluation locale de cette fonction pour chaque valeur de  $\sigma_D$  nécessite le calcul des variables suivantes dans l'ordre: les valeurs propres  $\sigma_1$  et  $\sigma_2$ , les vecteurs propres correspondants  $\mathbf{u}_1$  et  $\mathbf{u}_2$ , et finalement  $\mathcal{L}(\hat{\mathbf{h}}(m), \sigma_D)$ .

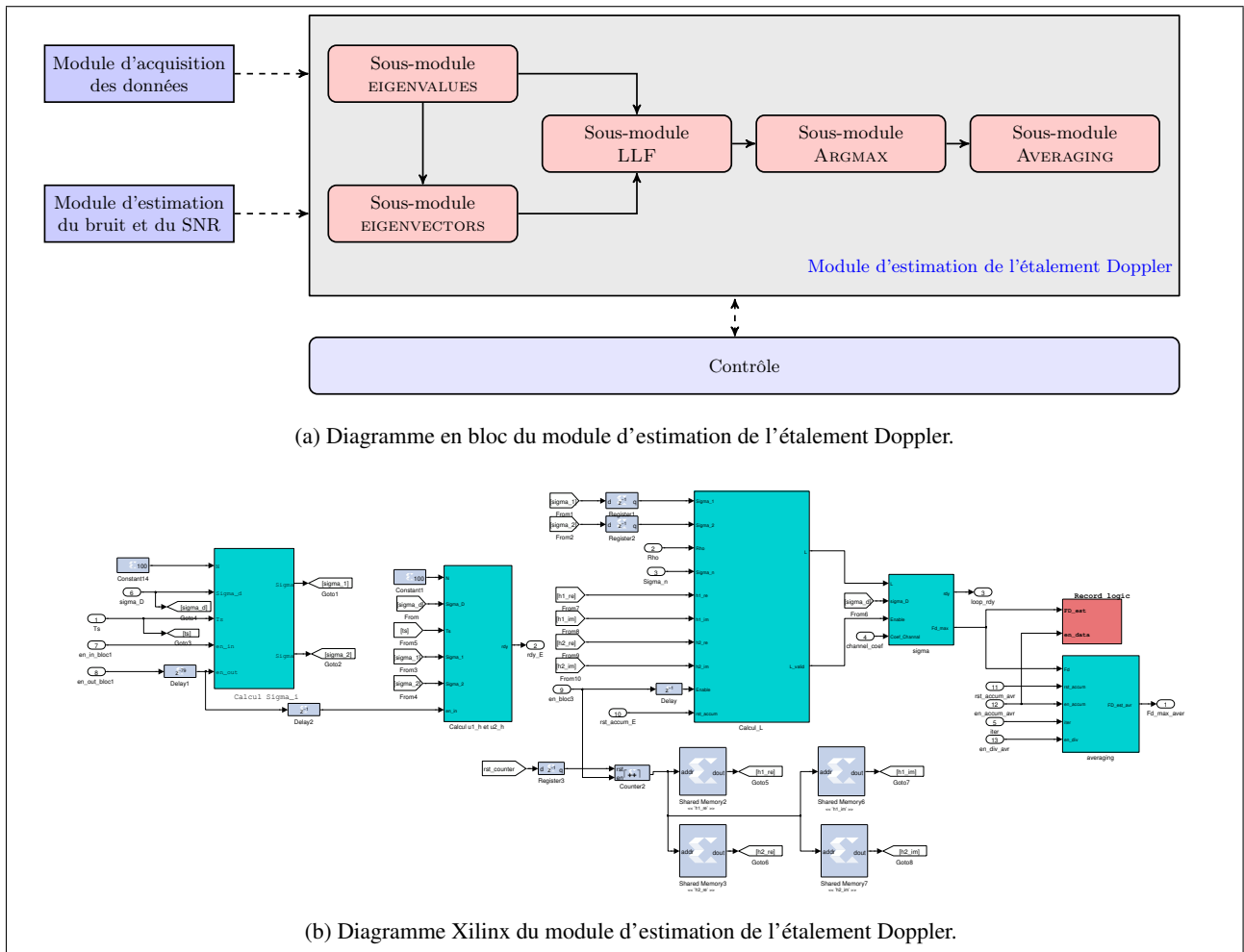


FIGURE 4.3 – Module d'estimation de l'étalement Doppler.

La figure 4.3 montre que le module d'estimation de l'étalement Doppler est composé de cinq sous-modules suivants:

- Le sous-module EIGENVALUES.

- Le sous-module EIGENVECTORS.
- Le sous-module LLF.
- Le sous-module ARGMAX.
- Le sous-module AVERAGING.

a. Le sous-module EIGENVALUES

Comme l'indique son nom, ce sous-module permet de calculer les deux valeurs propres  $\sigma_1$  et  $\sigma_2$  selon les deux expressions données par 3.7. Le sous-module prend comme entrées les signaux  $N$ ,  $T_s$ , et  $\sigma_D$  et fournit comme sorties  $\sigma_1$  et  $\sigma_2$ . Une fois qu'on obtient des sorties valides, un signal "ready" passe du niveau bas au niveau haut afin d'activer le sous-module suivant.

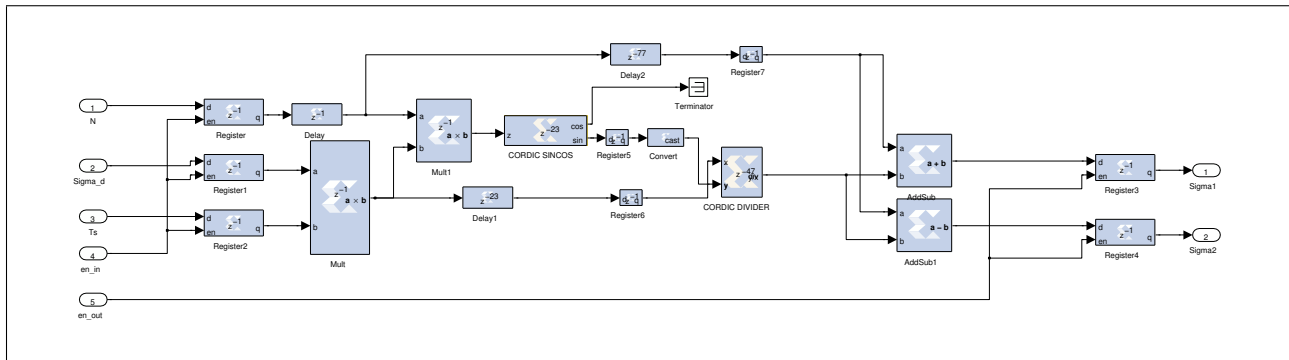


FIGURE 4.4 – Diagramme Xilinx du sous-module EIGENVALUES.

b. Le sous-module EIGENVECTORS

Ce sous-module prend comme entrées  $N$ ,  $T_s$ ,  $\sigma_n^2$ ,  $\sigma_1$ ,  $\sigma_2$ ,  $u_1^H$ ,  $u_2^H$ , et  $\hat{h}$ . Les deux signaux *Enable* et *rst\_accum* sont utilisés pour activer et contrôler certains blocs élémentaires au niveau de ce sous-module. Après son activation, ce module calcule les deux vecteurs propres  $u_1^H$  et  $u_2^H$  et stocke la partie réelle et imaginaire des éléments de chaque vecteur dans un module mémoire. Le calcul de ces deux vecteurs est basé sur les équations 3.8 et 3.9. Une fois qu'on obtient des sorties valides, le sous-module fournit un signal "ready" indiquant la fin du calcul.

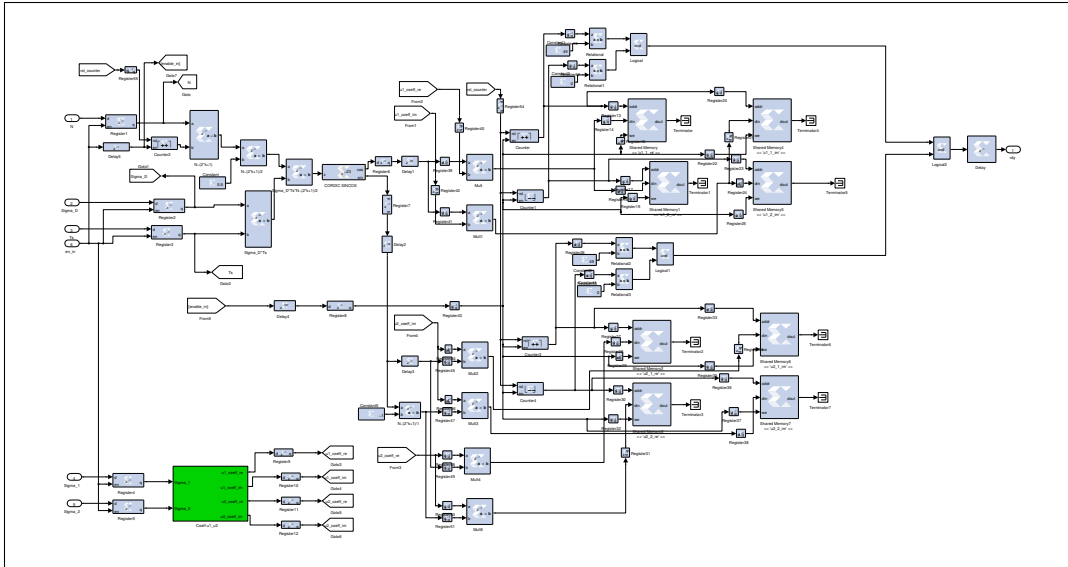


FIGURE 4.5 – Diagramme Xilinx de sous-module EIGENVECTORS.

c. Le sous-module LLF

Comme le montre la figure 4.6, ce sous-module prend comme entrées les signaux  $N, T_s, \hat{\rho}, \sigma_n^2, \sigma_1, \sigma_2, \mathbf{u}_1^H, \mathbf{u}_2^H$ , et  $\hat{\mathbf{h}}$  et évalue la fonction de coût pour chaque valeur candidate  $\sigma_D$  de l'étalement Doppler inconnu. Ce sous-module fournit deux signaux de sortie  $L\_valid$  and  $L$ . Le dernier contient la valeur de la fonction pour la valeur courante de  $\sigma_D$  et l'autre signal de sortie permet d'activer le sous-module suivant.

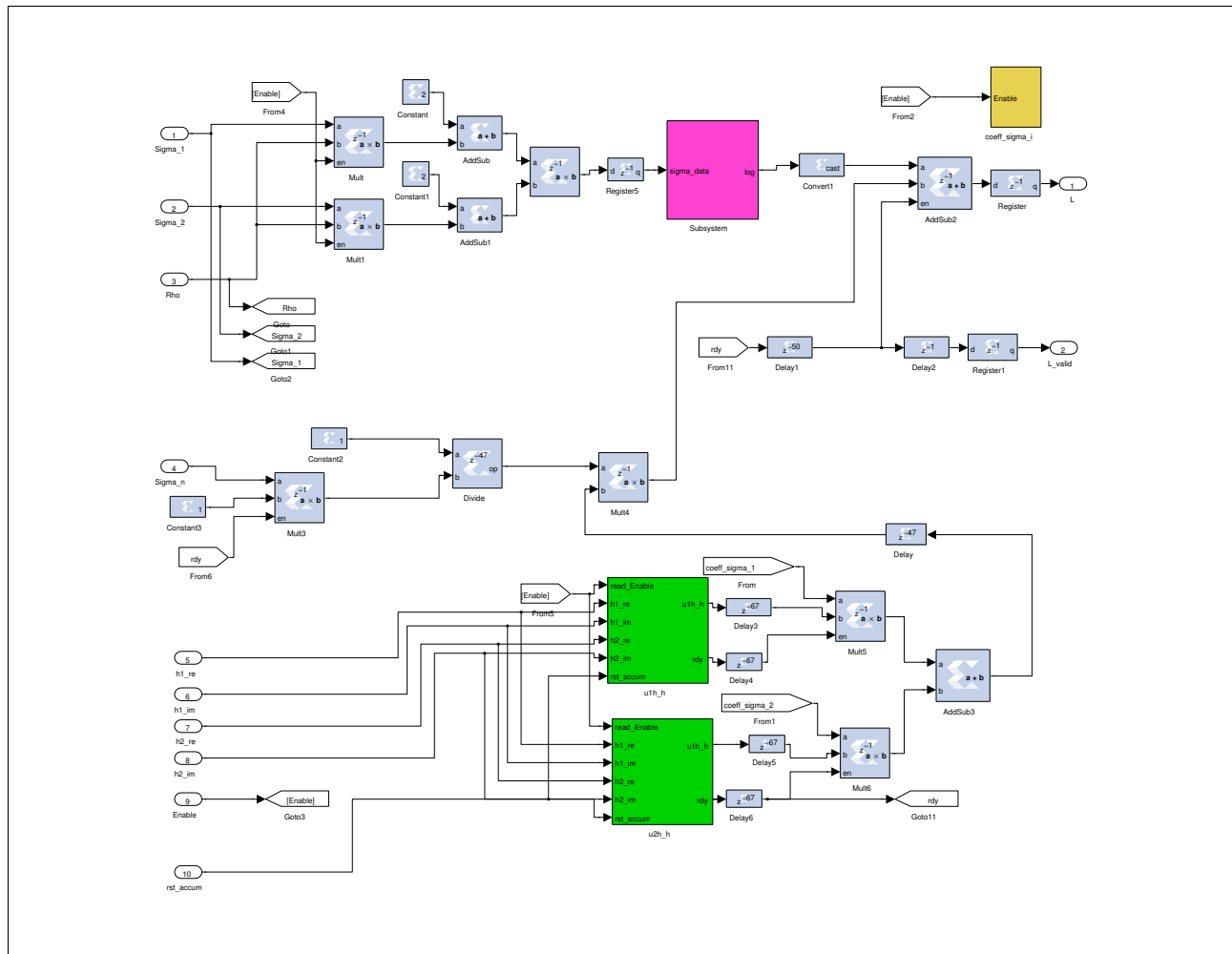


FIGURE 4.6 – Diagramme Xilinx du sous-module LLF.

#### d. Le sous-module ARGMAX

Ce sous-module prend comme entrées  $L$  et  $\sigma_D$ . Après son activation, il détermine l'argmax de la fonction de coût en comparant la valeur courante  $L$  avec la valeur calculée durant l'itération précédente et en gardant la valeur maximale ainsi que celle de  $\sigma_D$  correspondante. Ce sous-module fournit deux signaux de sortie  $Fd\_max$  and  $rdy$ . Le dernier permet de re-boucler en utilisant une nouvelle valeur de  $\sigma_D$ .

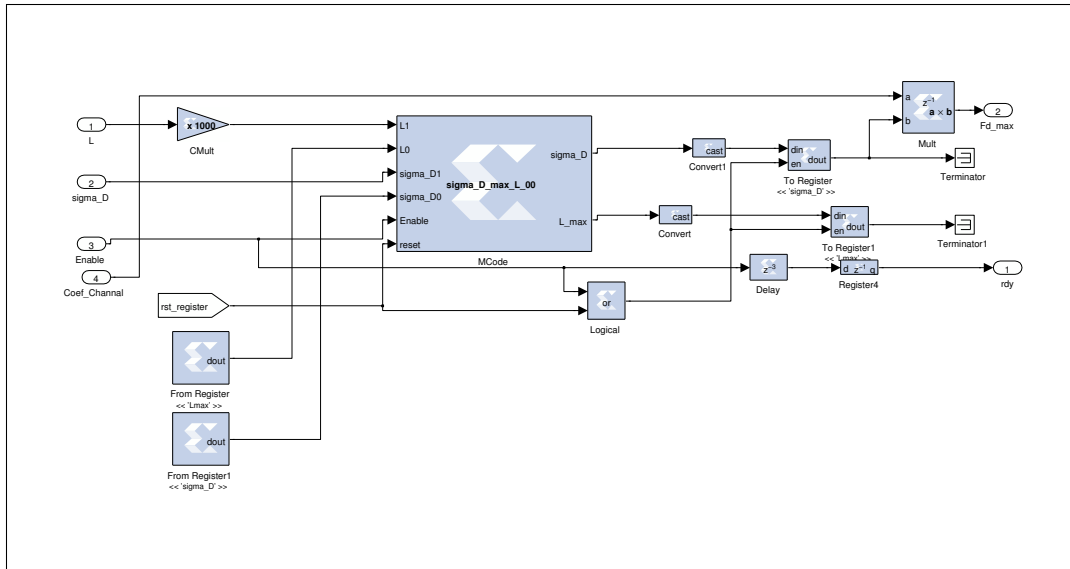


FIGURE 4.7 – Diagramme Xilinx du sous-module ARGMAX.

e. Le sous-module AVERAGING

Finalement, le sous-module AVERAGING permet de calculer la valeur moyenne des différentes estimées de l'étalement Doppler obtenues à partir des fenêtres d'approximation locales dans le but de minimiser le biais de l'estimation. Le nombre des fenêtres d'approximation est spécifié par l'utilisateur.

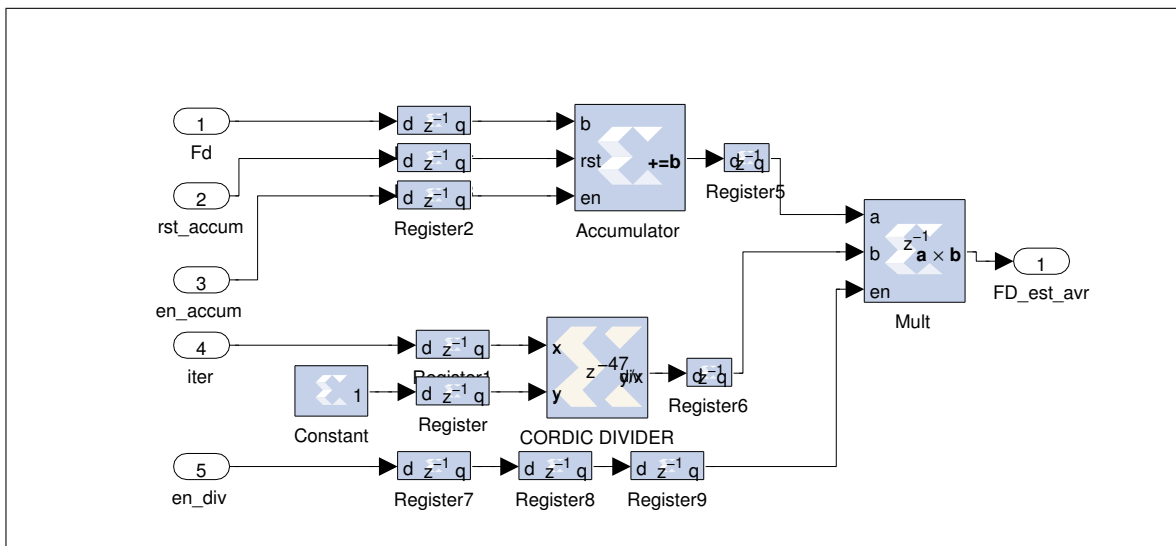


FIGURE 4.8 – Diagramme Xilinx du sous-module AVERAGING.

### 4.3.2 Chemin de contrôle

Le chemin de contrôle est constitué de trois machines à états finis (FSMs) qui permettent de contrôler les différents blocs constituant le chemin des données ainsi que de générer leurs signaux de commande. La figure 4.9 montre les diagrammes d'états de ces trois machines qui décrivent leur logique séquentielle. Nous avons utilisé le concept des machines de Moore lors de l'implémentation des ces trois machines. Ce type de machines est caractérisé par sa robustesse. Aussi, il garantit le bon fonctionnement du design puisque les signaux des sorties changent uniquement à la présence d'un front d'horloge contrairement aux machines de Mealy. En effet, au niveau de ces dernières, un changement des signaux d'entrée est capable d'engendrer un changement au niveau des signaux de sortie sans tenir compte de l'horloge pouvant ainsi générer le problème de la boucle de réaction asynchrone dans le cas d'interconnexion de plusieurs FSMs dans le même design. Les signaux de sortie décrits dans la figure 4.9 représentent les signaux de commande des différents blocs élémentaires utilisés pour construire les modules du chemin des données tels que le signal de commande d'un compteur, le signal de réinitialisation des mémoires et des accumulateurs, celui de commande des unités arithmétiques, etc. En outre, les trois machines sont implémentées en utilisant le bloc MCode fourni par l'outil XSG et qui permet d'utiliser des fonctions MATLAB basiques pour créer un bloc matériel au lieu d'utiliser la programmation en VHDL/Verilog.

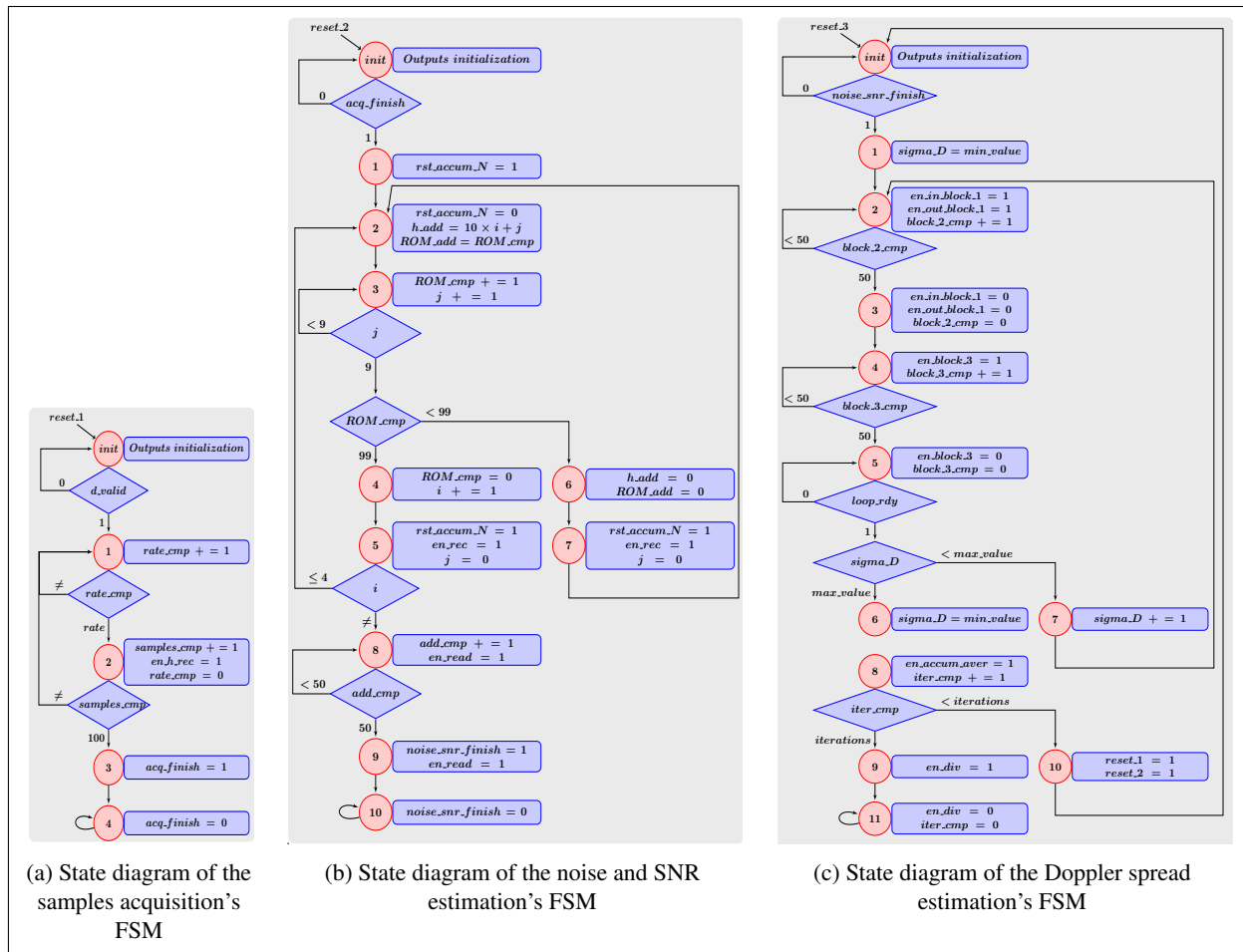


FIGURE 4.9 – Diagrammes d'états des différentes FSMs.

## 4.4 Génération du fichier « bitstream » et problèmes rencontrés

Après avoir développé notre design comme indiqué dans la figure 4.10, nous avons essayé de générer le fichier de programmation FPGA « bitstream ». Toutefois, lors de cette étape, nous étions confrontés à plusieurs problèmes. En effet, notre design a fonctionné pour une fréquence maximale de 10 MHz alors qu'il devait fonctionner pour une fréquence de 80 MHz suite à une contrainte imposée par la carte radio de notre plateforme PicoSDR. Afin de résoudre ce problème, nous avons cherché à optimiser le chemin critique du design en ajoutant des registres aux différents emplacements afin de subdiviser le chemin combinatoire. Nous avons également réduit la représentation binaire de quelques signaux qui ne nécessitaient pas une forte précision. Grâce à cette étape d'optimisation, nous avons réussi à satisfaire cette contrainte temporelle.



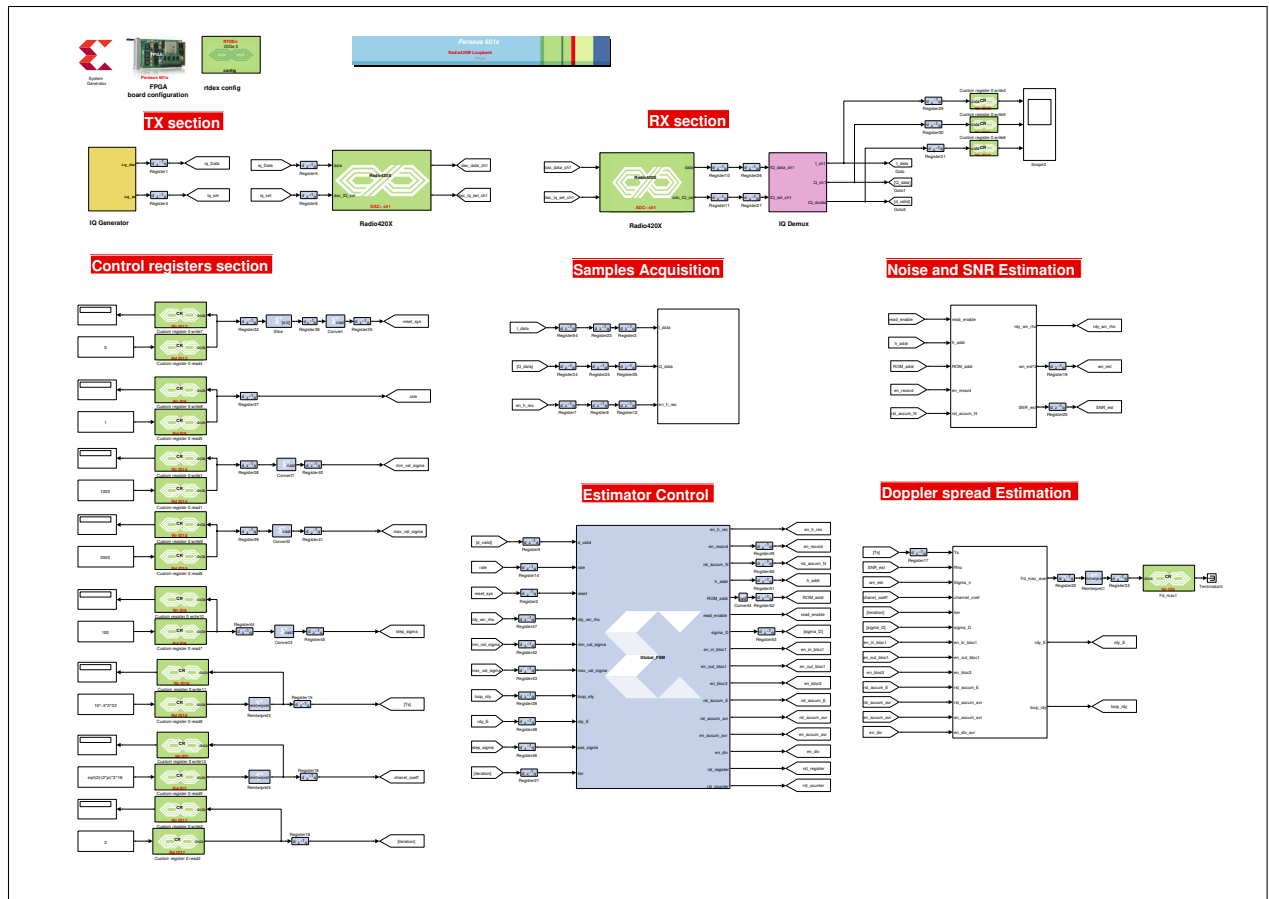


FIGURE 4.10 – Diagramme Xilinx du design.

## 4.5 Conclusion

Dans ce chapitre, Nous avons détaillé notre solution matérielle basée sur la technique MBD. L'implémentation est réalisée en utilisant les différents blocs élémentaires de XSG ainsi que le MBDK fourni par Nutaq. Quelques optimisations ont été nécessaires afin de satisfaire la contrainte temporelle imposée par la carte radio de la plateforme PicoSDR. Finalement, nous avons réussi à générer le fichier « bitstream » pour un design qui fonctionne à une cadence de 80 MHz

## Chapitre 5

# Évaluation des performances

### 5.1 Introduction

Pour évaluer les performances de notre design, nous utilisons la co-simulation matérielle au lieu d'une simulation classique. Dans ce chapitre, nous allons décrire la démarche suivie afin d'aboutir à cette co-simulation et nous allons présenter les différents résultats obtenus. Nous commençons par présenter le modèle de test que nous avons préparé au niveau de l'environnement Simulink. Ensuite, procédons à une présentation détaillée des résultats de la co-simulation.

### 5.2 Modèle Simulink de test

Pour effectuer la mesure de l'étalement Doppler et son évaluation par notre estimateur, nous avons développé le modèle de test suivant au niveau de Simulink comme le montre la figure 5.1.

Nous avons utilisé Les registres personnalisés fournis par le MBDK afin de configurer les paramètres des tests suivants :

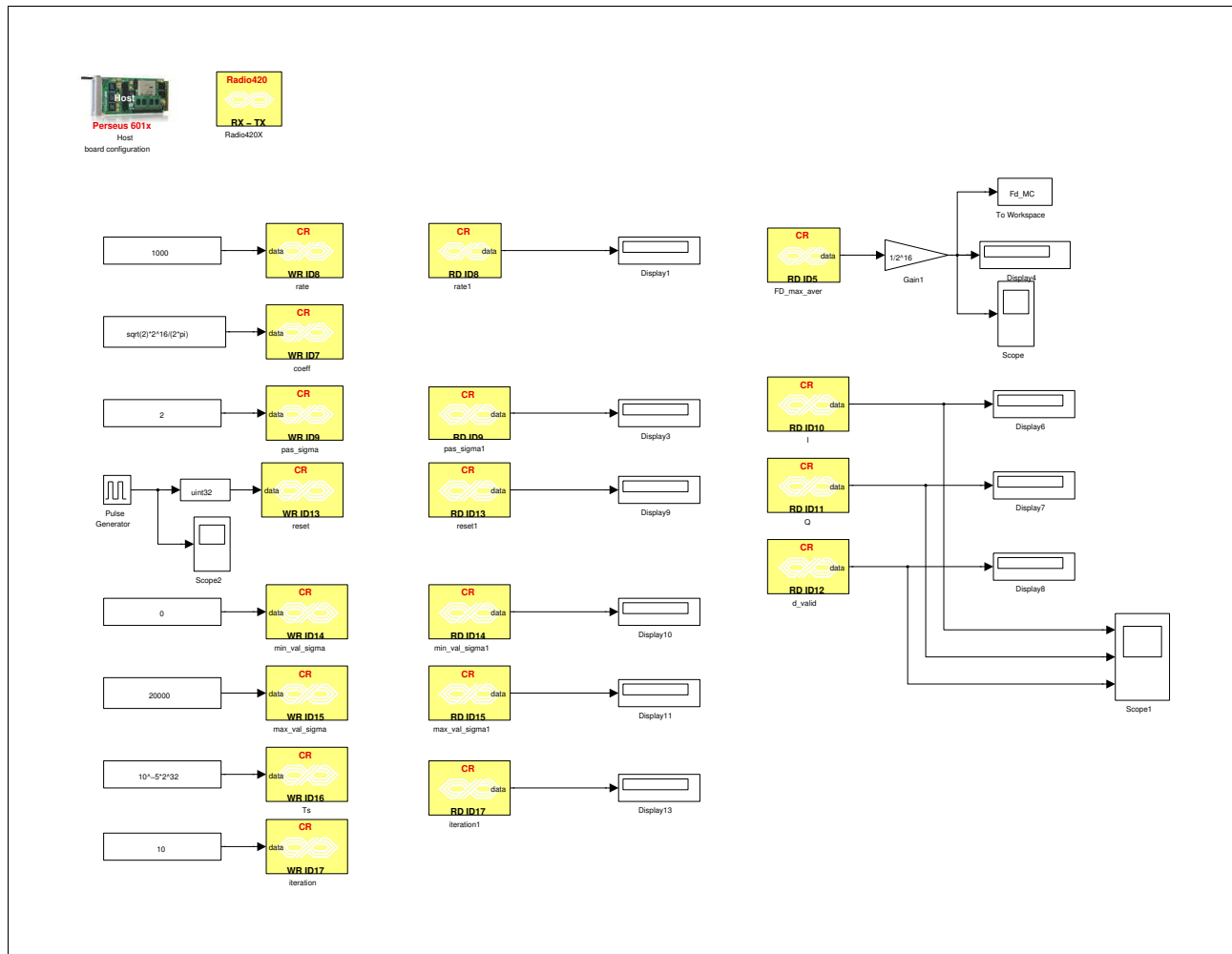


FIGURE 5.1 – Modèle Simulink de test

- Rate: définit la cadence utilisée au niveau du processus d'acquisition et reliée à chacune de l'horloge système du FPGA et la période symbole définie par le standard LTE.
- SigmaD\_step: représente le pas qu'on utilise pour le processus du « grid search » afin de trouver le  $\sigma_d$  adéquat qui maximise la fonction de coût.
- Reset: représente le signal d'initialisation du design. Nous l'avons utilisé au niveau des tests pour exécuter une boucle de co-simulation Monte-Carlo.
- $T_s$ : représente la période symbole.
- Iteration: représente le nombre de fenêtres locales d'approximation.
- Min\_val\_sigma, Max\_val\_sigma: définissent l'intervalle dans lequel le processus « grid search » est réalisé.

Le résultat est obtenu grâce au registre personnalisé « Fd\_aver ».

### 5.3 Résultats de la co-simulation

Afin de valider les performances du design de l'estimateur de l'étalement Doppler, nous avons comparé ses performances avec ceux obtenues à partir de la version MATLAB originale. Le modèle de canal considéré est le même pour chacune des versions MATLAB et PicoSDR/Anite. Nous avons implémenté nos tests à travers une boucle de co-simulation Monte-Carlo de 10000 itérations qui nous a permis de représenter les graphiques suivants :

- L'erreur quadratique moyenne normalisée (NMSE) en fonction de  $FdT_s$ .
- NMSE en fonction du SNR.
- Histogramme de l'estimée de la fréquence Doppler maximale normalisée.
- Histogramme de l'erreur quadratique normalisée (NSE) de la fréquence maximale de Doppler.
- La fonction de répartition (CDF) de NSE.

#### 5.3.1 NMSE en fonction de $FdT_s$

Les valeurs de l'estimée de la la fréquence Doppler maximale, que nous avons obtenues durant la phase de co-simulation, nous ont permis de calculer le NMSE de la version PicoSDR/Anite de l'estimateur. La comparaison avec la version MATLAB donne les résultats suivants:

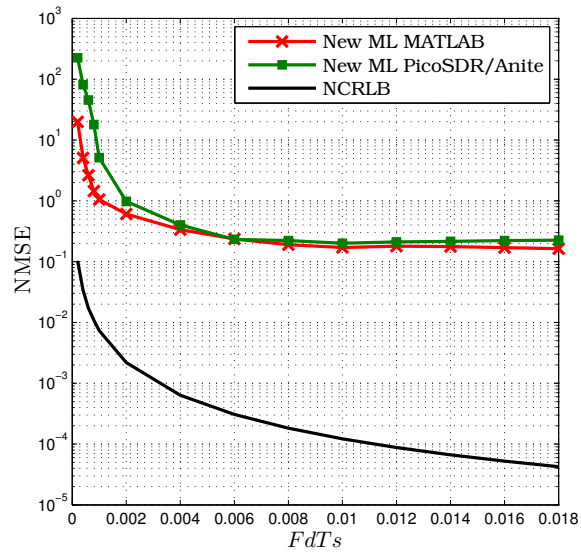


FIGURE 5.2 – NMSE vs.  $F_d T_s$  avec  $T_s = 10 \mu s$ ,  $SNR = -10 dB$  et  $N = 1000$ .

On remarque bien que les deux courbes de MATLAB et picoSDR/Anite sont pratiquement confondues sauf pour les petites valeurs de l'étalement Doppler où on constate un léger écart.

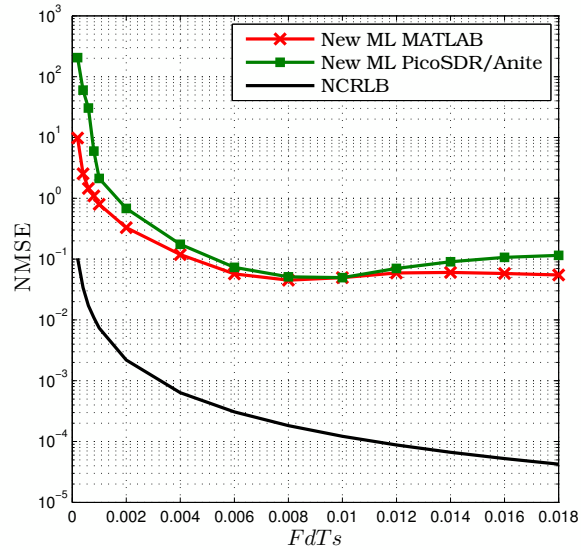


FIGURE 5.3 – NMSE vs.  $F_d T_s$  avec  $T_s = 10 \mu s$ ,  $SNR = -5 dB$  et  $N = 1000$ .

En augmentant la valeur de  $SNR$ , les deux courbes restent proches mais présentent une marge qui devient plus en plus importante. Cette marge est plus remarquable au niveau des figures ci-dessous.

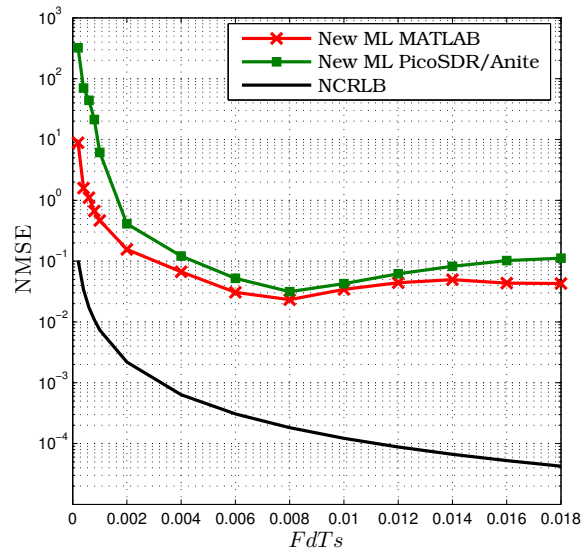


FIGURE 5.4 – NMSE vs.  $F_d T_s$  avec  $T_s = 10 \mu s$ ,  $SNR = 0 \text{ dB}$  et  $N = 1000$ .

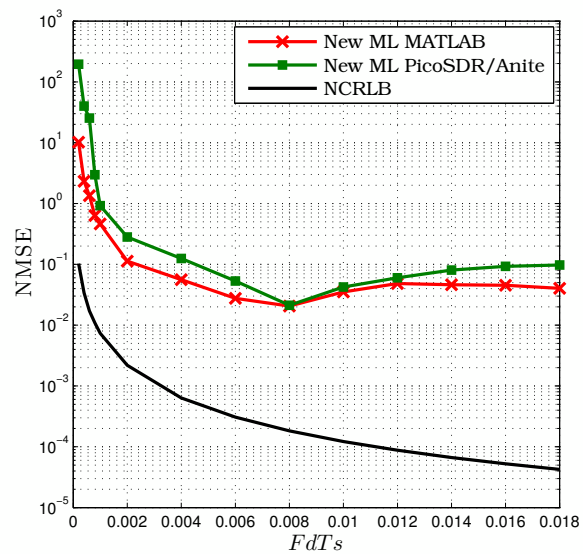


FIGURE 5.5 – NMSE vs.  $F_d T_s$  avec  $T_s = 10 \mu s$ ,  $SNR = 5 \text{ dB}$  et  $N = 1000$ .

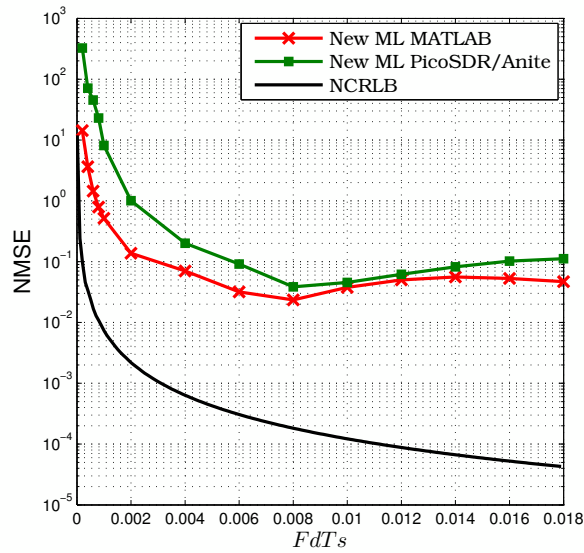


FIGURE 5.6 – NMSE vs.  $F_d T_s$  avec  $T_s = 10 \mu s$ ,  $SNR = 10 \text{ dB}$  et  $N = 1000$ .

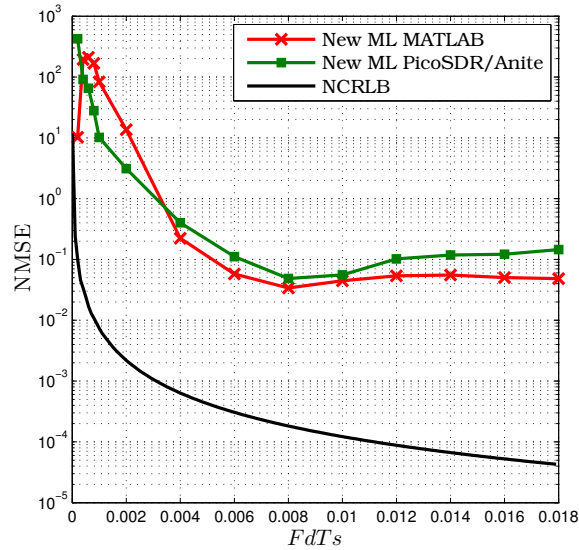


FIGURE 5.7 – NMSE vs.  $F_d T_s$  avec  $T_s = 10 \mu s$ ,  $SNR = 20 \text{ dB}$  et  $N = 1000$ .

Comme illustré ci-dessus, dans chacun des cas, les courbes de MATLAB et picoSDR/Anite sont proches l'une de l'autre et suivent la même tendance. La marge entre les deux courbes est due au fait que la version de MATLAB ne prenne pas en compte les effets des hautes fréquences causés par les filtres passe-haut, les amplificateurs de puissance (PAs) et les amplificateurs faible bruit (LNA). En outre, l'implémentation

provoque des erreurs dues aux limites de sa représentation binaire à point fixe par rapport à celle à point flottant au niveau de MATLAB. Il est en effet difficile de prévoir l'ordre de grandeur des différents signaux afin de choisir la représentation binaire adéquate qui minimise ces erreurs d'implémentation.

### 5.3.2 NMSE en fonction du SNR

Nous avons tracé la courbe de NMSE en fonction du  $SNR$  des deux version de l'estimateur pour différentes valeurs du  $F_d T_s$  et nous avons obtenu les résultats suivants:

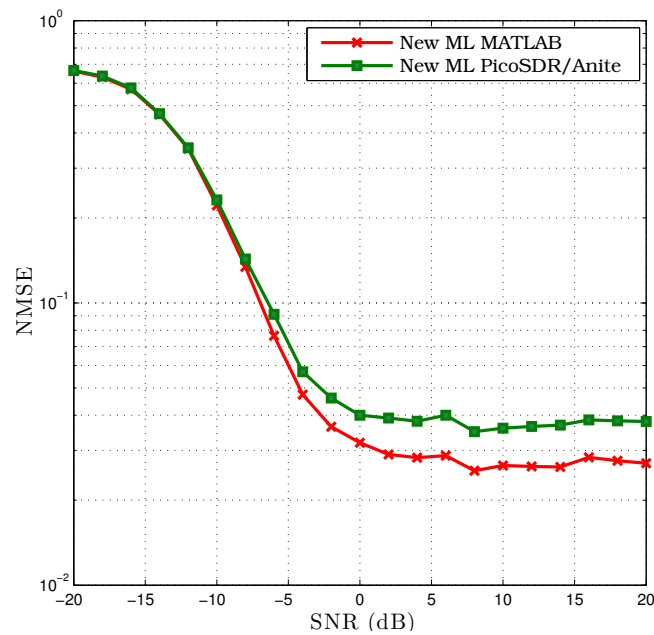


FIGURE 5.8 – NMSE vs. SNR avec  $F_d T_s = 0.006$  et  $N = 1000$ .



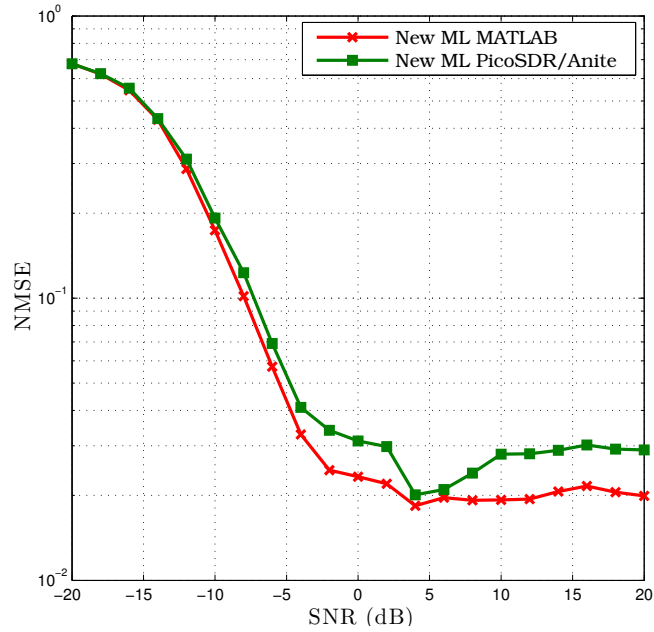


FIGURE 5.9 – NMSE vs. SNR avec  $F_d T_s = 0.008$  et  $N = 1000$ .

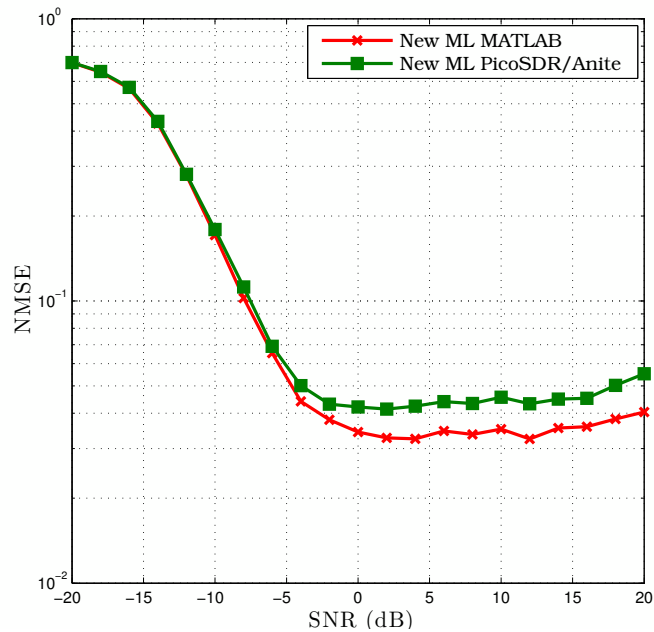


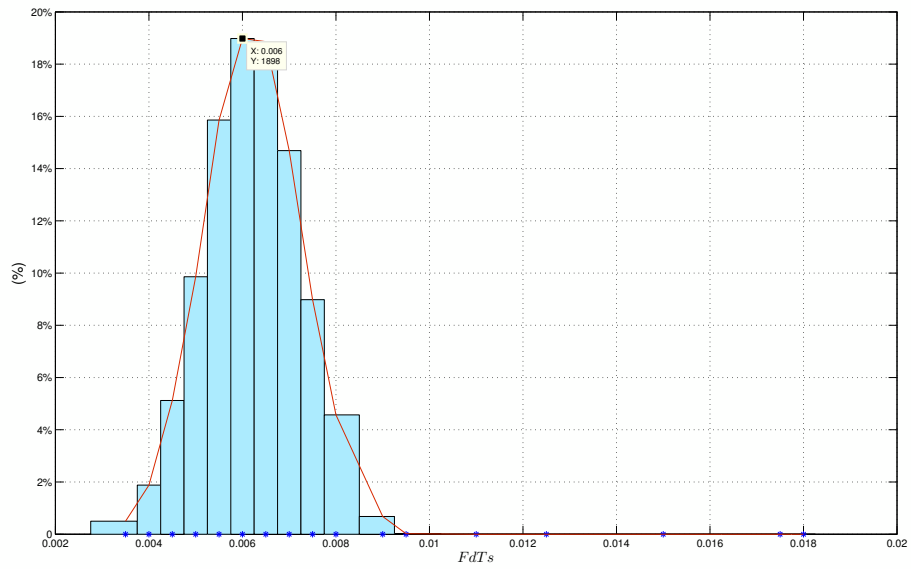
FIGURE 5.10 – NMSE vs. SNR avec  $F_d T_s = 0.01$  et  $N = 1000$ .

Comme le montre les Figures de 5.8 à 5.10, nous constatons encore une fois que la courbe de MATLAB et celle de PicoSDR /Anite sont relativement proches l'une de l'autre et suivent la même tendance. Pour des valeurs de  $SNR$  supérieurs à -5 dB, nous observons un léger écart, qui peut atteindre une valeur maximale de  $10^{-2}$ , dû aux mêmes raisons expliquées précédemment. Pour des valeurs de  $SNR$  supérieurs à 5 dB, la

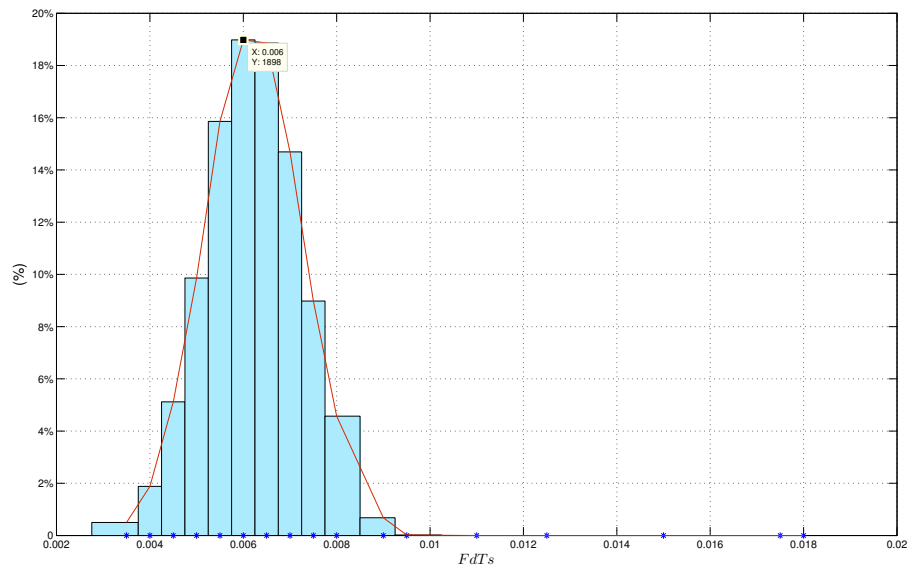
courbe de NMSE devient pratiquement constante, cela est dû au fait que la fonction de coût (equation 3.6) ne subit pas une grande variation lorsque la valeur de  $SNR$  devient de plus en plus importante.

### **5.3.3 Histogramme de l'estimée de la fréquence Doppler maximale normalisée**

Nous avons tracé l'histogramme illustrant l'estimée de la fréquence Doppler maximale normalisée pour la version MATLAB de l'estimateur ainsi que sa version PicoSDR/Anite pour différentes valeurs du  $SNR$ . Ainsi, nous avons obtenu les résultats suivants:

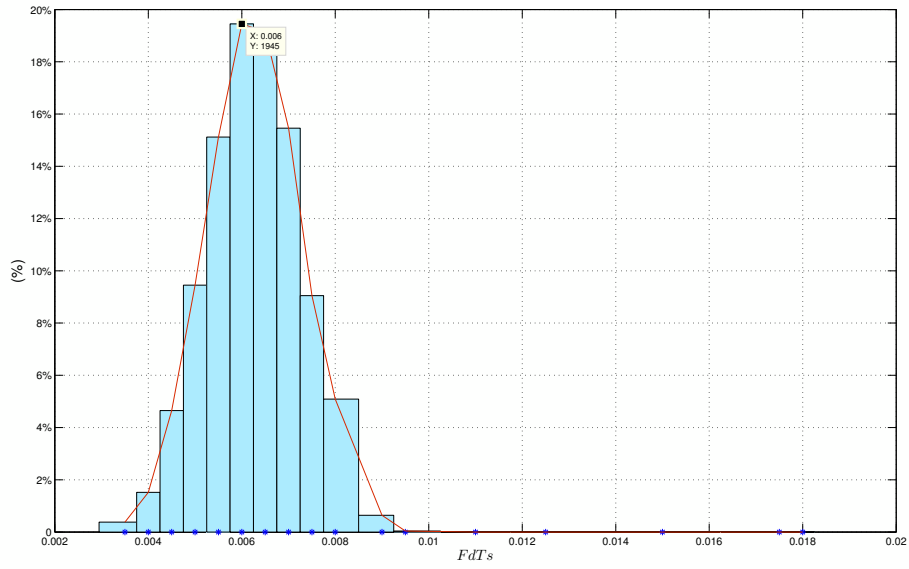


(a) La version MATLAB

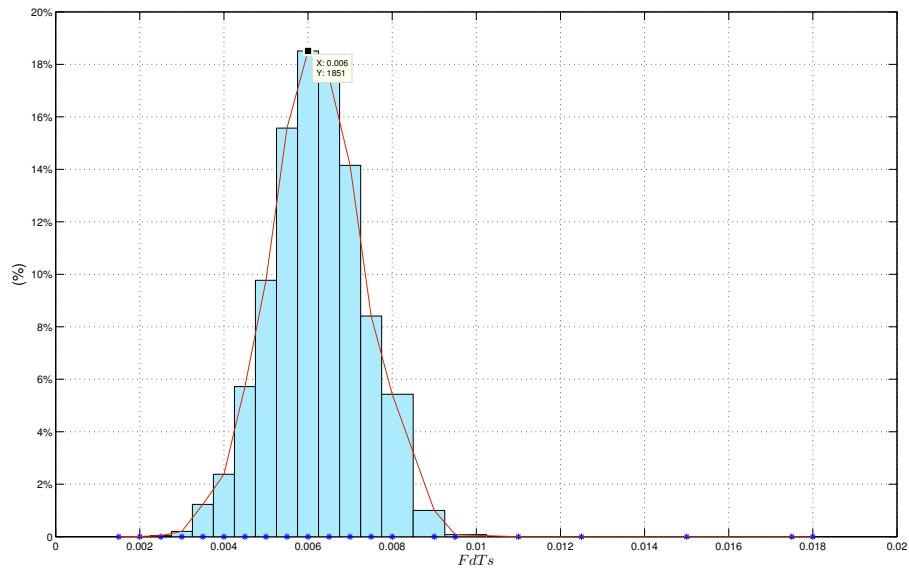


(b) La version PicoSDR/Anite

**FIGURE 5.11 – Histogramme de l'estimée de la fréquence Doppler maximale normalisée:  $F_d T_s = 0.006$ ,  $SNR = 0$  dB,  $N = 1000$  et  $MC = 10000$ .**

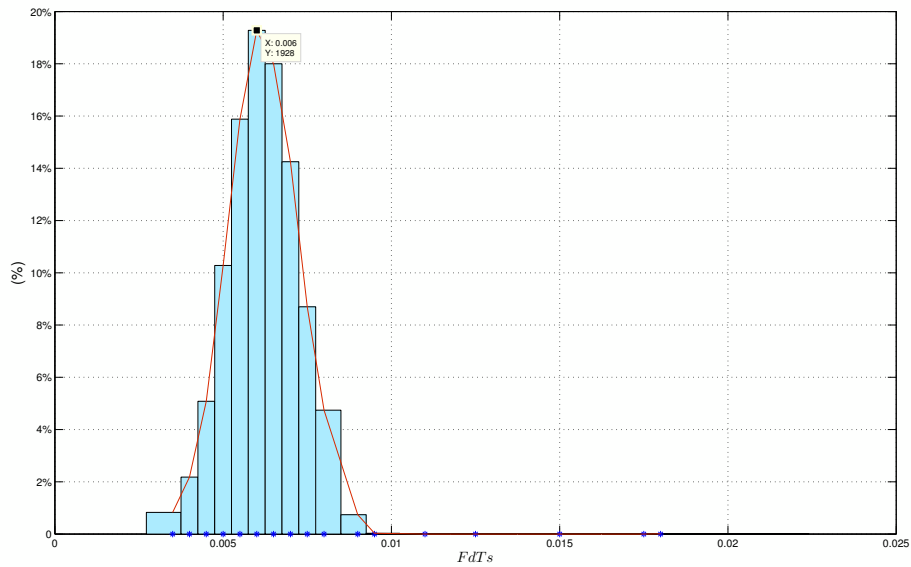


(a) La version MATLAB

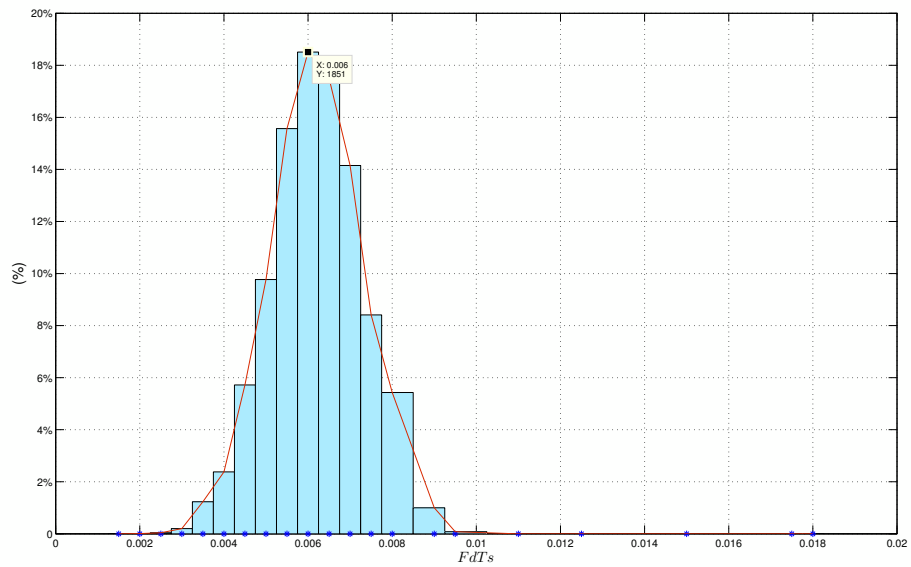


(b) La version PicoSDR/Anite

**FIGURE 5.12 – Histogramme de l'estimée de la fréquence Doppler maximale normalisée:  $F_d T_s = 0.006$ ,  $SNR = 10$  dB,  $N = 1000$  et  $MC = 10000$ .**



(a) La version MATLAB



(b) La version PicoSDR/Anite

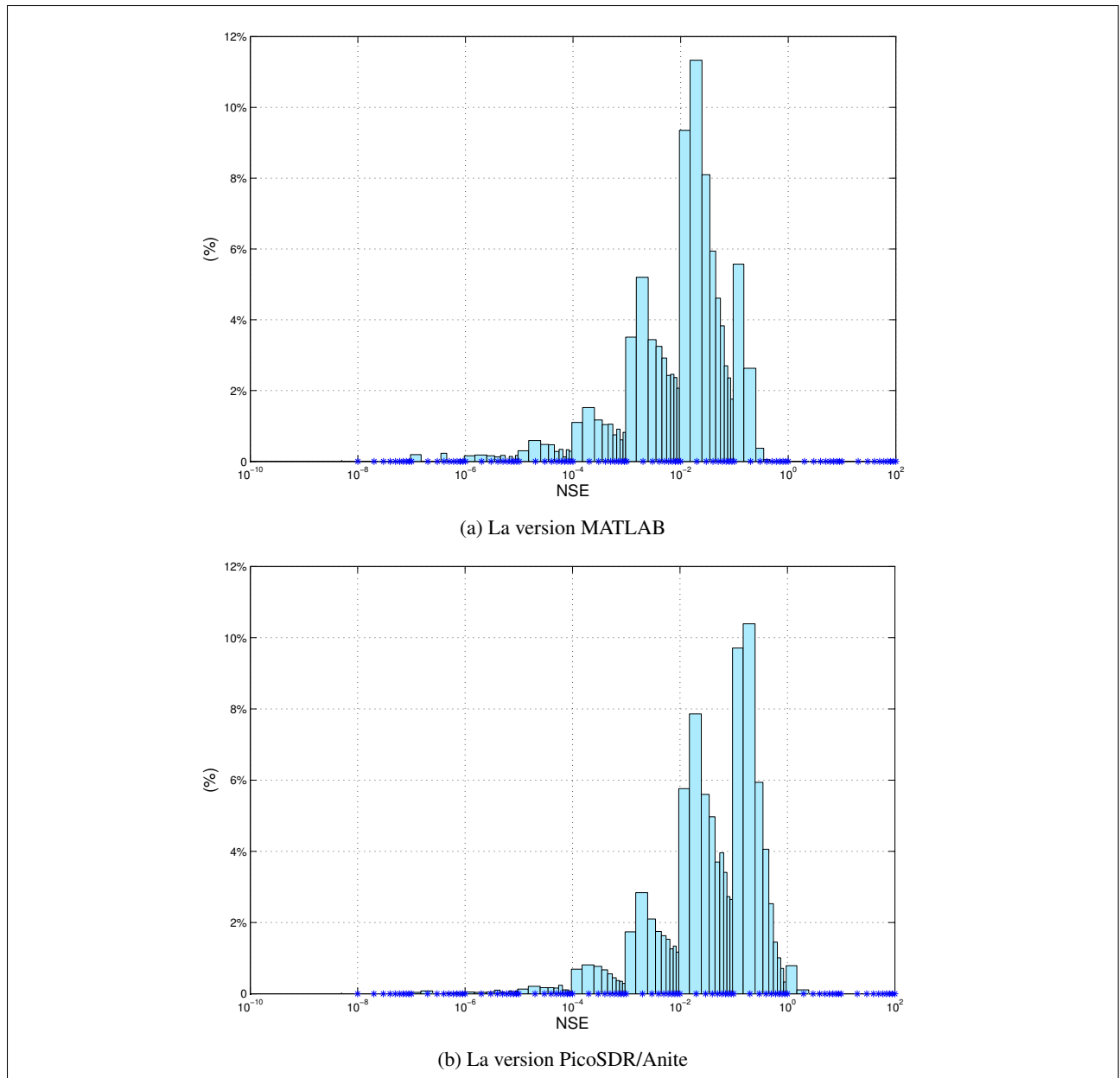
**FIGURE 5.13 – Histogramme de l’estimée de la fréquence Doppler maximale normalisée:  $F_d T_s = 0.006$ ,  $SNR = 20$  dB,  $N = 1000$  et  $MC = 10000$ .**

Comme illustré ci-dessus, nous avons obtenu des histogrammes gaussiens pour différentes valeurs du  $SNR$ . Ce résultat était prévisible puisque nous avons utilisé une boucle Monte-Carlo avec  $MC=10000$ .

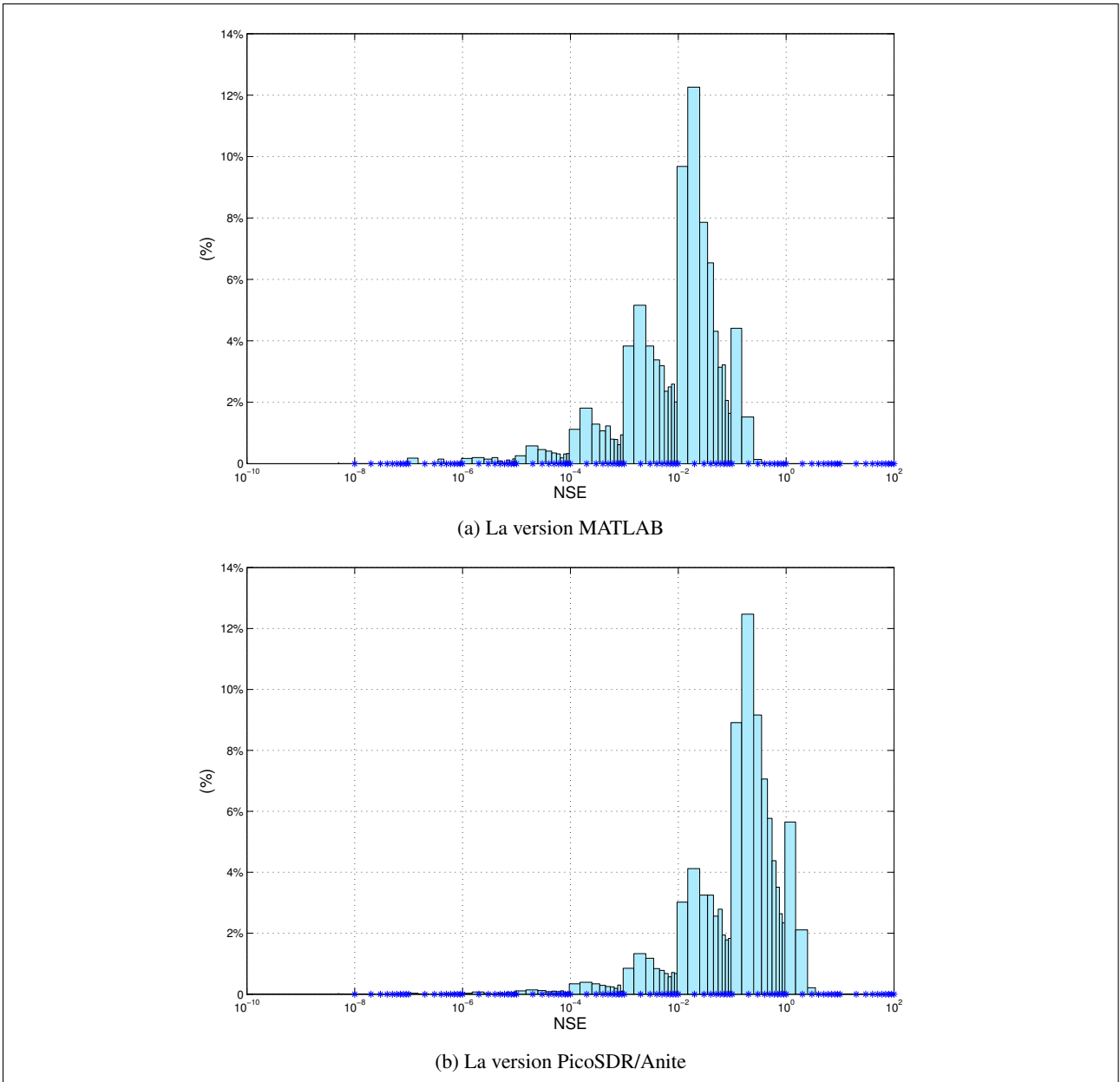
Nous avons aussi vérifié que la moyenne de la distribution gaussienne n'est autre que la valeur du  $F_d T_s$  à estimer.

### **5.3.4 Histogramme de l'erreur quadratique normalisée (NSE) de l'estimée de la fréquence maximale de Doppler**

Les figures ci-dessous représentent l'histogramme de la NSE de l'estimée de la fréquence maximale de Doppler pour chacune des versions MATLAB et PicoSDR/Anite pour différentes valeurs du  $SNR$ .

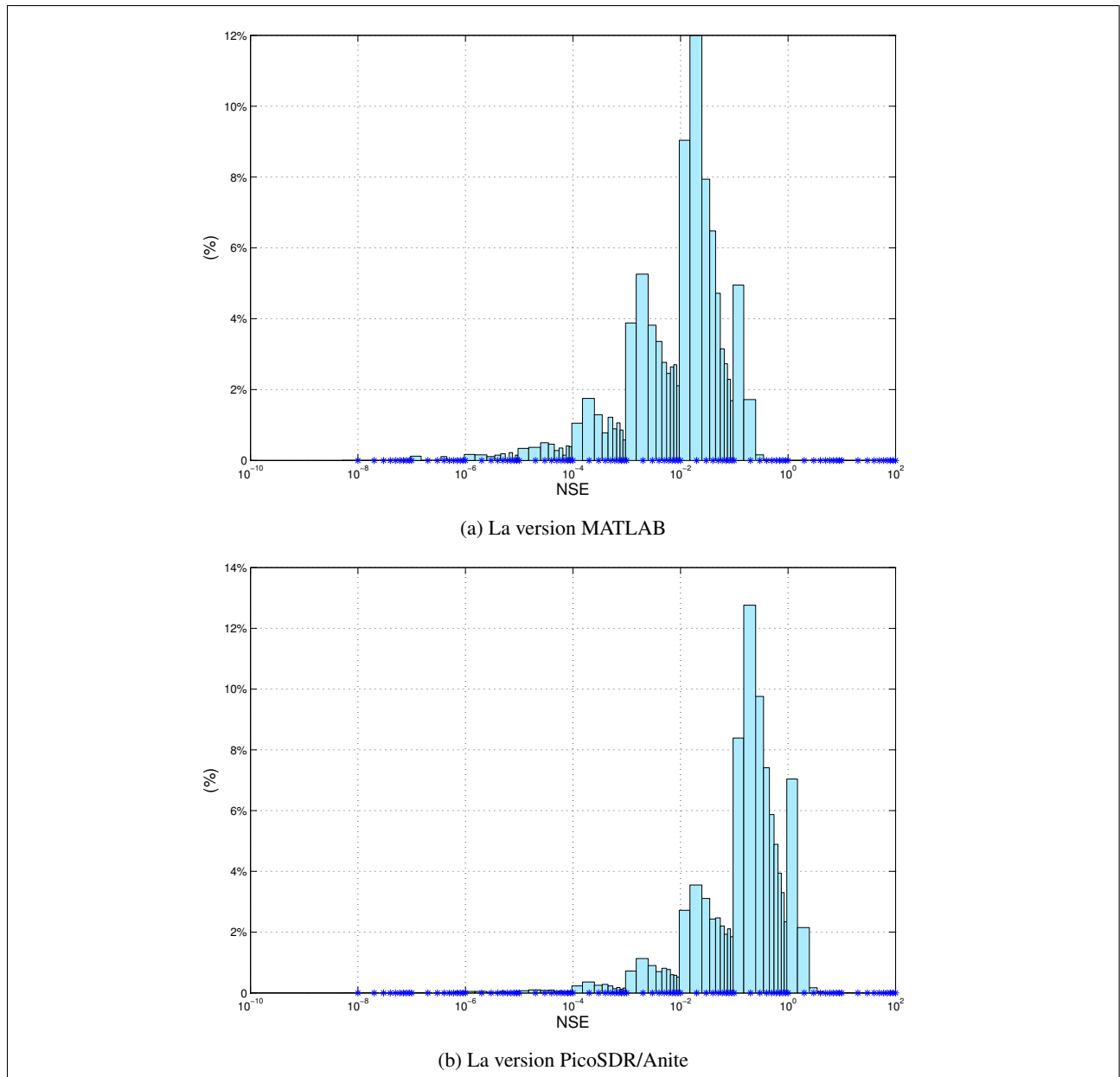


**FIGURE 5.14 – Histogramme de NSE de l’estimée de la fréquence maximale de Doppler:  $F_d T_S = 0.006$ ,  $SNR = 0$  dB,  $N = 1000$  et  $MC = 10000$ .**



**FIGURE 5.15 – Histogramme de NSE de l’estimée de la fréquence maximale de Doppler:  $F_d T_S = 0.006$ ,  $SNR = 10$  dB,  $N = 1000$  et  $MC = 10000$ .**





**FIGURE 5.16 – Histogramme de NSE de l’estimée de la fréquence maximale de Doppler:  $F_d T_S = 0.006$ ,  $SNR = 20$  dB,  $N = 1000$  et  $MC = 10000$ .**

Comme illustré ci-dessus, tous les histogrammes ont la même forme mais nous remarquons que ceux de la version de MATLAB sont centrés autour des valeurs de NSE se situant dans l’intervalle  $[10^{-4}, 10^{-1}]$ . Quant aux histogrammes de la version PicoSDR/Anite, ils sont centrés autour de l’intervalle  $[10^{-3}, 1]$ . Ainsi, la valeur de NSE de la version MATLAB est meilleure avec un rapport égal à 10.

### 5.3.5 La fonction de répartition (CDF) de NSE

Dans cette section, nous avons tracé la fonction de répartition en fonction de NSE pour différentes valeurs du  $SNR$ . Pour une précision donnée du NSE, nous remarquons un écart entre les version MATLAB et PicoSDR/Anite qui se voit accroître selon les valeurs du  $SNR$  suite à la dégradation des performances de l'estimation des valeurs élevées du  $SNR$ . En effet, ce phénomène est bien connu et il est attribué à la difficulté d'estimer un niveau faible de bruit avec une bonne précision.

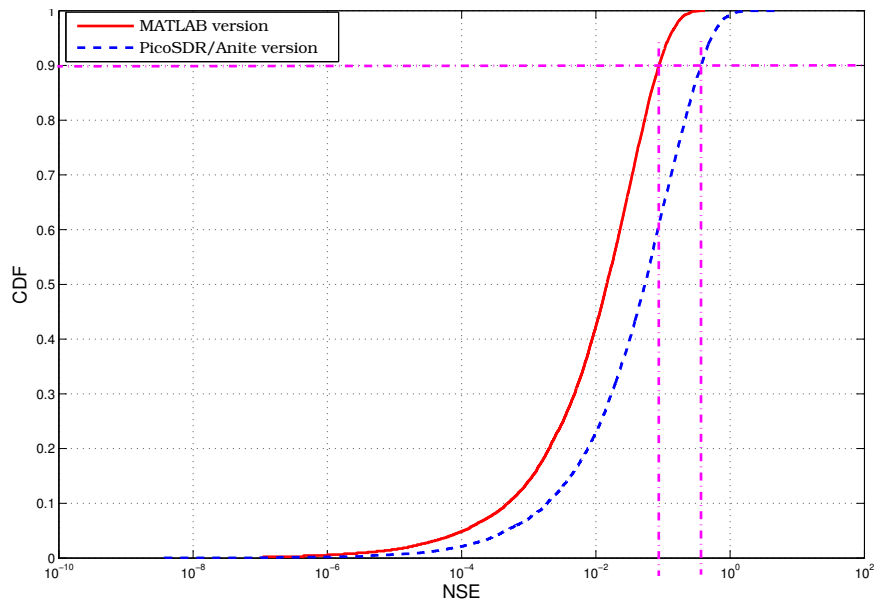


FIGURE 5.17 – CDF vs. NSE avec  $F_d T_s = 0.006$ ,  $SNR = 0$  dB et  $N = 1000$ .

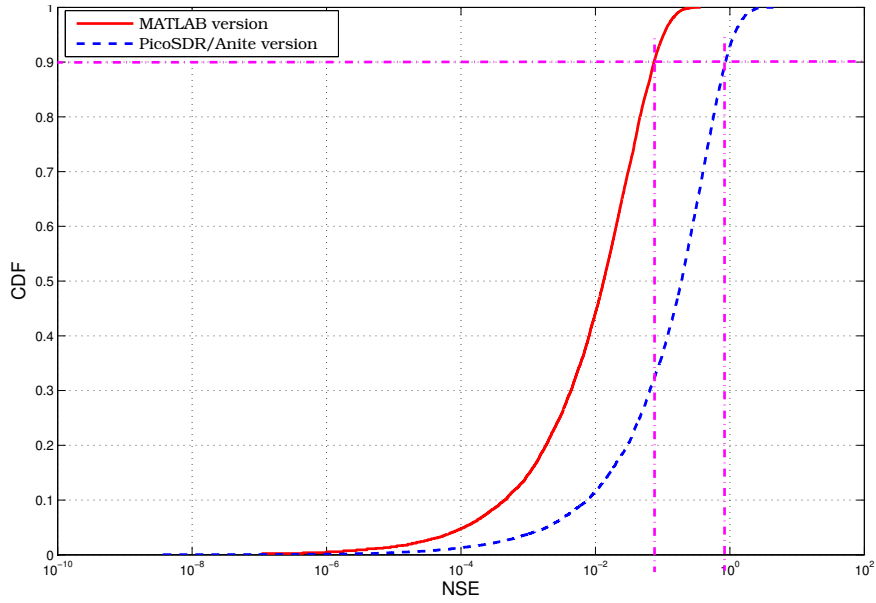


FIGURE 5.18 – CDF vs. NSE avec  $F_d T_s = 0.006$ ,  $SNR = 10$  dB et  $N = 1000$ .

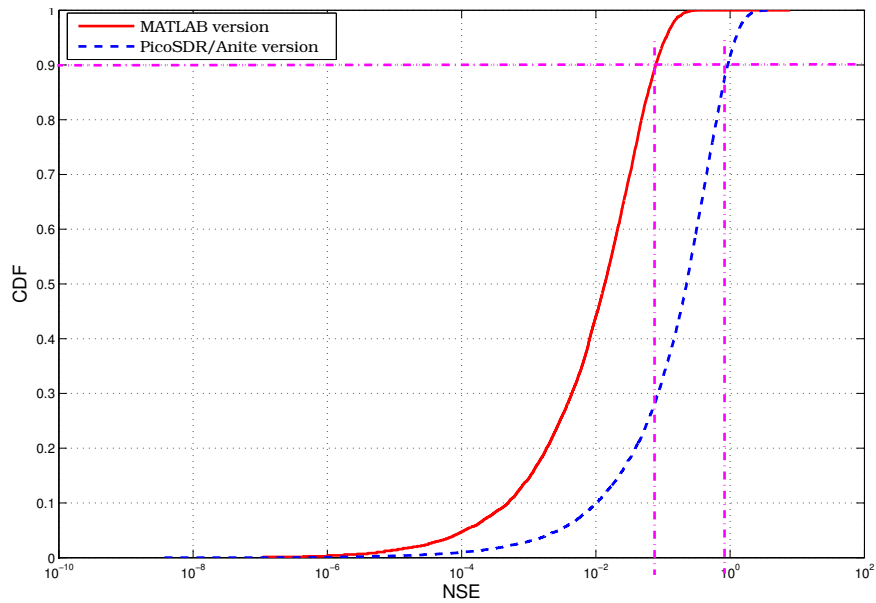


FIGURE 5.19 – CDF vs. NSE avec  $F_d T_s = 0.006$ ,  $SNR = 20$  dB et  $N = 1000$ .

## 5.4 Conclusion

Dans ce chapitre, nous avons présenté le modèle de test Simulink utilisé pour exécuter le fichier de bits-tream chargé au niveau de la plateforme PicoSDR à travers le logiciel ISE iMPACT. Nous avons effectué nos

tests à travers une boucle de co-simulation Monte-Carlo de 10000 itérations afin d'évaluer les performances de notre design. Les résultats obtenus pour chacune des versions MATLAB et PicoSDR/Anite sont bien proches avec un écart bien léger qui augmente en fonction des valeurs du  $SNR$ . Cet écart est expliqué par le fait que la version MATLAB ne prenne pas en considération les effets des hautes fréquences causées par les filtres passe-haut, les PAs et les LNAs. De plus, l'implémentation entraîne des erreurs dues à la limitation de la représentation binaire à point fixe par rapport à celle à point flottant dans le cas de MATLAB. Nous avons aussi vérifié que la moyenne de la distribution gaussienne obtenue correspond bien à la valeur de  $F_d T_s$  à estimer. En outre, les résultats obtenus montrent que notre estimateur présente certaines limites pour les valeurs élevées de l'étalement Doppler (canal à évanouissement rapide). Cependant, il reste meilleur que les autres estimateurs proposés dans la littérature.

## Chapitre 6

# Conclusion générale

Les systèmes de communication sans-fil ont connu une évolution technologique importante ces dernières années. Le déploiement des systèmes embarqués dans le domaine des communications sans-fil a permis de répondre rapidement aux demandes sans cesse croissantes. L'invention du FPGA a rendu possible le concept du matériel re-configurable pour la radio logicielle. Cependant, les méthodes traditionnelles de conception des radios architecturées autour du FPGA ne permettent pas une productivité élevée. Ce problème a été surmonté avec l'apparition de la technique de conception MBD. Les outils ainsi développés et basés sur cette technique ont permis une grande avancée dans le prototypage rapide de différentes applications dans le domaine des communications sans-fil.

Dans un tel domaine, le canal est caractérisé par son aspect d'évanouissement variant et lié à l'étalement Doppler. La connaissance de ce dernier représente un paramètre clé pour de nombreuses applications. Plusieurs estimateurs de l'étalement Doppler ont été proposés dans la littérature. Cependant, c'est l'estimateur qui montre le compromis le plus attractif entre la complexité du calcul et la précision de l'estimation qui est habituellement considérée comme étant le meilleur candidat pour l'implémentation matérielle dans la pratique.

Dans ce mémoire, nous avons adopté la technique MBD pour implémenter et valider un estimateur de l'étalement Doppler développé par notre groupe de recherche au sein de Wirelesslab sur la plateforme

PicoSDR fournie par Nutaq. Au meilleur de notre connaissance, c'est la toute première implémentation matérielle d'un estimateur de l'étalement Doppler dans la littérature. Nous avons exploité les différents blocs de la bibliothèque de XSG ainsi que le kit MBDK développé par Nutaq dans notre implémentation. En outre, nous avons utilisé, pendant la phase d'évaluation, un émulateur de canal qui reproduit les canaux radios du monde réel avec une grande fidélité et permet ainsi de procéder à des tests de performances OTA.

Enfin, les résultats de co-simulation ont été fournis pour illustrer les performances de notre design. En effet, nous avons comparé ces résultats avec ceux obtenus par le modèle équivalent développé sous MATLAB. Cette comparaison a montré que les résultats pour chacune des versions MATLAB et PicoSDR/Anite sont bien proches avec un écart faible et léger qui augmente avec le  $SNR$ . Cet écart reste inférieur à  $10^{-2}$  et il est expliqué par le fait que la version MATLAB ne prend pas en considération les effets des hautes fréquences causés par les filtres passe-haut, les PAs et les LNAs. De plus, l'implémentation entraîne des erreurs dues à la limitation de la représentation binaire à point fixe par rapport à celle à point flottant dans le cas de MATLAB. Ainsi, les performances de notre design peuvent être améliorées dans des travaux futurs en optimisant le module d'estimation du  $SNR$ . D'autres estimateurs de l'étalement Doppler peuvent être aussi implémentés afin de comparer leurs versions matérielles avec celle du l'estimateur choisi dans ce mémoire.

# Références

- [1] S. Affes and P. Mermelstein, “A new receiver structure for asynchronous CDMA: STAR-the spatio-temporal array-receiver,” *IEE Journal on Selected Areas in Communications*, vol. 16, no. 8, pp. 1411–1422, 1998.
- [2] C. Tepedelenlioglu, A. Abdi, G.B. Giannakis, and M. Kaveh, “Estimation of Doppler spread and signal strength in mobile communications with applications to handoff and adaptive transmission,” *Wireless Communications and Mobile Computing*, vol. 1, pp. 221–242, 2001.
- [3] G. L. Stüber, *Principles of Mobile Communication (2nd Edition)*. Kluwer, 2001.
- [4] Y.-R. Tsai and K.-J. Yang, “Approximate ML Doppler spread estimation over flat Rayleigh fading channels,” *IEEE Signal Processing Letters*, vol. 16, no. 11, pp. 1007–1010, 2009.
- [5] M. Souden, S. Affes, J. Benesty, and R. Bahroun, “Robust Doppler spread estimation in the presence of a residual carrier frequency offset,” *IEEE Transactions on Signal Processing*, vol. 57, no. 10, pp. 4148–4153, 2009.
- [6] F. Bellili and S. Affes, “A low-cost and robust maximum likelihood Doppler spread estimator,” in *Globecom-Wireless Communications Symposium*. IEEE, 2013, pp. 4430–4435.
- [7] V. B. Alluri, J. R. Heath, and M. Lhamon, “A new multichannel, coherent amplitude modulated, gate array technology implementation,” *IEEE Transactions on Signal Processing*, vol. 58, pp. 5369–5384, 2010.
- [8] J. Kim, H. Seungheon, and S. Choi, “Implementation of an SDR system using graphics processing unit,” *IEEE Communications Magazine*, vol. 48, no. 3, pp. 156–162, 2010.
- [9] U. Meyer-Baese, *Digital Signal Processing with Field Programmable Gate Array*. Springer, 2001.
- [10] Xilinx. (2015) System Generator for DSP. [Online]. Available: <http://www.xilinx.com/products/design-tools/vivado/integration/sysgen.html>
- [11] Actualités de Rohde & Schwarz, “Radios logicielles - vue d’ensemble et aspects matériels,” Brochure, 2004. [Online]. Available: [http://cdn.rohde-schwarz.com/pws/dl\\_downloads/dl\\_common\\_library/dl\\_news\\_from\\_rs/182/n182\\_radiocomunit\\_fr.pdf](http://cdn.rohde-schwarz.com/pws/dl_downloads/dl_common_library/dl_news_from_rs/182/n182_radiocomunit_fr.pdf)
- [12] V. Bhatnagar, G.S. Ouedraogo, M. Gautier, A. Carer, and O. Sentieys, “An FPGA software defined radio platform with a high-level synthesis design flow,” in *IEEE International Vehicular Technology conference (VTC’2013-Spring)*, 2013.
- [13] A.S. Rodriguez, M.C. Mensinger, A.I. Soo, and L. Yufeng, “Model-based software-defined radio (SDR) design using FPGA,” 2011.
- [14] ARTiSAN Software Tools, “ARTISAN REAL-TIME STUDIO SUPPORT FOR MODEL DRIVEN ARCHITECTURE (MDA),” ARTiSAN Software Tools, Whitepaper, 2002. [Online]. Available: [http://www.omg.org/mda/mda\\_files/ArtisanRealtimestudio.pdf](http://www.omg.org/mda/mda_files/ArtisanRealtimestudio.pdf)

- [15] Hoffmann, “Harmony-SE/SysML Deskbook : Model-Based Systems Engineering with Rhapsody,” Telelogic/I-Logix, Whitepaper, 2006.
- [16] MathWorks. (2015) Model-Based Design. [Online]. Available: <http://www.mathworks.com/help/simulink/gs/model-based-design.html>
- [17] Synopsys. (2015) Synplify DSP. [Online]. Available: <http://www.synopsys.com/Support/Training/Pages/SynplifyDSP.aspx>
- [18] *DSP Builder User Guide*, Altera Corporation, November 2009. [Online]. Available: [https://www.altera.co.jp/ja\\_JP/pdfs/literature/ug/ug\\_dsp\\_builder.pdf](https://www.altera.co.jp/ja_JP/pdfs/literature/ug/ug_dsp_builder.pdf)
- [19] MathWorks. (2015) HDL Coder: Generate Verilog and VHDL code for FPGA and ASIC designs. [Online]. Available: <http://www.mathworks.com/products/hdl-coder>
- [20] *Catapult C Synthesis*, Mentor Graphics, 2010. [Online]. Available: <http://pdf.directindustry.com/pdf/mentor-graphics/catapult-c-synthesis/60189-143056.html>
- [21] *Handel-C Language Reference Manual*, Celoxica, 2005. [Online]. Available: <http://babbage.cs.qc.edu/courses/cs345/Manuals/HandelC.pdf>
- [22] *ModelSim User’s Manual*, Mentor Graphics, 2012. [Online]. Available: [http://www.microsemi.com/document-portal/doc\\_view/131619-modelsim-user](http://www.microsemi.com/document-portal/doc_view/131619-modelsim-user)
- [23] *Incisive Enterprise Simulator : Multi-language simulation for low-power, metric-driven, mixed-signal verification*, Cadence, 2011. [Online]. Available: [http://www.europractice.stfc.ac.uk/vendors/cadence\\_incisive\\_enterprise\\_sim.pdf](http://www.europractice.stfc.ac.uk/vendors/cadence_incisive_enterprise_sim.pdf)
- [24] M. D. Ausint and G. L. Stüber, “Velocity adaptive handoff algorithms for microcellular systems,” *IEEE Transactions on Vehicular Technology*, vol. 43, no. 3, pp. 549–561, 1994.
- [25] G. Park , D. Hong, and C. Kang, “Level crossing rate estimation with Doppler adaptive noise suppression technique in frequency domain,” in *IEEE International Vehicular Technology conference (VTC’2003-Fall)*. IEEE, 2003, pp. 1192–1195.
- [26] K. Baddour and N. C. Beaulieu, “Robust Doppler spread estimation in nonisotropic fading channels,” *IEEE Transactions on Wireless Communications*, vol. 4, no. 6, pp. 2677–2682, 2005.
- [27] O. Mauritz, “A hybrid method for Doppler spread estimation,” in *IEEE International Vehicular Technology conference (VTC’2004-Fall)*. IEEE, 2004, pp. 962–965.
- [28] S. Mohanty, “VEPSD: Velocity estimation using the PSD of the received signal envelope in next generation wireless systems,” *IEEE Transactions on Wireless Communications*, vol. 4, no. 6, pp. 2655–2660, 2005.
- [29] . Krasny, H. Arslan, D. Koilpillai, and S. Chennakeshu, “Doppler spread estimation in mobile radio systems,” *IEEE Communication Letters*, vol. 5, no. 5, p. 197–199, 2001.
- [30] F. Bellili, R. Meftehi, S. Affes, and A. Stephenne, “Maximum Likelihood SNR Estimation of Linearly-Modulated Signals over Time-Varying Flat-Fading SIMO Channels,” *IEEE Transaction on Signal Processing*, vol. 63, pp. 441–456, 2014.
- [31] *Nutaq PicoSDR: FPGA-based, MIMO-Enabled tunable RF SDR solutions*, Nutaq, 2014. [Online]. Available: [http://nutaq.com/sites/default/files/PicoSDR\\_V1.4\\_02\\_14\\_2014\\_web.pdf](http://nutaq.com/sites/default/files/PicoSDR_V1.4_02_14_2014_web.pdf)
- [32] *PicoSDR User’s Guide*, Nutaq, 2013.
- [33] *Nutaq Perseus 601X: Virtex-6 AMC with FMC site*, Nutaq, 2012. [Online]. Available: [http://www.nutaq.com/sites/default/files/Perseus601x\\_Defense\\_Nutaq\\_LowRes.pdf](http://www.nutaq.com/sites/default/files/Perseus601x_Defense_Nutaq_LowRes.pdf)
- [34] *Nutaq Radio420X: Multimode SDR FMC RF transceiver*, Nutaq, 2014. [Online]. Available: [http://nutaq.com/sites/default/files/Radio420x\\_V1.3\\_03\\_24\\_2014\\_web.pdf](http://nutaq.com/sites/default/files/Radio420x_V1.3_03_24_2014_web.pdf)



- [35] *LMS6002D: Multi-band Multi-standard Transceiver with Integrated Dual DACs and ADCs*, Lime Microsystems, 2012. [Online]. Available: <http://www.limemicro.com/download/LMS6002Dr2-DataSheet-1.2r0.pdf>
- [36] Nutaq. Board Support Package (BSP). [Online]. Available: <http://nutaq.com/en/software/board-support-package>
- [37] *Nutaq MicroTCA BSP API*, Nutaq, 2013.
- [38] Nutaq. Board Software Development Kit (BSDK). [Online]. Available: <http://nutaq.com/en/software/board-software-development-kit>
- [39] Nutaq. Model-Based Design Kit (MBDK). [Online]. Available: <http://nutaq.com/en/software/model-based-design-kit>
- [40] *Propsim Quick Guide*, Anite Telecoms, 2013. [Online]. Available: [http://www.gigacomp.de/pdfs/Anite\\_Propsim\\_Quick\\_guide.pdf](http://www.gigacomp.de/pdfs/Anite_Propsim_Quick_guide.pdf)