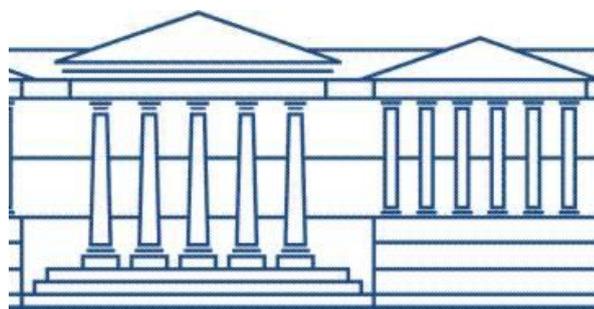


Mixed-Signal BISC Time-Interleaved ADC



UFPR

UNIVERSIDADE FEDERAL DO PARANÁ

Département d'ingénierie électrique – UFPR
Laboratoire du GICS
Stage du 27 Février au 31 Août 2011

Etudiant : Damien Benoit LUCAS

Tuteur de stage : Dr. François RIVET, Professeur associé

Maître de stage : Dr. André Augusto MARIANO, Professeur associé

Remerciements

Ces travaux de recherche ont été effectués au sein du laboratoire du GICS (Group of Integrated Circuits & Systems) de l'université fédérale du Paraná à Curitiba, Brésil.

Je tiens à remercier le Professeur André Mariano, mon maître de stage, pour m'avoir proposé ce stage et pour toute la confiance qu'il m'a témoignée en m'accueillant dans son laboratoire. Je le remercie également pour son aide et ce toujours dans la bonne humeur.

Je remercie aussi tout particulièrement Conrado Kamienski Mesadri avec qui il a été agréable de travailler durant ces 6 mois et qui m'a apporté ses connaissances dans le domaine de la conversion A/N.

J'adresse également mes remerciements au Professeur François Rivet, mon tuteur de stage d'avoir veillé au bon déroulement de celui-ci.

Enfin, je souhaite remercier tout le personnel du laboratoire et les doctorants que j'ai pu côtoyer, plus particulièrement Rodrigo Andriolo et Rodolfo Ronque étudiants tous deux en master pour leur accueil, leur disponibilité et leur gentillesse qu'ils ont su m'apporter pendant ces 6 mois passés avec eux.

Table des matières

INTRODUCTION.....	8
I. PRÉSENTATION DU LIEU DE TRAVAIL.....	9
I.1. UNIVERSITE FEDERALE DU PARANA "UFPR"	9
I.2. LE DEPARTEMENT D'INGENIERIE ELECTRONIQUE.....	9
I.2.1. <i>Histoire.....</i>	10
I.2.2. <i>Organisation</i>	11
I.2.3. <i>Recherche et développement.....</i>	11
I.3. LE GICS	12
II. PROBLEMATIQUE DU STAGE	13
II.1. CONTEXTE DE L'ETUDE.....	13
II.1.1. <i>Emergence de la radio logicielle</i>	13
II.1.2. <i>Problématique de la numérisation des données.....</i>	13
II.2. SUJET DE STAGE.....	16
II.2.1. <i>Etat de l'art.....</i>	16
II.2.2. <i>Objectif du stage.....</i>	16
II.3. PLANIFICATION DU PROJET	16
III. LES CONVERTISSEURS A/N ENTRELACES DANS LE TEMPS	18
III.1. LA CONVERSION ANALOGIQUE NUMERIQUE.....	18
III.1.1. <i>Caractéristiques fonctionnelles des convertisseurs A/N</i>	18
III.1.2. <i>Paramètres d'erreurs des ADC.....</i>	20
III.1.3. <i>Critère de performance.....</i>	22
III.1.4. <i>Architectures des ADC.....</i>	24
III.2. PRINCIPE DU TIADC	26
III.3. ETUDES DES ERREURS DE DISPARITE.....	27
III.4. MODELISATION DU TIADC SOUS MATLAB-SIMULINK	29
III.4.1. <i>Choix des paramètres des ADC et de simulation.....</i>	29
III.4.2. <i>Modélisation d'un ADC</i>	30
III.4.3. <i>Modélisation du système global TIADC</i>	32
IV. METHODE DE CORRECTION DES ERREURS	34
IV.1. PRESENTATION.....	34
IV.1.1. <i>Notation.....</i>	34
IV.1.2. <i>Algorithme de calibration</i>	34
IV.1.3. <i>Correction du signal.....</i>	36
IV.2. SIMULATION ET VALIDATION SOUS MATLAB-SIMULINK	38
IV.2.1. <i>Impact du nombre d'échantillons sur la précision de l'estimation des erreurs.....</i>	38
IV.2.2. <i>Correction des erreurs.....</i>	39
V. IMPLEMENTATION SUR FPGA.....	43
V.1. MISE EN PLACE DU TEST DE NOTRE METHODE	43
V.2. DECOMPOSITION DU TRAVAIL EN MODULES ET FONCTIONS.....	45
V.3. SIMULATIONS	46
VI. CONCLUSION	47
VII. BIBLIOGRAPHIE	49
VIII. ANNEXES	50
VIII.1. CARACTERISTIQUES DES CONVERTISSEURS ANALOGIQUE-NUMERIQUE	50
VIII.2. SIMULATION AVEC UN SIGNAL MODULE EN AMPLITUDE	50

Table des illustrations

Figure 1 - Bâtiment du département d'ingénierie électrique	9
Figure 2 - Partenaires du département électrique de l'UFPR	11
Figure 3 - Organisation des Laboratoires de recherche du département électronique	12
Figure 4 - Partenaires du GICS	12
Figure 5 - Principe de la radio logicielle idéal	13
Figure 6 - Classement de diverses architectures CAN en fonction de la résolution et de la fréquence	14
Figure 7 - Radio logicielle restreinte	14
Figure 8 - Principe de la conversion par entrelacement temporel	15
Figure 9 - Effet des erreurs d'offset, gain et Clock Skew sur le spectre d'un TIADC	15
Figure 10 - Architecture de calibration du TIADC avec des séquences orthogonales	16
Figure 11 - (i) signal analogique (ii) signal échantillonné (iii) puis quantifié	18
Figure 12 - Modèle SIMULINK de l'échantillonneur-bloqueur	18
Figure 13 - Caractéristique de transfert idéale d'une loi de quantification uniforme par arrondi (ADC de 3 bits)	19
Figure 14 - Erreur de codage de la quantification uniforme par défaut	20
Figure 15 - Erreur de codage de la quantification uniforme par arrondi	20
Figure 16 - (i) Erreur d'offset (ii) Erreur de gain	21
Figure 17 - SFDR Spurious Free Dynamic Range	23
Figure 18 - Principe de la conversion par entrelacement temporel avec correction d'erreur	26
Figure 19 - Modèle SIMULINK de l'ADC1	30
Figure 20 - Spectre du signal numérisé par l'ADC1	31
Figure 21 - Modèle SIMULINK du TIADC ' <i>TIADC_sim</i> '	32
Figure 22 - Spectre du signal numérisé par le TIADC	33
Figure 23 - Méthode d'interpolation utilisée pour corriger l'erreur de Δt_k	37
Figure 24 - % Erreur d'estimation de l'offset, du gain et du Clock Skew en fonction du nombre d'échantillons	38
Figure 25 - Spectre du signal TIADC non corrigé puis corrigé en connaissant la fréquence d'entrée	39
Figure 26 - Spectre du signal TIADC (avec Δt_k) non corrigé puis corrigé par interpolation	40
Figure 27 - Spectre du signal TIADC (avec G_k , O_k et Δt_k) non corrigé puis corrigé par interpolation	40
Figure 28 - Impact de la fréquence d'entrée du TIADC (avec Δt_k) sur le SFDR du signal numérisé	41
Figure 29 - Spectre du signal TIADC (avec G_k , O_k et Δt_k) non corrigé puis corrigé par interpolation	41
Figure 30 - Schéma bloc de test de la méthode de correction implémenté sur FPGA	43
Figure 31 - Modèle des ADC utilisés pour réaliser le test sur FPGA	43
Figure 32 - Calibration du TIADC avec 64 échantillons	44
Figure 33 - Schéma simplifié du fonctionnement de notre système sur FPGA	45

Notations

Δt_k	erreur de retard d'ouverture
A	amplitude du signal d'entrée
$ENOB$	effective number of bit
f_0	fréquence du signal d'entrée
f_s	fréquence d'échantillonnage
G_k	erreur de gain
M	nombre de convertisseurs entrelacés
N	résolution du convertisseur
N_{ech}	nombre d'échantillons pris par chaque ADC pour estimation et correction
N_{fft}	nombre de points de la FFT
O_k	erreur d'offset
$SINAD$	signal-to-noise and distortion ratio
$SFDR$	spurious-free dynamic range
SNR	signal-to-noise ratio
T_0	période du signal d'entrée
T_s	période d'échantillonnage

Acronymes

ADC	Analog to Digital Converter (convertisseur analogique numérique)
CIEL	Centro de Instrumentação Eletrônica
COPEL	COmpania Paranaense de Energia
DAC	Digital to Analog Converter (convertisseur numérique analogique)
FFT	Fast Fourier Transform : (transformé de Fourier rapide)
FPGA	Field Programmable Gate Array
GICS	Group of Integrated Circuits and Systems
LAMMI	Lab. Microeletrônica, Medidas e Instrumentação
LASICO	Lab. De Simulação, Controle e Otimização
LIEC	Lab. Instrumentação Eletrônica e Comunicação
LUT	Look Up Table
MATLAB	MATrix LABoratory
RTL	Register Transfert Level
SOC	System On Chip
TELEPAR	TELEfonia do PARaná
TIADC	Time-Interleaved Analog to Digital Converter : (convertisseur analogique numérique entrelacé dans le temps)
UFPR	Universidade Federal do Paraná (PR)
VHDL	Very High speed integrated circuit hardware Description Language
VHDL-AMS	Very High speed integrated circuit hardware Description Language Analog and Mixed Signal

INTRODUCTION

Les applications de radiocommunication connaissent une véritable explosion. L'émergence des futures générations de systèmes, notamment la quatrième génération, se présente comme une réponse à l'accroissement permanent de la diversité des standards et des applications demandées et supportées. Cette évolution entraîne plus de translation en fréquence et donc plus de problème au niveau des mixers (plage de fonctionnement, injection de bruit, ...). Il semble donc intéressant d'annuler ce passage par les mixers en rapprochant le convertisseur analogique numérique le plus possible de l'antenne. Cela permet aussi de supprimer d'autres fonctions analogiques telles que les amplificateurs ou filtres. Ces fonctions peuvent être facilement implémentées dans le domaine numérique, réduisant ainsi la complexité et donc la consommation. On se rapproche alors de la notion de radio logicielle (*Software Radio*) [1] qui fait référence à un ensemble de techniques qui permet la reconfiguration de tout système de communication sans modifier l'architecture matérielle.

Cependant, la puissance consommée et la miniaturisation du convertisseur analogique-numérique sont des points critiques pour les récepteurs multistandard intégrés sur un terminal mobile. Ces besoins entraînent donc de grands efforts dans le design d'un ADC (*Analog-to-Digital Converter*) de haute résolution et de haute fréquence de fonctionnement.

Les solutions permettant d'augmenter la fréquence d'échantillonnage de l'ADC, sans une augmentation conséquente de la puissance consommée, suscitent donc de plus en plus d'intérêt. L'une d'entre elles est l'entrelacement temporel des ADC connu sous le nom TIADC (*Time-Interleaved Analog to Digital Converter*) [2] et [3]. La fréquence d'échantillonnage augmente proportionnellement avec le nombre d'ADC mais des problèmes de disparité entre les erreurs des différentes ADC réduisent la résolution effective du TIADC.

Ce rapport présente ainsi une recherche sur la correction de ces erreurs aussi bien analogiques que numériques. Je commence tout d'abord par une présentation du laboratoire 'GICS' où j'ai réalisé ces recherches, puis je détaille le sujet de mon stage en expliquant le cahier des charges. Une étude plus détaillée du fonctionnement des ADC et du problème de la disparité de leurs erreurs lors de l'entrelacement temporel est détaillé dans une deuxième partie. Ensuite je développe l'étude théorique et pratique qui permet de résoudre les problèmes d'erreurs du TIADC, et enfin je conclus sur le travail réalisé et les améliorations à apporter.

Mots clefs : radio logicielle, récepteur multistandard, récepteur large bande, ADC, TIADC, algorithme d'estimation d'erreur, méthodes de calibration TIACD, FPGA.

I. PRÉSENTATION DU LIEU DE TRAVAIL

I.1. Université Fédérale du Paraná "UFPR"

L'Université Fédérale du Paraná est la plus ancienne université du Brésil et est un symbole de Curitiba. Enveloppée d'une histoire de luttes et de conquêtes depuis 1912, l'UFPR est la référence dans l'enseignement supérieur pour l'Etat et pour le Brésil.

Symbole majeur de l'*inteligência paranense*, l'université montre son importance et son excellence à travers les licences, spécialisations, masters et doctorats qui sont guidés par le principe d'indissociabilité entre l'enseignement, la recherche et l'activité d'extension. L'extension est une activité pratiquée par les universités fédérales du Brésil. Elèves et professeurs mettent leurs connaissances à profit de la société en développant des projets qui l'aideront. Un exemple de cette activité d'extension est le PET (programa educational tutorial) qui est une association dirigée par des étudiants recrutés sur concours et qui reçoivent une petite rémunération du gouvernement. Un des objectifs de cette association est d'apporter des compléments sur les enseignements suivis au département.

La fonction sociale de l'université est valorisée par le biais du trio enseignement, recherche et activité d'extension. De plus, ces trois actions doivent permettre à cette université, sous forme de connaissance, de technologie et de culture, d'avoir les ressources publiques qui lui permettent d'exister en tant qu'Institution Fédérale de l'Enseignement Supérieur.

Au-delà des campi à Curitiba, l'UFPR est présente dans la province paranense (du Paraná) et la côte de l'État fédéral, avec comme rôle actif de développeur socio-économique et de la qualité de vie des paranenses (habitants du Paraná) par le biais de l'accès à l'enseignement supérieur et des activités développées par la communauté académique en faveur de la société du Paraná et du Brésil.

I.2. Le département d'ingénierie électronique

Ce département qui regroupe 49 enseignants-chercheurs et environ 800 étudiants est en phase d'agrandissement avec notamment la construction d'un nouveau bâtiment permettant d'accueillir plus d'élèves.



Figure 1 - Bâtiment du département d'ingénierie électrique

I.2.1. Histoire

Le Département d'Electricité (actuellement Département d'Ingénierie Electrique) de l'Université Fédérale du Paraná fut créé en 1966, conjointement avec la mise en place de la Licence en Ingénierie Electrique. Ce cours, pionnier dans l'état, est né d'une demande du gouvernement de l'état, menée principalement par la COPEL – Compagnie Paranaense de l'Energie, qui a commencé à l'époque un remarquable effort visant à l'électrification de l'état du Paraná. A cette époque, le Paraná était considéré comme l'un des états brésiliens les plus mal desservis en énergie électrique, qui servait seulement 20% des habitants. Une grande partie de l'installation électrique paranaense était composée de centrales diesel-électriques appartenant à des particuliers ou à des préfectures municipales, qui étaient en grande partie dans un état déplorable. Les coupes d'énergie électrique pouvaient durer des jours entiers et la fiabilité des services était en-dessous du tolérable. Malgré le fait qu'elle fut fondée en 1958, ce n'est que dans les années 60 que la COPEL a commencé à fonctionner de manière efficace, assumant les fonctions de la compagnie *Forza e Luz do Paraná*, de capital étranger, qui desservait la région métropolitaine de Curitiba.

En 1976, le cours d'ingénierie électrique de l'UFPR a commencé à offrir, en plus de l'électrotechnique, une spécialité en télécommunications. Une fois de plus, cette décision fut prise en accord avec l'état économique de l'état, motivée par l'expansion du système téléphonique du Paraná. La TELEPAR (ancien nom de la compagnie de télécommunications Oi) a soutenu la création de cette spécialité dans le cours de l'UFPR, avec pour but d'obtenir des ingénieurs spécialisés afin de promouvoir l'expansion de ses services. Le cours d'ingénierie électrique de l'UFPR fut le premier de la région sud à offrir une formation dans le domaine de l'ingénierie des télécommunications et fut responsable de la formation de la majorité des ingénieurs en télécommunications de la TELEPAR. Après 1996, avec l'ouverture du marché des télécommunications du Brésil à l'initiative privée, d'autres entreprises de prestation de services apparurent dans ce domaine, qui ont également été se fournir en personnel provenant du cours d'ingénierie électrique de l'UFPR.

En 1982, le cours d'Ingénierie Electrique de l'UFPR a ajouté la spécialité en Electronique à ses options. Poursuivant son duo avec le panorama socio-économique local, cette décision fut basée sur la demande croissante d'ingénieurs spécialisés pour les industries qui commencèrent à s'installer dans la zone industrielle de Curitiba. La multinationale Siemens, avec ses partenaires brésiliens de l'époque, a établi la société Equitel à Curitiba, usine d'équipement de télécommunications, suivie par la société japonaise Furukawa avec une usine de câbles téléphoniques. Conjointement avec d'autres entreprises nationales et étrangères, le scénario local dessine un avenir d'industrialisation rapide pour le Paraná, qui a nécessité la formation de professionnels en électricité avec un profil différent de celui jusqu'alors formé. Le cours d'Ingénierie Electrique de l'UFPR a réorganisé son programme d'études à cette occasion et est devenu l'un des principaux fournisseurs de main-d'œuvre qualifiée pour les entreprises qui s'installèrent dans le Paraná. En quelques années, le profil socio-économique du Paraná a présenté un changement vers une base industrielle dans laquelle les ingénieurs formés par l'UFPR ont agi dans plusieurs domaines, contribuant ainsi à la croissance rapide des industries installées dans la région de Curitiba.

Le département électrique travaille aujourd'hui avec de nombreux partenaire. (cf. Fig. 2)



Figure 2 - Partenaires du département électrique de l'UFPR

I.2.2. Organisation

Comme vu précédemment les trois principaux domaines d'activité du département sont les suivants :

- **Energie**
 - Système de potentiel
- **Electronique**
 - Circuit d'instrumentation électronique
 - Contrôle et automatique
- **Télécommunications**
 - Electromagnétisme appliqué
 - Systèmes de communications

En 2009, l'événement de grande importance fut la création du programme nocturne du Cours d'Ingénierie Electrique. En plus de continuer à offrir, durant la journée, les spécialités de la formation en électrotechnique et en électronique-télécommunications, le groupe chargé du projet de l'équipe de nuit a constaté la nécessité d'un nouveau profil de professionnels dans la région de Curitiba et dans tout le Brésil. Vint alors la spécialité en systèmes électroniques embarqués, offerte d'une manière sans précédent dans le pays dans le programme nocturne du Cours d'Ingénierie Electrique de l'UFPR. L'offre de la spécialité en Systèmes Electroniques Embarqués au programme nocturne du Cours d'Ingénierie Electrique de l'UFPR représente l'opportunité d'une formation dans un domaine de haute technicité, avec une forte demande de professionnels et de la plus haute importance dans la scène électronique du Brésil.

I.2.3. Recherche et développement

Le département d'ingénierie électrique est composé du CIEL (Centro de Instrumentação Eletrônica) comprenant 4 laboratoires (LIEC, LAMMI, GICS et LASICO).

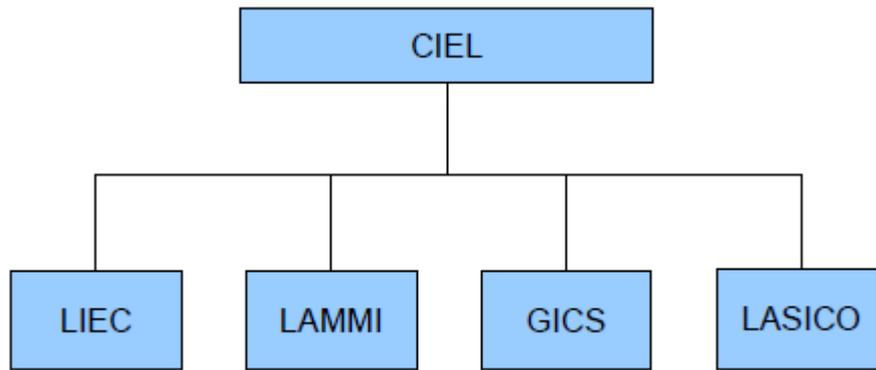


Figure 3 - Organisation des Laboratoires de recherche du département électronique

En ce qui me concerne j'ai travaillé dans la partie des circuits intégrés qui appartient au laboratoire du GICS.

I.3. Le GICS

Le GICS est dirigé par les professeurs Oscar Gouveia Filho et André Mariano, qui est responsable des élèves français en Projet de Fin d'Etudes. L'un des domaines de recherche du laboratoire est l'intégration de circuits et systèmes en technologie VLSI en partenariat avec les entreprises et laboratoires suivants.



Figure 4 - Partenaires du GICS

Le laboratoire travaille aussi dans les domaines suivants : Near Field Communication (NFC), compensation de variation dans le processus de fabrication, dispositifs mobiles téléphoniques, Software Defined Radio (SDR), communication sans fil et Radar mobile de nouvelle génération.

II. Problématique du stage

II.1. Contexte de l'étude

II.1.1. Emergence de la radio logicielle

A cette époque, l'enjeu technique et économique consiste à pouvoir émettre et recevoir des informations de différents standards utilisant le même terminal. En effet, les différents standards existants, ne constituent pas des offres concurrentes mais une unique offre avec une multitude de services. Ces différents services mis en jeu ne doivent pas mettre en cause la couche physique qui constitue l'investissement le plus important du réseau. D'un point de vue technologique, le terminal universel du réseau de télécommunications mobiles de demain doit traiter des signaux de fréquence, d'amplitude, de bande passante et de modulation différentes.

La radio logicielle [1] est la solution théorique permettant de répondre à ces évolutions en donnant la possibilité d'utiliser la même infrastructure matérielle et de sélectionner le standard de manière logicielle. L'utilisateur n'aura qu'à sélectionner le service souhaité (voix, vidéo, visiophonie, ...). Le terminal communique avec la station de base pour télécharger le logiciel approprié au standard de communications et réserver les ressources matérielles et logicielles nécessaires à la communication.

Les objectifs commerciaux consistent donc à proposer différentes applications autour d'un matériel standard évitant ainsi un empilement de chaînes de traitements dédiées chacune à un standard particulier. L'intérêt se trouve aussi pour les stations de base où on cherche à diminuer l'empilement des récepteurs radiofréquences permettant ainsi le traitement de plusieurs canaux simultanément. Pour une station de base GSM (*Global System for Mobile communications*) par exemple, chaque canal de transmission est numérisé par une chaîne de réception séparée. L'utilisation de plusieurs chaînes induit des pertes de la puissance consommée, de performance et du prix. Cette recherche est prioritaire dans le cas des stations de base dont les avantages sont absolus en termes d'intégration et de consommation.

Partant de cette idée très simple, la radio logicielle est devenue un domaine incontournable dans l'évolution des télécommunications.

II.1.2. Problématique de la numérisation des données

Du fait des meilleures performances des DAC par rapport aux ADC, la chaîne de réception présente a priori plus de difficultés de réalisation que la chaîne d'émission.

Idéalement, l'architecture du récepteur radio est composée de trois éléments : une antenne très large bande, un Convertisseur Analogique Numérique (*CAN ou ADC en anglais*) ultra rapide et large bande et un système de traitement numérique du signal. (cf. Fig. 5)

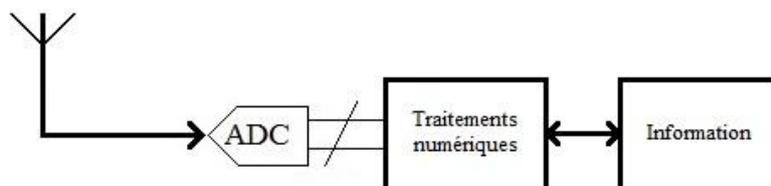


Figure 5 - Principe de la radio logicielle idéal

Afin d'augmenter les capacités de reprogrammation, l'idée principale consiste à placer l'ADC au plus proche de l'antenne pour réaliser un maximum de traitements en numérique. Les avantages visés sont la flexibilité, la reconfigurabilité et la portabilité de l'application. En effet, les principales limitations des composants analogiques sont liées essentiellement à la difficulté de reconfiguration, la sensibilité aux perturbations et la dégradation des performances. Cependant, il est important de signaler que les circuits numériques souffrent du fait qu'ils ne peuvent assurer à ce jour des traitements rapides aux fréquences radio, fluctuant et évoluant en fonction des progrès de l'électronique.

Dans le cadre de la radio logicielle idéale, les ADC doivent pouvoir répondre à différentes exigences, notamment une très haute fréquence de fonctionnement et une résolution d'environ 18 bits. Actuellement, les convertisseurs commercialisés présentent des caractéristiques variant entre 2.5 MHz pour 24 bits, 210 MHz pour 14 bits et 2.2 GHz pour 8 bits (cf. Fig. 6).

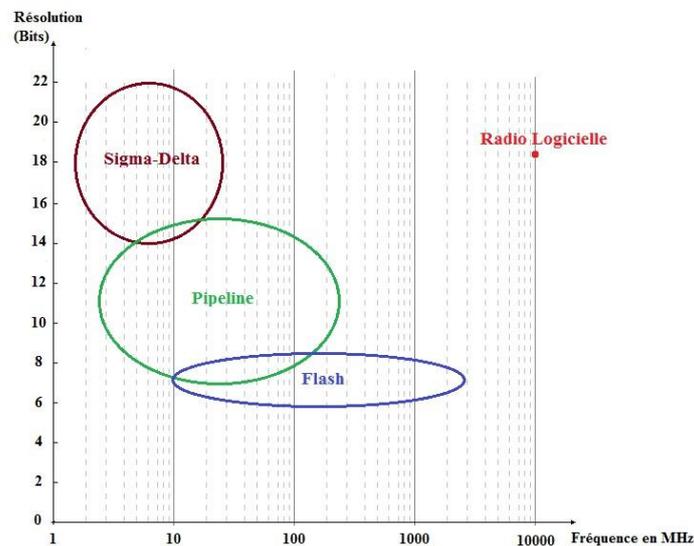


Figure 6 - Classement de diverses architectures CAN en fonction de la résolution et de la fréquence

Comme le montre la figure 6, la radio logicielle dite idéale est irréalisable dans l'avenir proche. Les convertisseurs imposent un partitionnement analogique numérique laissant encore la place à une partie analogique et nous parlons alors de radio logicielle restreinte (SDR, *Software Defined Radio*) [1]. (cf. Fig. 7)

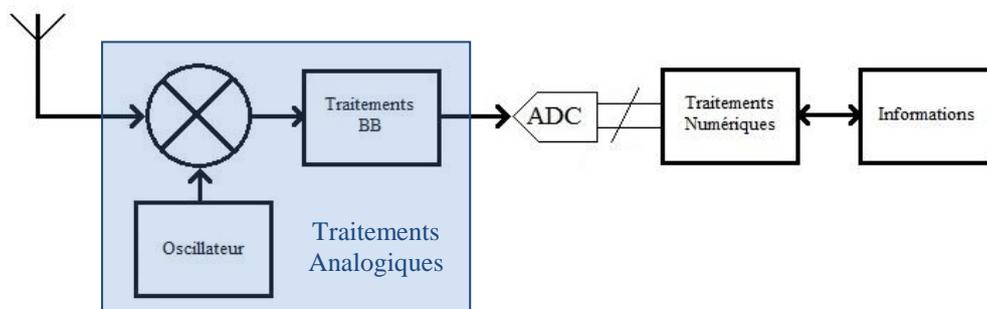


Figure 7 - Radio logicielle restreinte

L'objectif est de supprimer de plus en plus de passage par les mixeurs en augmentant les performances des ADC. Le principal verrou à cela est la proposition de nouvelles architectures de conversion analogique numérique assurant simultanément de meilleures performances en termes de rapidité, précision et consommation.

Un seul ADC classique ne peut pas garantir ces demandes. Une des solutions qui permet d'augmenter la fréquence d'échantillonnage est le TIADC (cf. Fig. 8). Il s'agit d'entrelacer dans le temps plusieurs ADC. La fréquence d'échantillonnage du système résultant est multipliée par le nombre d'ADC utilisés.

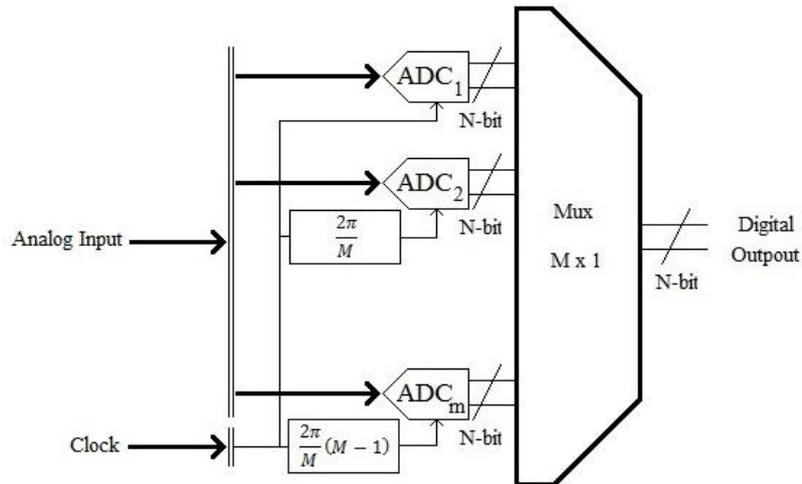


Figure 8 - Principe de la conversion par entrelacement temporel

Malheureusement, les ADC utilisés ne sont pas idéaux, chacun d'entre eux est caractérisé par ses propres erreurs engendrées principalement par les défauts du processus de fabrication. Les disparités entre les caractéristiques de transfert des différents ADC (cf. partie III.3) élémentaires causent l'apparition de raies spectrales parasites qui réduit la résolution effective du TIADC. (cf. Fig. 9)

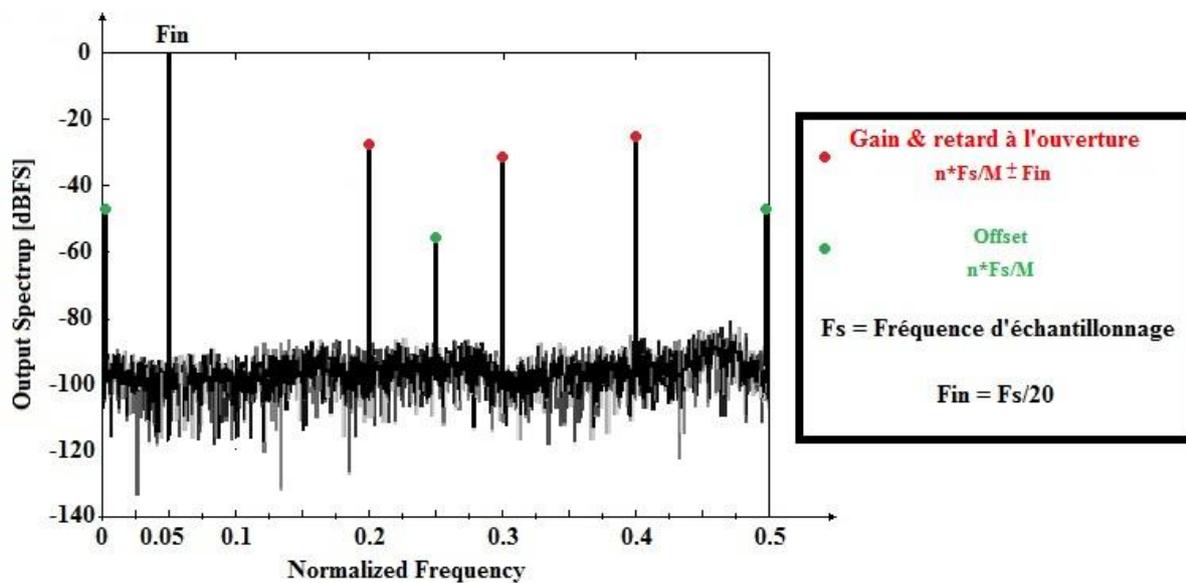


Figure 9 - Effet des erreurs d'offset, gain et Clock Skew sur le spectre d'un TIADC

II.2. Sujet de Stage

II.2.1. Etat de l'art

Une méthode intéressante pour corriger ce type d'erreurs est la suivante : On estime les erreurs d'offset, de gain et de phase pour chacun des ADC dans une architecture TIADC en utilisant des séquences orthogonales. Son principe qui se base sur l'utilisation de séquences orthogonales est largement utilisé dans les communications et notamment en CDMA. Un algorithme d'estimation et de correction des erreurs a été développé [2] puis le principe de fonctionnement étant validé par MATLAB, l'étape suivante s'est portée sur une implémentation en VHDL-AMS pour vérifier la faisabilité de l'implémentation de l'algorithme au niveau système où la partie analogique est décrite en langage comportemental VHDL-AMS et la partie logique en VHDL [3].

Cette méthode fonctionne correctement, cependant la partie multiplication analogique par les séquences orthogonales introduit elle aussi d'autres erreurs. (cf. Fig. 10)

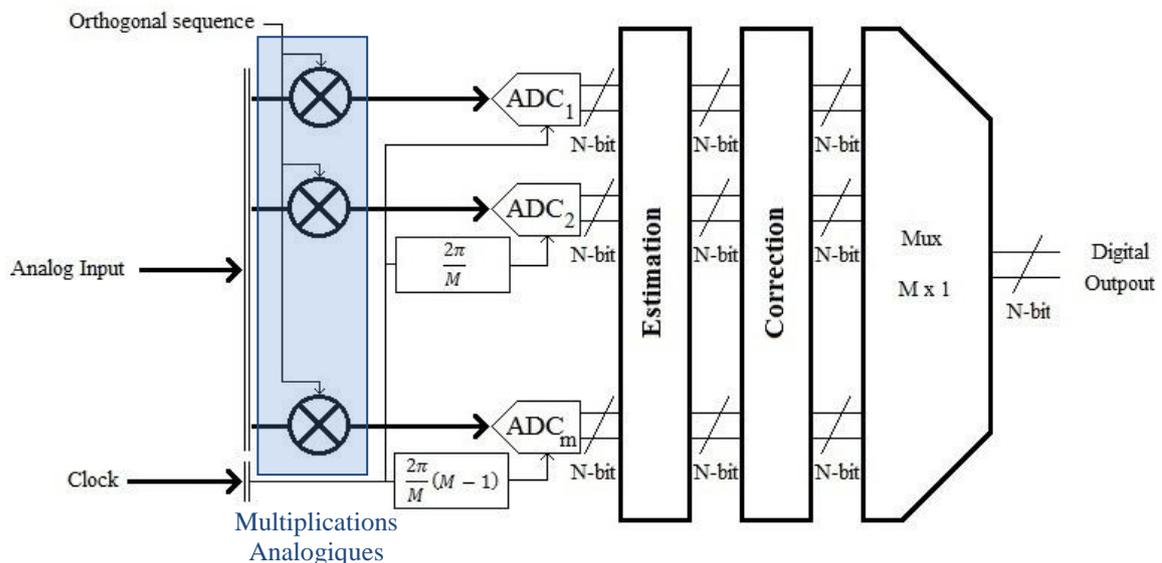


Figure 10 - Architecture de calibration du TIADC avec des séquences orthogonales

II.2.2. Objectif du stage

Ainsi l'objectif du stage est de trouver un moyen de corriger les erreurs introduites par le système de correction orthogonale.

Pour cela nous allons modéliser un TIADC comprenant toutes les erreurs de chaque ADC et celles introduites par la multiplication analogique. Ce modèle sera fait sous Matlab-SIMULINK afin de permettre un temps de simulation beaucoup plus rapide que celui proposé dans [3] en VHDL-AMS et ainsi faciliter la recherche d'une solution pour corriger la partie analogique du système.

II.3. Planification du projet

Lire des publications concernant l'utilisation de séquences orthogonales dans le calibrage d'un TIADC, bien comprendre la théorie et en tirer les informations nécessaires à l'avancement du projet. Les deux publications principales sont [2] et [3].

Valider la théorie de l'estimation des erreurs d'offset, de gain et de phase en réalisant un modèle SIMULINK 'TIADC_sim' qui considère la partie multiplication analogique idéale.

Retrouver la structure analogique issue de la publication [3] (utilisation de CADENCE) et définir un modèle mathématique avec variations des niveaux haut et bas (différents de +1, -1) en analogique puis réalisation d'un modèle SIMULINK 'MultReel_sim' de cette structure.

Implantation de 'MultReel_sim' dans le modèle SIMULINK et étude de l'influence de ses défauts.

Etude de la répercussion de ses défauts analogiques sur l'algorithme d'estimation des erreurs et mise en place d'une solution.

Le Tableau 1 montre les objectifs fixés et le travail réellement réalisé.

Mois	Mars			Avril			Mai			Juin			Juillet			Aout																
Semaines	S 1	S 2	S 3	S 4	S 5	S 6	S 7	S 8	S 9	S 10	S 11	S 12	S 13	S 14	S 15	S 16	S 17	S 18	S 19	S 20	S 21	S 22	S 23	S 24	S 25	S 26	S 27	S 28	S 29	S 30	S 31	
Revue de la littérature	OK																															
Modélisation TIADC																																
Modélisation séquenceur																																
Correction des erreurs dues au séquenceur																																
Développement d'un algorithme sans séquenceurs																																
Test sur FPGA																																

Tableau 1 - Diagramme de GANT

Nous pouvons voir alors qu'en cour de stage certains objectifs ont changé. En effet j'ai arrêté en cour de route la modélisation du séquenceur car j'ai réussi à modifier l'algorithme pour ne plus avoir besoin d'utiliser les séquences orthogonales. De ce fait le problème des erreurs analogiques introduites en plus a été réglé et je me suis concentré sur la partie test. Le but étant de réaliser un système TIADC en utilisant la modélisation du TIADC sous MATLAB-SIMULINK puis en envoyant les données sur une carte FPGA qui elle se charge de mettre en œuvre l'algorithme d'estimation et de correction des erreurs.

III. Les convertisseurs A/N entrelacés dans le temps

III.1. La conversion analogique numérique

III.1.1. Caractéristiques fonctionnelles des convertisseurs A/N

L'opération de conversion analogique numérique consiste à transformer un signal continu dans le temps en un signal discrétisé qui se propage dans des circuits numériques. Ces signaux numériques sont une suite de mots binaires régulièrement espacés dans le temps, ne prenant qu'un nombre fini de valeurs. Conceptuellement, la conversion analogique-numérique peut être divisée en deux étapes : *l'échantillonnage temporel* et la *quantification*. La figure 11 présente successivement ces trois étapes pour un DAC dont la sortie du signal numérique est sur 3 bits :

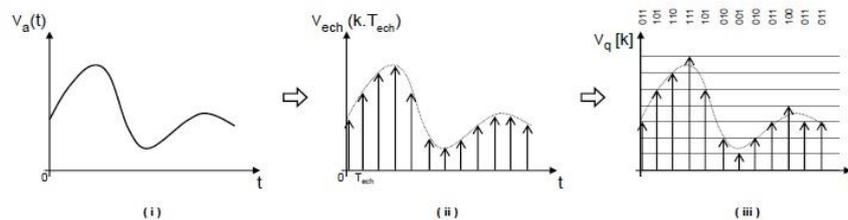


Figure 11 - (i) signal analogique (ii) signal échantillonné (iii) puis quantifié

III.1.1.1. L'échantillonnage

L'échantillonnage périodique permet de représenter un signal continu dans le temps avec un ensemble de valeurs discrètes $x(n)$ à des instants multiples de T_s , appelée période d'échantillonnage. Quand le signal continu est multiplié par un train d'impulsions périodiques nommé peigne de Dirac, nous obtenons à la sortie une version échantillonnée du signal d'entrée.

$$\begin{array}{ll} \text{Signal sinusoidal pur} & \text{Signal échantillonné} \\ x(t) = A \times \sin(2\pi f_0 t) & \rightarrow x(n) = A \times \sin(2\pi f_0 n T_s) \end{array} \quad (3.1)$$

D'un point de vue spectral, le processus d'échantillonnage affecte le spectre du signal échantillonné. Le spectre du signal échantillonné est alors périodique avec la fréquence d'échantillonnage. Cette périodicité surgit suite à l'opération de translation du spectre du signal analogique en bande de base à des multiples entiers de la fréquence d'échantillonnage. Afin que le signal reconstitué soit identique au signal d'origine, il faut satisfaire le critère de Nyquist. Ce critère stipule que pour qu'un signal de temps continu soit correctement reconstitué par ses échantillons, il doit être échantillonné avec une fréquence d'échantillonnage supérieure à deux fois la fréquence la plus grande du signal. Sinon, on serait face à un recouvrement du signal par les bandes images et la reconstitution du signal devient impossible.

La figure 12 montre le bloc d'échantillonnage utilisé sous SIMULINK pour notre modélisation de l'ADC. C'est sur ce block que l'erreur de clock skew intervient en modifiant le moment d'échantillonnage.

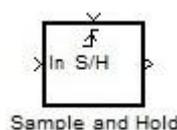


Figure 12 - Modèle SIMULINK de l'échantillonneur-bloqueur

III.1.1.2. La quantification

▪ Caractéristique de transfert

Le pas de quantification et la précision d'un ADC dépendent du nombre de bits en sortie, appelé **résolution**. Pour un ADC à N bits, le nombre d'états possibles en sortie est 2^N , ce qui permet d'exprimer des signaux numériques de 0 à 2^{N-1} en code binaire naturel. Un ADC est caractérisé également par la plage de variation acceptable de la tension analogique d'entrée, appelée **Pleine Echelle** (FS pour *Full Scale* en anglais) et que nous noterons V_{PE} .

La pleine échelle est divisée en autant de plages d'égales dimensions (cas de la quantification uniforme) qu'il y a d'états possibles de la sortie numérique. Chaque plage est associée à un code numérique représentant la tension analogique d'entrée. L'opération de quantification revient à appliquer au signal d'entrée une caractéristique de transfert en marche d'escalier, comme le montre la figure 13.

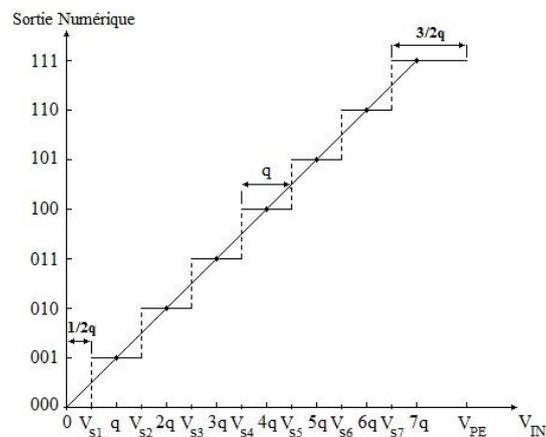


Figure 13 - Caractéristique de transfert idéale d'une loi de quantification uniforme par arrondi (ADC de 3 bits)

Le quantum q (pas de quantification) dépend de la résolution N et de la pleine échelle de conversion V_{PE} par la relation suivante :

$$q = \frac{V_{PE}}{2^N} \quad (3.2)$$

La dynamique d'entrée est le rapport entre la plus grande amplitude du signal d'entrée admissible par l'ADC et le pas de quantification. Par exemple dans notre modèle ou nous allons utiliser un convertisseur de résolution $N=14$ bits avec $V_{PE} = 2$, le quantum est de $1.2207e-004$ et la dynamique d'entrée est de 84dB.

▪ Erreur de quantification (ou de codage) : $e_q(n)$

Si le quantificateur était idéal et avec une résolution infinie, la caractéristique de transfert (figure xxx) serait une droite sur laquelle une équivalence serait faite entre chaque valeur analogique et le code de sortie. Mais, en réalité toute une plage de valeurs sera convertie en un seul nombre par l'utilisation d'une caractéristique en marche d'escalier. Ceci explique que la quantification, par les approximations utilisées, est un processus irréversible qui provoque une erreur systématique ne dépendant que du pas de quantification utilisé. Ainsi, pour un signal analogique à l'entrée de l'ADC exprimé par l'équation 3.1, la sortie du quantificateur idéal s'écrit sous la forme :

$$x(n) = A \times \sin(2\pi f_0 n T_s) + e_q(n) \quad (3.3)$$

La figure 14 donne l'erreur de codage d'un ADC de 3 bits pour une quantification uniforme par défaut.

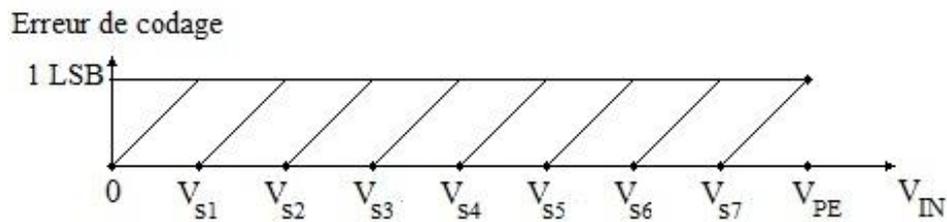


Figure 14 - Erreur de codage de la quantification uniforme par défaut

L'erreur de quantification est comprise entre 0 et 1 LSB. Ainsi, tous les signaux analogiques compris entre V_{S2} et V_{S3} , par exemple, sont représentés par le code binaire 010. Un simple changement de convention, dans la fixation des tensions de seuil, permet de réduire l'erreur de quantification en valeur absolue. (cf. Fig. 15)

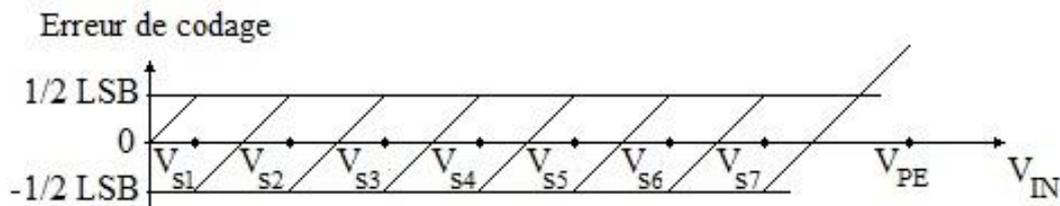


Figure 15 - Erreur de codage de la quantification uniforme par arrondi

Ainsi, on utilisera plutôt la quantification uniforme par arrondi, pour laquelle la droite de transfert idéale passe par le centre des "marches" de la caractéristique. (cf. Fig. 13)

L'erreur de quantification dépend non seulement de la fonction de transfert du CAN idéal, mais aussi du signal d'entrée considéré. Le bruit de quantification généré présente une densité spectrale uniforme dans la bande $[-fs/2, fs/2]$, $fs=1/Ts$ dont la valeur efficace est donnée par :

$$B_q = \frac{q}{\sqrt{12}} \quad (3.4)$$

Finalement, les équations (3.2) et (3.4) montrent que plus la résolution du ADC est importante, plus la plage analogique définie par le quantum q est réduite, diminuant de ce fait le bruit de quantification : une meilleure résolution engendre un faible bruit de quantification. L'objectif du TIADC étant d'avoir une haute résolution, le bruit de quantification ne sera pas un problème dans notre modélisation.

III.1.2. Paramètres d'erreurs des ADC

L'opération de la quantification induit une erreur systématique, mais un ADC est aussi caractérisé par ses paramètres d'erreurs qui sont liés à l'imperfection de l'électronique utilisée. En effet, plusieurs sources physiques de bruit au niveau des semi-conducteurs induisent des erreurs statiques et dynamiques dans les ADC.

III.1.2.1. Erreurs statiques

La description statique du quantificateur idéal (cf. Fig. 13) utilise une caractéristique indépendante du signal d'entrée. Dans le cas réel, cette caractéristique va s'écarter de sa position d'origine. La manière avec laquelle cet écart est aperçu définit la nature de l'erreur que nous exposons [4].

- Erreur d'offset :

On appelle offset un décalage horizontal de la caractéristique de transfert d'un ADC, l'erreur d'offset est exprimée usuellement en LSB (cf. Fig. 16). Il s'agit alors d'une constante additive au signal de sortie de l'ADC. Comparée à l'équation (3.1), la sortie de l'ADC en présence d'une erreur d'offset aura la forme suivante :

$$x(n) = A \times \sin(2\pi f_0 n T_S) + O_k \tag{3.5}$$

- Erreur de gain :

L'erreur de gain permet de mesurer l'écart entre la pente de la caractéristique idéale de transfert et la pente de la caractéristique réelle obtenue par régression linéaire des centres des paliers (cf. Fig. 16). Cette erreur peut altérer la pleine échelle du convertisseur et par la suite la valeur du quantum.

$$x(n) = AG_k \times \sin(2\pi f_0 n T_S) \tag{3.6}$$

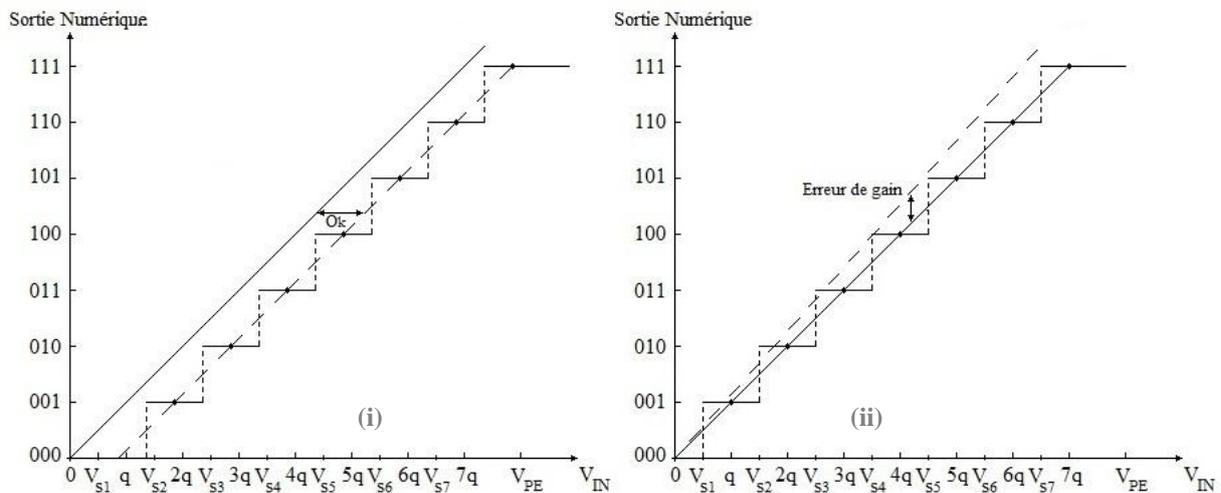


Figure 16 - (i) Erreur d'offset (ii) Erreur de gain

III.1.2.2. Erreurs Dynamiques

Contrairement aux erreurs statiques résultant du déplacement de la fonction du transfert de l'ADC, les erreurs dynamiques sont résultantes de sources supplémentaires d'erreurs induites par la variation du signal analogique [4].

- Gigue à l'ouverture :

La variation ou gigue d'ouverture est une erreur sur l'instant de prise et provoque une erreur appelée "jitter" assimilable à l'addition d'un bruit sur le signal.

- Incertitude à l'ouverture :

Le temps d'incertitude à l'ouverture correspond au retard entre le front actif du signal d'échantillonnage et la prise effective de l'échantillon par l'ADC sur le signal d'entrée.

- Retard à l'ouverture : (Clock-skew Δt_k)

Lors de la prise d'échantillon de l'échantillonneur, une incertitude persiste quand à l'instant auquel on a bloqué effectivement la valeur. Car bien que les temps de réaction soient courts, et notamment pour des signaux rapides, cette incertitude peut avoir des impacts non négligeables. Contrairement aux erreurs précédentes qui créent des erreurs aléatoires, le retard à l'ouverture est fixe. Il est le plus important mais le fait qu'il soit fixe le rend beaucoup moins problématique, car il peut être compensé.

III.1.3. Critère de performance

Du fait qu'il existe toute une gamme d'ADC pour différentes applications, un ensemble de critères est défini pour évaluer les performances des convertisseurs. Cependant, les critères de choix d'un ADC pour une application de mesure, ne sont pas forcément les mêmes pour une application de télécommunications. Ceci rend difficile le choix d'un convertisseur, d'autant plus que les fabricants de ADC utilisent des méthodes différentes de spécification des paramètres pour une meilleure distribution de leurs produits. Indépendamment de sa structure, les caractéristiques fonctionnelles du convertisseur sont spécifiées par un ensemble de paramètres. Ces paramètres changent en fonction des conditions de test, du bruit des générateurs, de variation de l'ensemble des paramètres technologiques du circuit lors de sa fabrication et de l'évolution de ces paramètres avec le vieillissement du circuit. La combinaison des paramètres d'erreur caractérisant l'ADC peut altérer le fonctionnement global du convertisseur [4] et [5].

- Rapport signal sur bruit :

La quantification d'un signal analogique sur un nombre limité de bits fait que le signal converti présente, par principe, une erreur par rapport au signal d'entrée. Le rapport signal sur bruit *SNR* (*Signal to Noise Ratio*) d'un ADC réel est exprimé par le rapport entre la puissance du signal fondamentale *S* et la puissance du bruit *B* excepté la composante continue et les raies harmoniques du signal.

$$SNR = 20 \log \left(\frac{S}{B} \right) \quad (3.7)$$

Pour un ADC idéal, c'est l'erreur de quantification qui a de l'importance et le SNR est entièrement déterminé par sa résolution et par la forme du signal d'entrée. La valeur efficace d'un signal sinusoïdal dont l'amplitude est égale à la pleine échelle de l'ADC est donnée par :

$$V_{eff} = \frac{2^N}{2\sqrt{2}} \quad (3.8)$$

Le rapport des équations (3.4) par (3.8) exprimé en décibel (dB) définit le SNR_{th} théorique de l'ADC idéal :

$$SNR_{th} = 20 \log \left(\frac{S}{B} \right) = 20 \log \frac{2^N \sqrt{12}}{2\sqrt{2}} = 6.02 \cdot N + 1.76 \text{ dB} \quad (3.9)$$

- Rapport signal sur bruit et distorsion SINAD :

Rapport entre la valeur RMS du signal S et la valeur RMS de la somme des amplitudes de toutes les autres fréquences présentes dans le signal sauf la tension continue.

- Nombre effectif de bits ENOB :

ENOB *Effective Number Of Bits* est une mise en évidence efficace des limitations d'un convertisseur A/D. Ce paramètre illustre la précision effective de votre convertisseur une fois les différentes erreurs additionnées. On s'aperçoit ainsi que le magnifique convertisseur 16 bits acheté très cher a en fait une largeur de code utile en sortie de, par exemple, seulement 8 bits.... C'est ce problème que l'on retrouve au niveau des TIADC pour lesquels on utilise des ADC de hautes résolutions mais où la résolution de sortie du système est fortement diminuée à cause des erreurs de disparité entre les convertisseurs (cf. partie III.3).

Le calcul de ENOB est basé sur la valeur du SINAD et est donné en retournant l'équation (3.9) :

$$N = \frac{(SINAD - 1,76dB)}{6,02} \equiv ENOB \quad (3.10)$$

- Plage dynamique sans parasite SFDR :

SFDR *Spurious Free Dynamic Range* est l'amplitude qui sépare le signal utile de l'harmonique d'amplitude la plus élevée (cf. Fig. 17). C'est un paramètre qui détermine la dynamique de l'ADC et qui est important pour les ADC utilisés en télécommunication. C'est le paramètre principal que nous utiliserons pour évaluer les performances de notre travail.

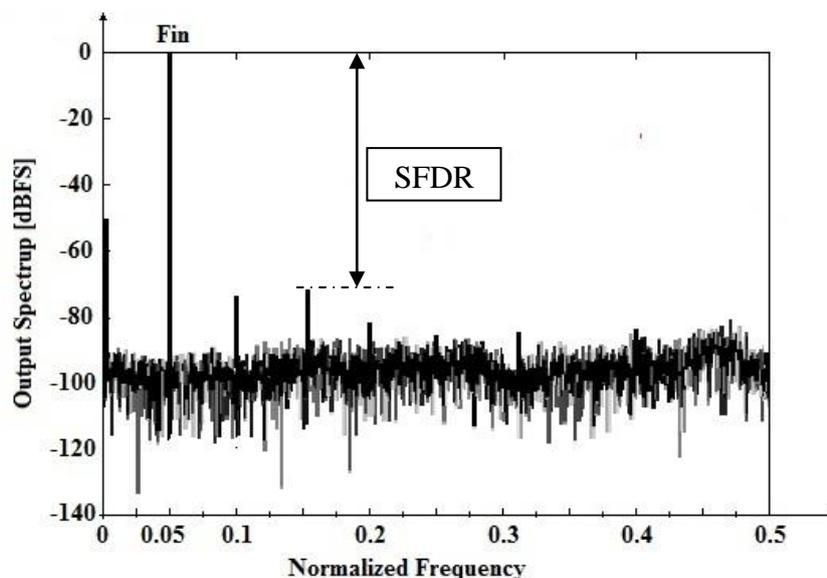


Figure 17 - SFDR Spurious Free Dynamic Range

III.1.4. Architectures des ADC

Dans cette partie nous allons présenter brièvement les différentes architectures classiques de convertisseur analogique-numérique [4]. Chaque architecture a ses avantages, ses inconvénients et ses applications. Nous allons voir pourquoi un ADC seul ne peut répondre aux exigences de la SDR en termes de rapidité, consommation et résolution.

- Convertisseurs flash (parallèle)

Les convertisseurs flash sont certainement les convertisseurs les plus rapides. Ils peuvent actuellement atteindre une vitesse de quelques GHz. Cependant, ils ont une faible résolution qui ne dépasse pas les 8 bits car si l'on souhaite réaliser un ADC parallèle sur 12 bits, il ne faudra 4095 comparateurs, et on imagine sans peine la taille de la puce et le coût induit.

- Convertisseurs pipeline et SAR

Les convertisseurs pipeline et SAR permettent d'avoir un compromis entre vitesse et résolution avec une meilleure résolution pour les SAR.

- Convertisseurs Sigma Delta

Les convertisseurs Sigma Delta sont clairement les plus précis. Ils peuvent avoir une résolution de 24 bits. Ils ont aussi d'autres avantages : une haute dynamique, une grande linéarité et une faible sensibilité aux problèmes d'imperfections des circuits. L'inconvénient majeur de ce convertisseur est sa faible vitesse qui ne dépasse pas 5 MHz.

Pour récapituler les caractéristiques des différentes architectures de conversion, le tableau 2 présente pour chacune de ces architectures la vitesse maximale, la résolution et la consommation.

Architecture	Vitesse	Résolution	Consommation
Parallèle	Des centaines de MHz	8 à 10	--
Pipeline	Des dizaines de MHz	12 à 14	-
SAR	Une dizaine de MHz	14 à 16	+
Sigma Delta	5 MHz	16 à 20	++

Tableau 2 - Performance des architectures ADC

On remarque que la vitesse maximale d'échantillonnage est liée à la résolution du convertisseur. En effet, la vitesse d'échantillonnage augmente considérablement pour les convertisseurs de faible résolution. Inversement une bonne résolution du convertisseur est pénalisée en vitesse. D'où la réunion des ces deux caractéristiques reste difficile surtout lorsqu'on cherche à augmenter la vitesse maximale. La consommation du convertisseur est un critère crucial lui aussi et liée essentiellement à la nature de l'application envisagée.

Or pour des récepteurs radio fréquence fonctionnant en mode large bande ou multistandard, les ADC sont amenés à numériser un signal présentant une large plage dynamique et par conséquent un nombre de bits dépassant 16 bits est exigé pour la conversion analogique numérique. La numérisation nécessite aussi une grande fréquence d'échantillonnage. La puissance consommée et la miniaturisation du convertisseur sont des points critiques pour les récepteurs multistandard intégrés sur un terminal mobile.

Les architectures de conversion présentées jusqu'à présent sont donc incapables de satisfaire l'émergence de ces nouveaux récepteurs RF. On peut essayer de se rapprocher de ces nouvelles exigences par l'utilisation de technologies plus avancées, mais cela se fait toujours au dépend du coût.

Ainsi dans la partie III.2 nous allons présenter la technique d'entrelacement d'ADC dans le temps qui permet d'augmenter la fréquence d'échantillonnage tout en aillant une bonne résolution, sans augmentation conséquente de la consommation ni du coût.

III.2. Principe du TIADC

Le principe du TIADC à entrelacer dans le temps M ADC "identiques". Tous les ADC reçoivent le même signal analogique à convertir, cependant le moment de conversion est différent entre chacun. En effet l'horloge qui définit le moment d'échantillonnage est déphasée de $2\pi/M$ entre ADC1 et ADC2, entre ADC2 et ADC3, Après la phase de conversion de chaque ADC à la fréquence f_s/M , le multiplexeur, contrôlé par un compteur, regroupe les échantillons des différentes voies d'une manière cyclique. La fréquence d'échantillonnage du système résultant est multipliée par M le nombre d'ADC utilisés. (cf. Fig. 18)

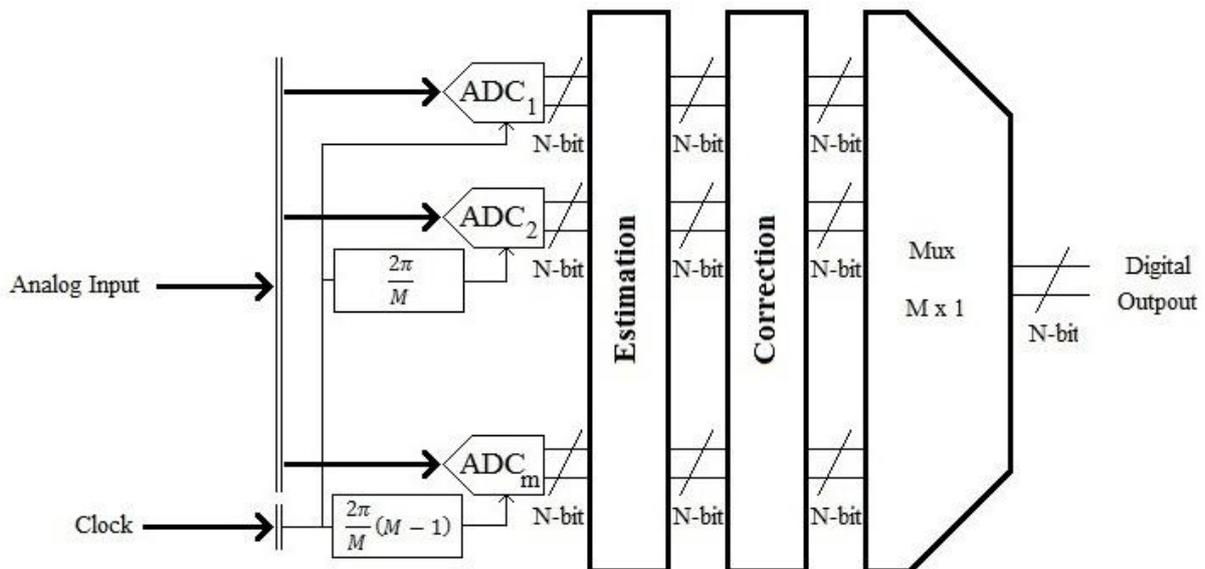


Figure 18 - Principe de la conversion par entrelacement temporel avec correction d'erreur

Aujourd'hui, les TIADC atteignent une fréquence d'échantillonnage de 20 GS/s. Cette rapidité est obtenue, certes, grâce à l'évolution technologique mais aussi grâce à la maîtrise de la technique d'entrelacement : le nombre de convertisseurs à entrelacer ne cesse d'augmenter et par conséquent, le taux d'échantillonnage se multiplie pour atteindre des valeurs importantes. Paradoxalement, la majorité des TIADC disponibles sur le marché ont une résolution n'excédant pas 10 bits. L'augmentation du taux d'échantillonnage du système se fait donc au dépend de la précision.

En effet, les erreurs statiques et dynamiques déjà présentées sont des erreurs fixes pour un convertisseur donné mais aléatoires d'un ADC à un autre. Elles sont liées à l'imperfection du processus de fabrication, au vieillissement des circuits ou aux conditions d'utilisations antérieures. Pour deux ADC dis "identiques", ces erreurs ont de faibles chances d'être identiques et lors de leur utilisation dans une structure entrelacée, l'amplitude de ces erreurs et leurs positions fréquentielles changent. Les disparités entre les convertisseurs analogiques numériques entraînent l'apparition d'erreurs dans le signal numérique à la sortie de la structure entrelacée. Ainsi, la précision du TIADC est dégradée par rapport à celle de chaque ADC de la structure.

III.3. Etudes des erreurs de disparité

On définit comme signal d'étude, un signal sinusoïdal donné par l'équation suivante :

$$x(t) = A \times \sin(2\pi f_0 t) \quad (3.11)$$

Où A et f_0 sont respectivement l'amplitude maximale et la fréquence du signal d'entrée.

Soit k avec $1 \leq k \leq M$ l'ordre de l'ADC utilisé dans le TIADC. Les instants d'échantillonnage de chaque ADC sont exprimés par la relation suivante :

$$T_k(n) = nMT_s + kT_s \quad (3.12)$$

n indique l'ordre des échantillons et T_s est la période d'échantillonnage de chaque voie. Ainsi, en négligeant le bruit de quantification, la sortie d'une voie idéale de la structure entrelacée s'écrit sous la forme suivante :

$$y_k(t) = A \sin(2\pi f_0 t) \times \overbrace{\sum_{n=-\infty}^{+\infty} \delta(t - nMT_s - kT_s)}^{p(t)} \quad (3.13)$$

▪ Influence de l'erreur d'offset

L'erreur d'offset, décrite dans le paragraphe III.1.2, est une constante additive notée O_k pour le $k^{\text{ième}}$ ADC. En tenant compte de l'erreur d'offset, la sortie idéale devient :

$$y_k(t) = (A \sin(2\pi f_0 t) + O_k) \times \sum_{n=-\infty}^{+\infty} \delta(t - nMT_s - kT_s) \quad (3.14)$$

L'erreur sur le signal d'entrée due à la présence de l'erreur d'offset sur la voie correspond à la différence entre les équations (3.14) et (3.13) et exprimée par :

$$e_{\text{offset}_k}(t) = O_k \times \sum_{n=-\infty}^{+\infty} \delta(t - nMT_s - kT_s) \quad (3.15)$$

L'analyse temporelle de l'erreur d'offset pour M voies est compliquée. Nous passons ainsi au domaine fréquentiel pour dégager l'effet de disparité des erreurs d'offset entre les M voies. Nous utiliserons alors la Transformée de Fourier (TF).

$$TF(O_k) = O_k(f) = \int_{-\infty}^{+\infty} O_k e^{-2j\pi f t} dt = O_k \delta(f) \quad (3.16)$$

$$\begin{aligned} TF(p(t)) = P(f) &= \int_{-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} \delta(t - nMT_s - kT_s) e^{-2j\pi f t} dt \\ &= \frac{1}{MT_s} \sum_{n=-\infty}^{+\infty} \delta\left(f - \frac{nf_s}{M}\right) e^{-jnk\frac{2\pi}{M}} \end{aligned} \quad (3.17)$$

La Transformée de Fourier de $e_{off_k}(t)$, produit de convolution des fonctions $P(f)$ et $O_k(f)$ s'écrit sous la forme :

$$TF(e_{off_k}(t)) = TF(O_k \times p(t)) = \frac{1}{2\pi} (O_k(f) * P(f)) \quad (3.18)$$

$$TF(e_{off_k}(t)) = E_{OFF_k}(f) = \sum_{n=-\infty}^{+\infty} \frac{O_k}{2\pi MT_s} \delta\left(f - \frac{nf_s}{M}\right) e^{-jnk\frac{2\pi}{M}}$$

Finalement la Transformée de Fourier de l'erreur engendrée par la disparité d'offset dans un TIADC composé de M ADC est exprimée par :

$$E_{OFF_k}(f) = \sum_{k=1}^M \sum_{n=-\infty}^{+\infty} \frac{O_k}{2\pi MT_s} \delta\left(f - \frac{nf_s}{M}\right) e^{-jnk\frac{2\pi}{M}} \quad (3.19)$$

L'équation (3.19) montre l'apparition de plusieurs raies non-harmoniques : la composante DC et des composantes localisées à $n*fs/M$.

- Influence de l'erreur de gain (Annexe xxx)

$$E_{G_k}(f) = \sum_{k=1}^M \sum_{n=-\infty}^{+\infty} \frac{jAG_k}{4\pi MT_s} \left[\delta\left(f - \frac{nf_s}{M} + f_0\right) - \delta\left(f - \frac{nf_s}{M} - f_0\right) \right] e^{-jnk\frac{2\pi}{M}} \quad (3.20)$$

- Influence de l'erreur de clock skew (Annexe xxx)

$$E_{CK_k}(f) = \sum_{k=1}^M \sum_{n=-\infty}^{+\infty} \frac{Af_0\Delta t_k}{2MT_s} \left[\delta\left(f - \frac{nf_s}{M} - f_0\right) + \delta\left(f - \frac{nf_s}{M} + f_0\right) \right] e^{-jnk\frac{2\pi}{M}} \quad (3.21)$$

Les équations (3.20) et (3.21) mettent en évidence l'apparition de plusieurs raies non-harmoniques localisées à des fréquences égales à $n*fs/M \pm f_0$.

Dans la suite, nous allons voir comment le modèle du TIADC comprenant toutes ces erreurs de disparité entre les ADC a été créé sous MATLAB-SIMULINK.

III.4. Modélisation du TIADC sous MATLAB-SIMULINK

III.4.1. Choix des paramètres des ADC et de simulation

- Paramètre des ADC utilisés :

Afin de rapprocher notre travail du monde réel, nous avons décidé d'utiliser les caractéristiques d'un ADC existant. Le choix s'est porté sur un ADC de 14 bits [6] de la société NXP Semiconductors. La limitation de la résolution est due aux tests que l'on veut réaliser sur la carte FPGA par la suite. (cf. partie V pour explication)

ADC1412D

Résolution : 14 bits

Gain : $\pm 0.5\%$ Pleine échelle

Offset : $\pm 2\text{mV}$

Distorsions : $\alpha 2$ second harmonique 82 dB ; $\alpha 3$ troisième harmonique 81 dB

- Paramètres de simulation :

En ce qui concerne la simulation il y a quelques paramètres très importants à prendre en compte, notamment le nombre d'échantillons utilisés pour effectuer l'estimation des erreurs de chaque ADC. Comme on veut tracer une FFT du signal échantillonné afin de pouvoir voir le spectre de celui-ci, nous devons utiliser un nombre d'échantillon *Nech* pour chaque ADC correspondant à une puissance de 2 notée *L*. La taille de la FFT *Nfft* est alors égale à *M* fois *Nech*.

$$Nech = 2^L \quad (3.22)$$

$$Nfft = M \times Nech \quad (3.23)$$

Ensuite, il y a la fréquence du signal utilisé pour l'estimation qui doit être normalisée afin de travailler sur un nombre entier de période et donc d'avoir une estimation des erreurs qui fonctionne mieux. Les fréquences d'échantillonnage f_s et de signal d'entrée f_0 doivent être cohérentes. L'amplitude du signal d'entrée est notée *A*.

$$f_s = 1 \quad (3.24)$$

$$f_0 \text{ "normalised" } = \frac{\text{round}\left(\frac{f_0 \text{ "wanted" }}{f_s \times Nech}\right)}{f_s \times Nech} \quad (3.25)$$

III.4.2. Modélisation d'un ADC

- Design : Le modèle de l'ADC 'ADC1' est détaillé dans la figure 29.

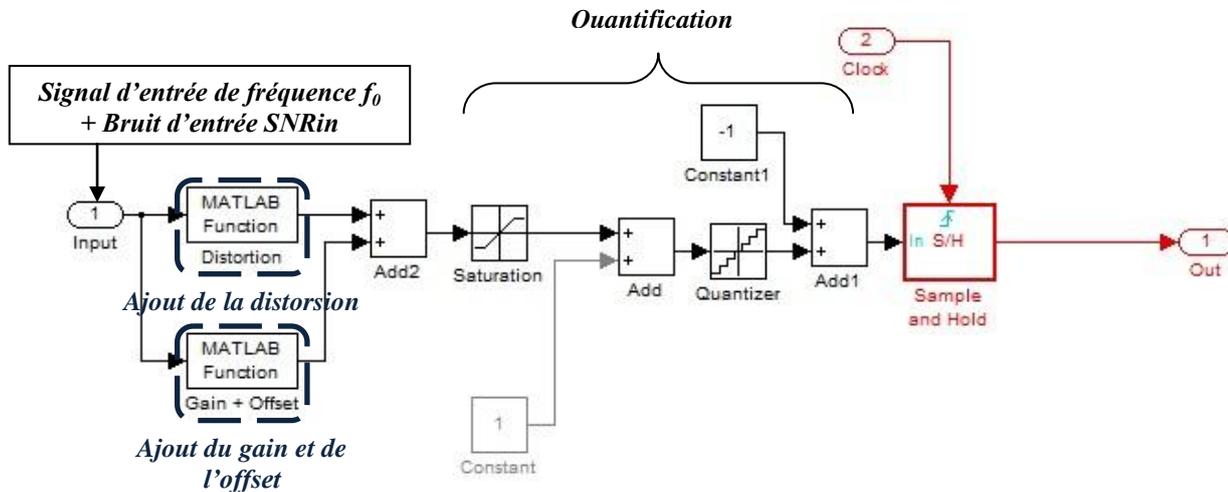


Figure 19 - Modèle SIMULINK de l'ADC1

Avant de faire la conversion nous avons mis un étage de saturation égal à la tension de pleine échelle que nous allons utiliser dans toutes nos simulations et qui est de 2Volts crête à crête.

Pour l'étape de quantification on rajoute +1 avant pour réaliser une quantification entre 0 et 2, car sinon SIMULINK peut engendrer des erreurs de quantification autour de 0. Après la quantification on enlève 1.

L'échantillonnage est réalisé par le block "Sample and Hold", lequel est commandé par une horloge qui par la suite servira aussi à intégrer l'erreur de Clock Skew.

Les différents paramètres d'erreurs d'offset, de gain et de distorsion de l'ADC (cf. partie III.4.1) sont donnés par :

Erreurs d'offset : 0.001 Volts
 Erreurs de gain : 1.008
 Distorsion : Dist2 = 0.00022
 Dist3 = 0.00044

Pour rajouter ces erreurs on utilise la fonction polyval() (cf. Fig. 19) :

→ $\text{polyval}([G(1) \text{ Off}(1)], u)$ et $\text{polyval}([Dist3(1) Dist2(1) 0 \ 0], u)$.

- Simulation : Les paramètres de simulation sont représentés dans le tableau 3.

f_s	f_0 "wanted"	L	N_{ech}	N_{fft}	SNR_{in}	A
1	1/10	12	4096	4096	100dB	0,9

Tableau 3 - Paramètres de simulation de l'ADC

Comme on peut le voir sur la figure 20 la modélisation de l'ADC est correctement réalisée. On retrouve bien un signal numérisé en sortie avec les harmoniques α_2 et α_3 environ égaux à -82dB et -81dB. Notre ADC de 14 bits a en faite une résolution de 13,2 bits. Il faut noter que nous prenons le pire des cas pour notre calcul car nous utilisons le SFDR et non le SINAD.

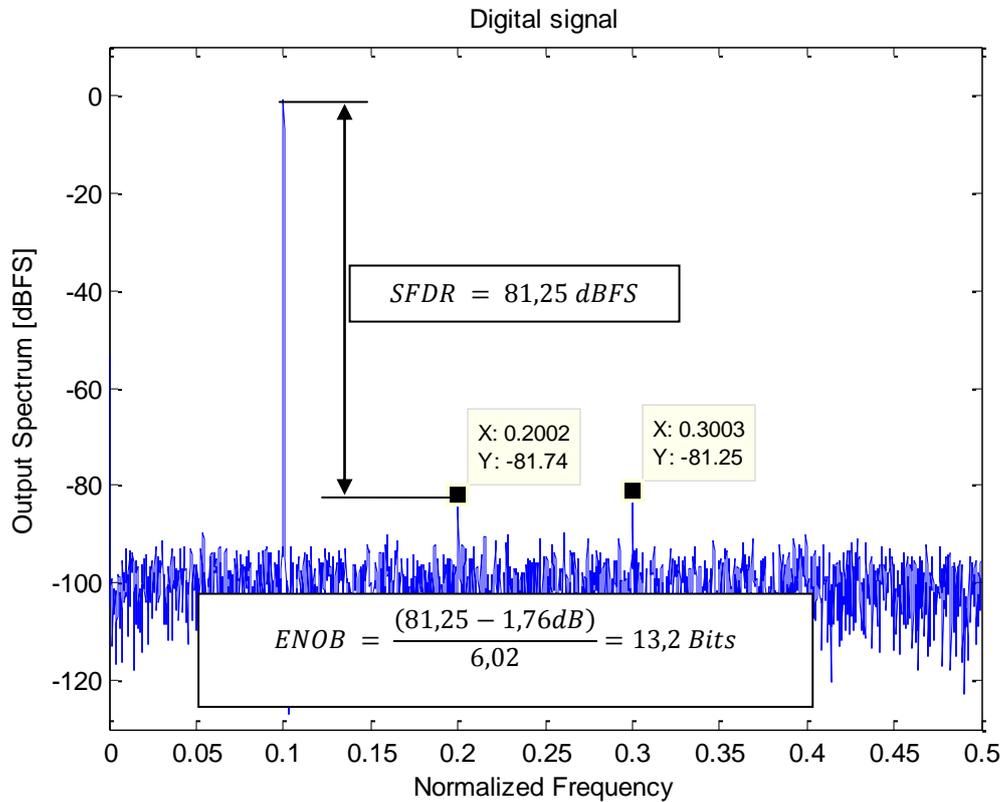


Figure 20 - Spectre du signal numérisé par l'ADC1

Maintenant nous pouvons donc passer à la modélisation du TIADC qui va réutiliser 'ADC1'.

III.4.3. Modélisation du système global TIADC

- Design : Le modèle du TIADC 'TIADC_sim' détaillé dans la figure 21

Signal d'entrée + bruit

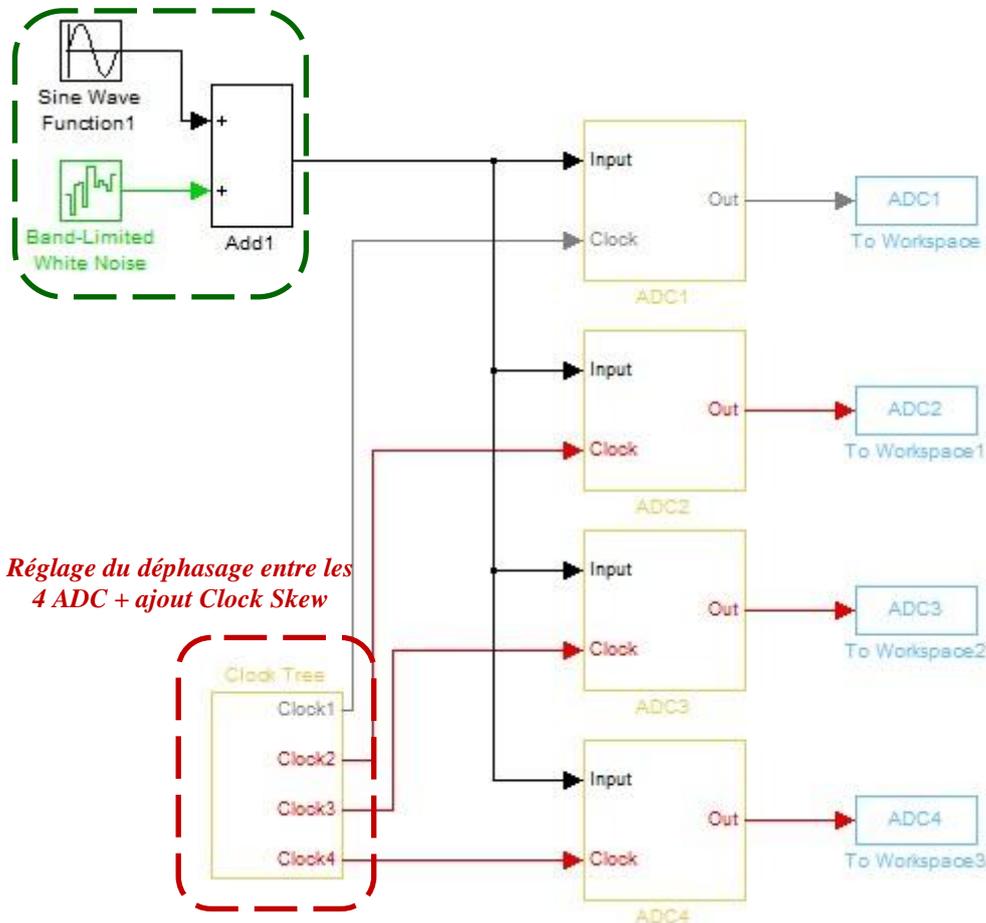


Figure 21 - Modèle SIMULINK du TIADC 'TIADC_sim'

Le TIADC est composé de 4 ADC chacun réalisé de la même manière que 'ADC1'. Ils reçoivent tous le même signal analogique à convertir. Cependant ils ont chacun une horloge différente pour fixer le moment d'échantillonnage. Pour cela on a utilisé la fonction retard sur chaque horloge avec laquelle on peut aussi réaliser l'erreur de Clock-Skew Δt_k .

Les différents paramètres d'erreurs d'offset, de gain et de retard d'ouverture des quatre ADC "identiques" (cf. partie III.4.1) sont donnés par :

Erreurs d'offset : {0.001 0.002 0.0015 -0.001} Volts

Erreurs de gain : {1.008 0.992 0.996 1.004}

Erreurs de retard d'ouverture : {0, 1, 3, 3} $T_s/10$

Distorsion : Dist2 = {0.00022 0.0002 0.00024 0.00022}

Dist3 = {0.00044 0.0004 0.00042 0.00044}

- Simulation : Les paramètres de simulation sont représentés dans le tableau 4.

f_s	f_o "wanted"	L	N_{ech}	N_{fft}	SNR_{in}	A
1	1/10	12	4096	16384	100dB	0,9

Tableau 4 - Paramètres de simulation du TIADC

Comme on peut le voir sur la figure 22 la modélisation du TIADC est correctement réalisée car on voit bien apparaître les composantes parasites dues aux erreurs de disparité aux fréquences $n*f_s/M$ et $n*f_s/M \pm f_o$. On retrouve aussi les harmoniques α_2 et α_3 .

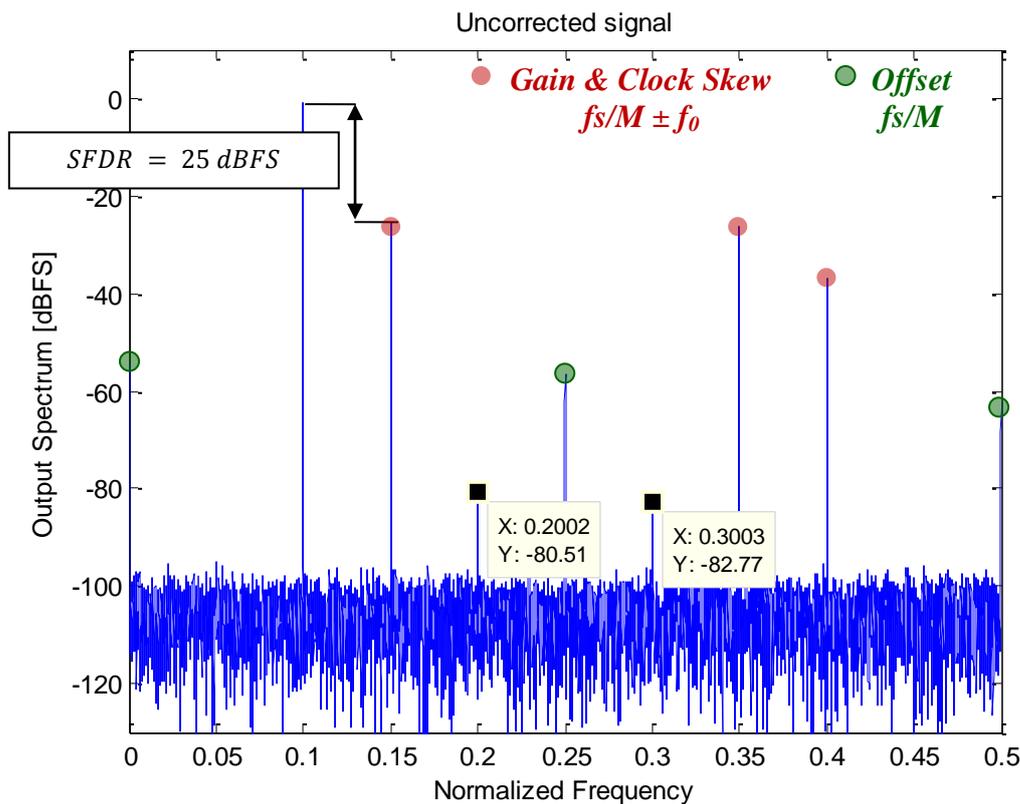


Figure 22 - Spectre du signal numérisé par le TIADC

$$ENOB = \frac{(25 - 1,76dB)}{6,02} = 3,86 \text{ Bits}$$

A cause de ces erreurs de disparité entre les ADC on voit que le gain en fréquence se fait au détriment de la résolution du système. Il faut donc trouver une solution pour pouvoir corriger ces erreurs.

IV. Méthode de correction des erreurs

IV.1. Présentation

Après avoir étudié les différentes publications sur le thème des corrections d'erreurs de disparité dans le TIADC, nous avons développé notre propre algorithme d'estimation du gain, de l'offset et du retard à l'ouverture. Cette méthode de calibration est en partie similaire à celle utilisée dans les articles [1] et [2] mais présente l'avantage de ne pas utiliser de séquence et donc de ne pas introduire d'erreur supplémentaire au niveau analogique.

Rq : Pour que la méthode fonctionne correctement il faut que la fréquence utilisée en signal d'entrée et la fréquence d'échantillonnage soient cohérentes.

IV.1.1. Notation

Un signal sinusoïdal est appliqué à l'entrée du TIADC. Ce signal sera noté par :

$$x(t) = A \times \sin(2\pi f_0 t) \quad (4.1)$$

Où A et f_0 sont respectivement l'amplitude maximale et la fréquence du signal d'entrée.

M désigne le nombre d'ADC dans le TIADC et T_s la période d'échantillonnage du système complet. La longueur du vecteur d'observation (taille de FFT) est notée N_{fft} .

G_k , O_k et Δt_k désignent respectivement le gain, l'offset et le Clock Skew associés au $k^{\text{ième}}$ ADC, avec $1 \leq k \leq M$. Le jitter δt_k est modélisé par un bruit blanc avec une distribution gaussienne dont l'écart type est égal à 0.5% de la période d'échantillonnage.

Nous supposons que le signal respecte le critère de Nyquist pour la reconstitution du signal et que les erreurs de gain, d'offset et de Clock Skew sont statiques ou varient lentement dans le temps. Cela signifie que ces erreurs peuvent être considérées comme des constantes pour chaque ADC.

IV.1.2. Algorithme de calibration

Rq : $N_{ech} = N_{fft}/M$ est le nombre d'échantillons utilisés pour l'estimation.

Le $n^{\text{ième}}$ échantillon du signal d'entrée du $k^{\text{ième}}$ ADC est exprimée par :

$$x_{k,\tilde{n}}(n) = A \times \sin(2\pi f_0 \tilde{n} T_s) \quad (4.2)$$

Avec $\tilde{n} = (k + Mn)$, et $0 \leq n \leq N_{ech} - 1$

Soit $y_k(n)$ le $n^{\text{ième}}$ échantillon du signal à la sortie du $k^{\text{ième}}$ ADC. Nous pouvons écrire :

$$y_k(n) = \tilde{x}_{k,\tilde{n}}(n) + O_k \quad (4.3)$$

Où

$$\tilde{x}_{k,\tilde{n}}(n) = A G_k \times \sin(2\pi f_0 (\tilde{n} T_s + \Delta t_k + \delta t_k)) \quad (4.4)$$

En utilisant la propriété de trigonométrie de $\sin(a+b)$, nous pouvons écrire :

$$\tilde{x}_{k,\tilde{n}}(n) = AG_k \begin{pmatrix} \sin(2\pi f_0 \tilde{n} T_S) \cos(2\pi f_0 (\Delta t_k + \delta t_k)) \\ + \cos(2\pi f_0 \tilde{n} T_S) \sin(2\pi f_0 (\Delta t_k + \delta t_k)) \end{pmatrix} \quad (4.5)$$

Puisque $2\pi f_0 (\Delta t_k + \delta t_k) \ll 1$ l'équation (4.3) devient :

$$y_k(n) = AG_k \sin(2\pi f_0 \tilde{n} T_S) + 2\pi f_0 AG_k (\Delta t_k + \delta t_k) \cos(2\pi f_0 \tilde{n} T_S) + O_k \quad (4.6)$$

IV.1.2.1. Estimation de l'offset

La première étape de cet algorithme est l'estimation d'offset. Cette estimation est facilement réalisée à l'aide de l'équation suivante :

$$\overline{O_k} = moy(y_k) \quad (4.7)$$

IV.1.2.2. Estimation du gain et du retard d'ouverture

La deuxième étape est consacrée à l'estimation des erreurs de gain et de Clock Skew. Tout d'abord nous enlevons l'Offset estimé auparavant pour retrouver le signal d'entrée avec la présence des erreurs de gain, de Clock Skew et de jitter.

En utilisant l'équation (4.6), on obtient :

$$\tilde{y}_k(n) = y_k(n) - O_k \quad (4.8)$$

$$\tilde{y}_k(n) = AG_k \sin(2\pi f_0 \tilde{n} T_S) + 2\pi f_0 AG_k (\Delta t_k + \delta t_k) \cos(2\pi f_0 \tilde{n} T_S)$$

Sachant que l'erreur du jitter δt_k est faible par rapport à celle du Clock Skew Δt_k , on peut simplifier le système précédent afin d'estimer seulement les erreurs de gain et de Clock Skew. Le système contiendra deux inconnus G_k et Δt_k et sera résolu en utilisant la propriété de moyenne nulle des fonctions cosinus et sinus. On notera $T_k = 2\pi f_0 \Delta t_k$.

Soient les deux signaux suivants :

$$V_{k,\tilde{n}}(n) = \sin(2\pi f_0 \tilde{n} T_S) \quad (4.9)$$

$$W_{k,\tilde{n}}(n) = \cos(2\pi f_0 \tilde{n} T_S) \quad (4.10)$$

On obtient le gain grâce à l'équation suivante :

$$AG_k = \frac{2}{Nech} \times \tilde{y}_k(n) \times V_{k,\tilde{n}}^T(n) \quad (4.11)$$

Car :

$$\sum_{n=0}^{Nech-1} AG_k \sin(2\pi f_0 \tilde{n} T_S)^2 = \sum_{n=0}^{Nech-1} \left(\frac{AG_k - \cos(4\pi f_0 \tilde{n} T_S)}{2} \right) = \frac{Nech AG_k}{2} + 0$$

$$\sum_{n=0}^{Nech-1} 2\pi f_0 AG_k (\Delta t_k + \delta t_k) \cos(2\pi f_0 \tilde{n} T_S) \sin(2\pi f_0 \tilde{n} T_S) = 0$$

Et on obtient l'erreur de phase par l'équation :

$$T_k = \frac{\tilde{y}_k(n) \times W_{k,\tilde{n}}^T(n)}{\tilde{y}_k(n) \times V_{k,\tilde{n}}^T(n)} \quad (4.12)$$

Car :

$$\sum_{n=0}^{Nech-1} 2\pi f_0 AG_k (\Delta t_k + \delta t_k) \cos(2\pi f_0 \tilde{n} T_S) \cos(2\pi f_0 \tilde{n} T_S) = \frac{AG_k T_k Nech}{2} + 0$$

$$\sum_{n=0}^{Nech-1} AG_k \sin(2\pi f_0 \tilde{n} T_S) \cos(2\pi f_0 \tilde{n} T_S) = 0$$

IV.1.3. Correction du signal

IV.1.3.1. Correction en connaissant la fréquence d'entrée (C_{y_k})

La correction d'un signal d'entrée sinusoïdal connu peut être réalisée à l'aide de la formule suivante :

$$C_{y_k}(n) = \frac{y_k(n) - \overline{O_k} - T_k W_{k,\tilde{n}}(n) AG_k}{AG_k} \quad (4.13)$$

Il faut connaître la fréquence du signal à corriger car pour supprimer l'erreur de Δt_k on a besoin d'utiliser le signal W_k (cf. équation (4.10)). Le signal d'entrée et W_k devront aussi être bien synchronisés de manière à avoir un sinus et un cosinus.

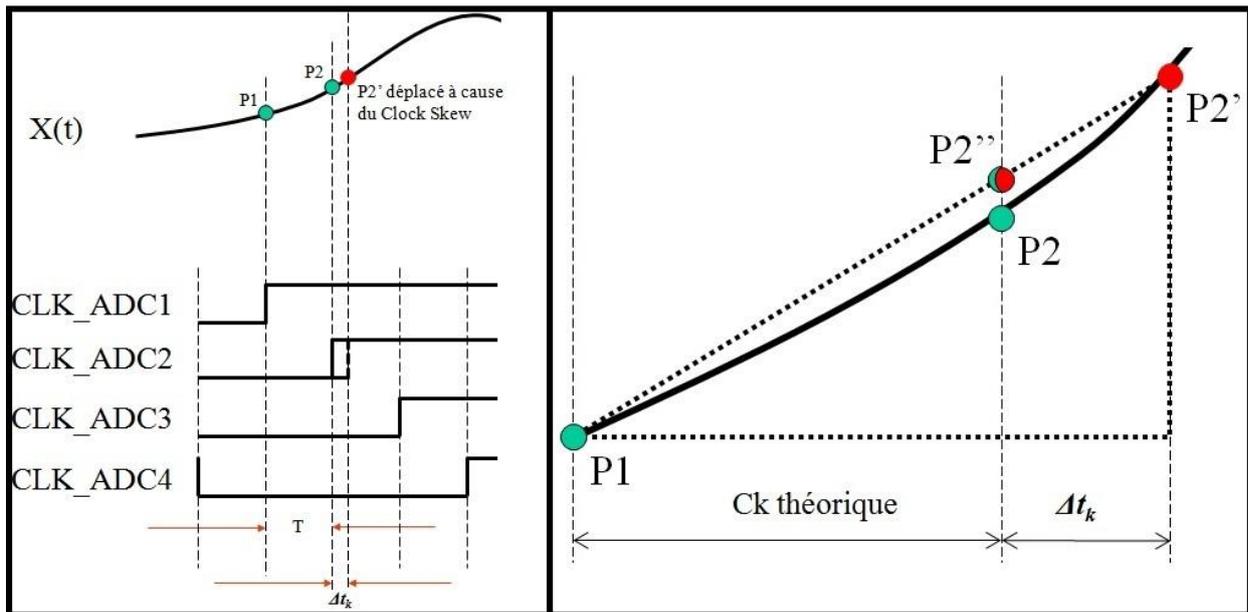
On notera que le signal corrigé aura toujours une amplitude de '1', et que pour corriger les erreurs on n'a jamais besoin de connaître la valeur de l'amplitude du signal d'entrée.

Cependant il est difficile de connaître avec précision la fréquence du signal que l'on va convertir. On peut imaginer placer un détecteur de fréquence à l'entrée du TIADC mais les différents temps de propagation, et la précision du détecteur font que cette méthode est difficile à mettre en œuvre.

IV.1.3.2. Correction utilisant l'interpolation pour le Clock Skew (C''_{y_k})

Il faut noter que cette interpolation sera effectuée après avoir corrigé l'offset et le gain à l'aide de l'équation (4.14).

$$C''_{y_k}(n) = \frac{y_k(n) - \overline{O_k}}{AG_k} \quad (4.14)$$


 Figure 23 - Méthode d'interpolation utilisée pour corriger l'erreur de Δt_k

En utilisant la propriété de Thales sur la figure 23 on obtient les équations suivantes :

$$\frac{P2'' - P1}{P2' - P1} = \frac{Ck_{th}}{Ck_{th} + \Delta t_k} \quad (4.15)$$

$$P2'' = P1 + (P2' - P1) \times \frac{Ck_{th}}{Ck_{th} + \Delta t_k}$$

On peut voir que le point P2 est l'échantillon parfait pour l'ADC2, mais à cause de l'erreur de Clock Skew nous obtenons le point P2'. Pour améliorer le signal digital nous devons approximer le point P2' au plus proche possible du point P2. Pour cela nous utilisons la relation de Thales (cf. équation (4.15)) et nous obtenons le point P2''. Comme P2'' est plus proche du point P2, le SFDR est amélioré.

Il faut noter que l'on prend pour base de notre interpolation l'ADC1. On définit l'ADC1 comme ayant une erreur de Clock Skew égale à 0.

La Correction des erreurs se fait donc en deux étapes suivant l'équation (4.14) puis l'équation (4.16) :

$$C''_{y_k}(n) = C'_{y_{k-1}}(n) + (C'_{y_k}(n) - C'_{y_{k-1}}(n)) \times \frac{Ck_{th}}{Ck_{th} + \Delta t_k} \quad (4.16)$$

IV.2. Simulation et validation sous MATLAB-Simulink

Pour valider notre algorithme, nous l'avons codé sous Matlab puis nous avons effectué différentes simulations à l'aide du modèle 'TIADC_sim'.

IV.2.1. Impact du nombre d'échantillons sur la précision de l'estimation des erreurs

Dans un premier temps nous étudierons l'impact du nombre d'échantillons N_{ech} utilisé pour réaliser l'estimation des erreurs sur la précision de cette estimation. (cf. Fig. 24)

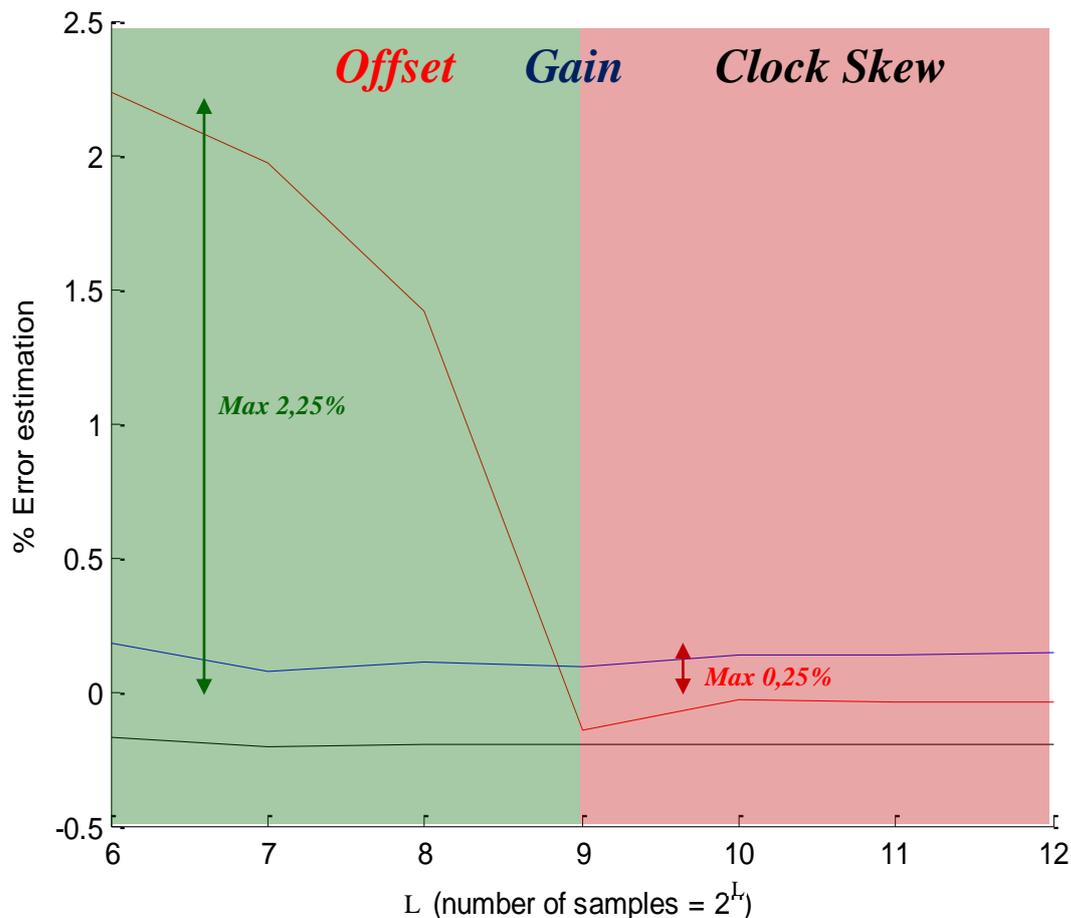


Figure 24 - % Erreur d'estimation de l'offset, du gain et du Clock Skew en fonction du nombre d'échantillons

La figure ci-dessus nous montre que pour $L = 6$ soit $N_{ech} = 64$ nous avons une estimation des erreurs qui est tout à fait acceptable et qu'au dessus de $L = 9$ soit $N_{ech} = 512$ la précision de l'estimation est très bonne et n'évolue plus.

Cette méthode d'estimation des erreurs d'offset, de gain et de retard d'ouverture peut donc se faire sans un trop grand nombre d'échantillons, d'où un gain en temps, mais également en ressource mémoire. Ceci est très important pour le coût et la taille du système. Lors des tests nous utiliserons d'ailleurs un nombre d'échantillons égale à 64 (cf. partie V).

IV.2.2. Correction des erreurs

- Méthode idéale avec fréquence du signal à numériser connue :

La figure 25 montre les spectres de sortie du TIADC à 4 ADC avant et après la calibration.

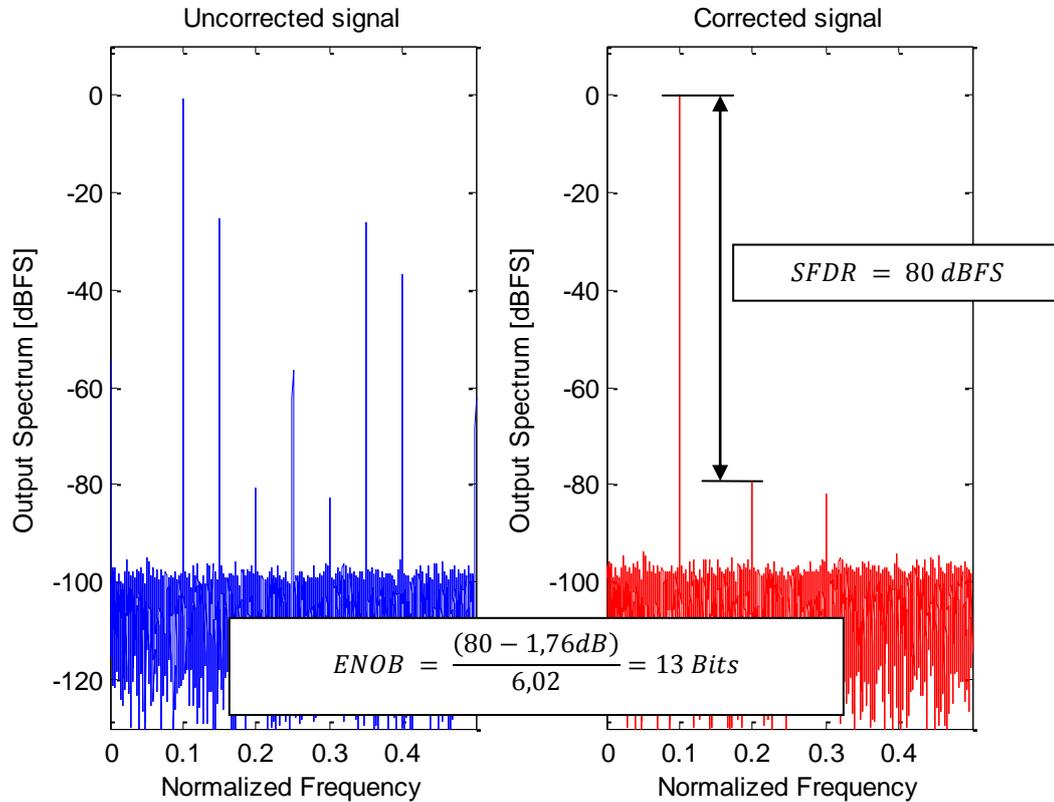


Figure 25 - Spectre du signal TIADC non corrigé puis corrigé en connaissant la fréquence d'entrée

Après la correction, on constate une élimination totale des composantes spectrales liées aux erreurs. En effet, l'estimation précise de ces erreurs permet d'éliminer les composantes parasites situées aux fréquences $n*fs/M$ et $n*fs/M \pm fo$.

Cette méthode de correction permet l'estimation et la correction des erreurs d'offset, de gain et de retard d'ouverture. Elle permet d'avoir une structure entrelacée qui assure une conversion très proche du nombre total de bits de chaque ADC avec une augmentation considérable de la vitesse. En effet, le nombre de convertisseurs de la structure entrelacée TIADC à calibrer est théoriquement sans limite puisque cette méthode permet la correction de plusieurs voies simultanément. D'où, cette méthode est intéressante d'un point de vue résolution et vitesse d'échantillonnage. Un autre critère important pour cette méthode est sans doute le coût réduit d'implémentation qui ne dépend pas du nombre de convertisseurs à entrelacer.

Cependant cette méthode très efficace à un inconvénient majeur qui est le fait d'avoir besoin de connaître la fréquence du signal d'entrée. C'est pour cela que nous avons mis en place la seconde méthode de correction qui utilise l'interpolation pour l'erreur de Clock-Skew.

- Méthode plus réaliste en utilisant l'interpolation la correction de Δt_k :

Comme nous avons pu le voir dans la simulation de la figure 25, la correction de l'offset et du gain se fait très bien. Ainsi dans un premier temps nous allons nous concentrer exclusivement sur la correction de Δt_k .

Pour la première simulation (cf. Fig. 26), seule l'erreur Δt_k et la distorsion est introduite dans le TIADC. Pour la seconde (cf. Fig. 27), toutes les erreurs sont introduites dans le TIADC. Cependant on peut remarquer que le signal corrigé est le même dans les deux cas, on obtient un SFDR de 39dB.

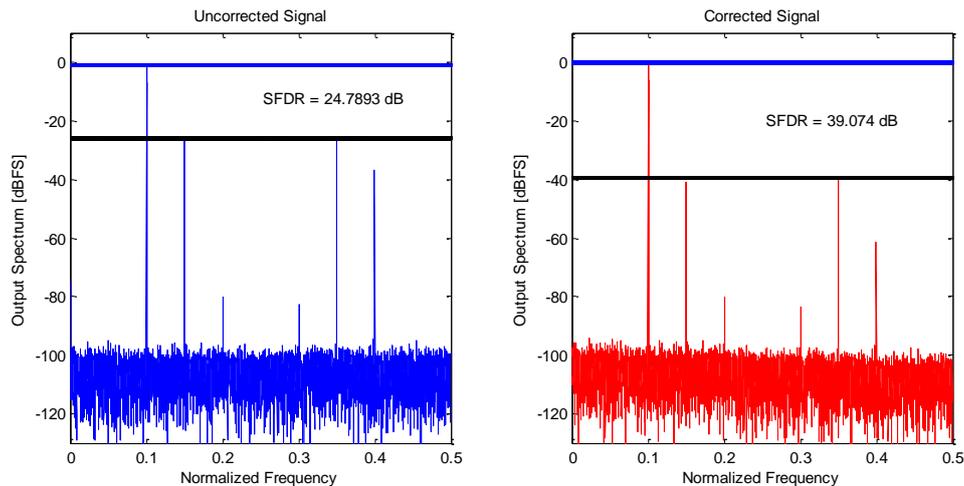


Figure 26 - Spectre du signal TIADC (avec Δt_k) non corrigé puis corrigé par interpolation

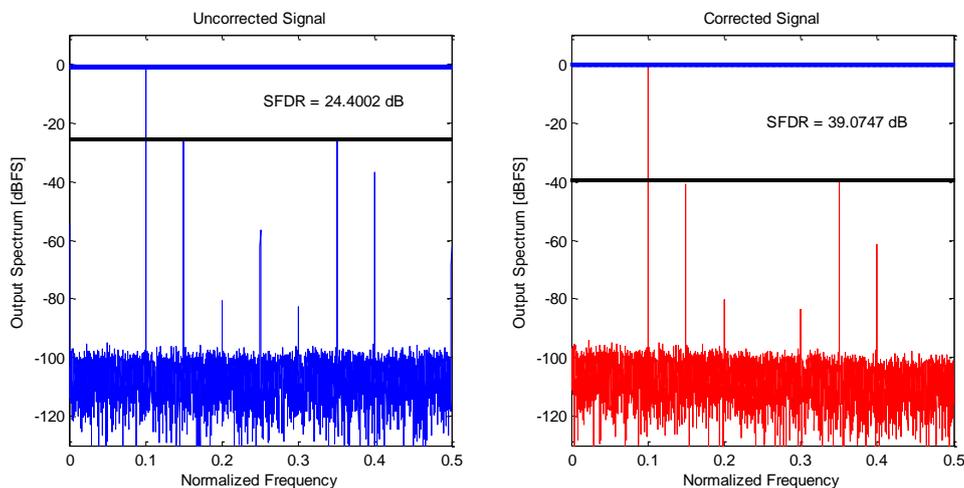


Figure 27 - Spectre du signal TIADC (avec G_k , O_k et Δt_k) non corrigé puis corrigé par interpolation

$$ENOB = \frac{(39 - 1,76dB)}{6,02} = 6,18 \text{ bits}$$

On peut donc conclure que seule la correction de l'erreur Δt_k pose problème. Avec la méthode d'interpolation on arrive à corriger en partie cette erreur avec un gain de 14,6dB avec un signal d'entrée ayant une fréquence de 1/10. La méthode d'interpolation n'est donc pas très optimale.

Cependant la figure 28 montre que cette méthode d'interpolation s'avère de plus en plus efficace quand la fréquence du signal d'entrée f_0 diminue par rapport à f_s .

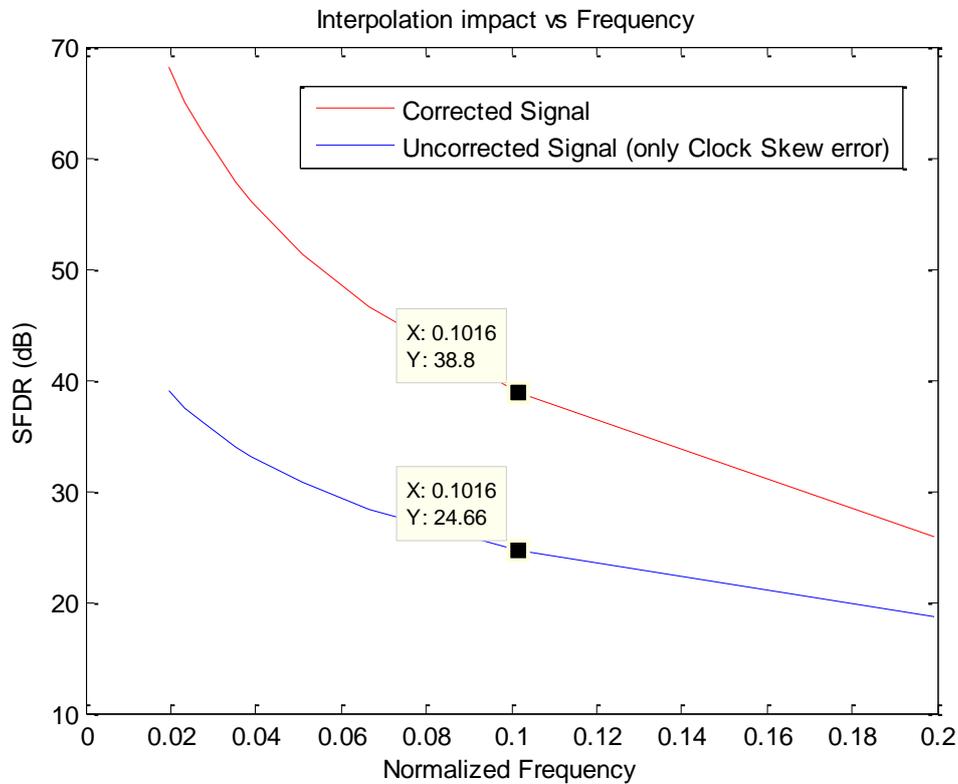


Figure 28 – Impact de la fréquence d'entrée du TIADC (avec Δt_k) sur le SFDR du signal numérisé

On réalise alors une simulation avec f_0 plus petit devant f_s . On choisi $f_0=f_s/50$ (cf. Fig. 29)

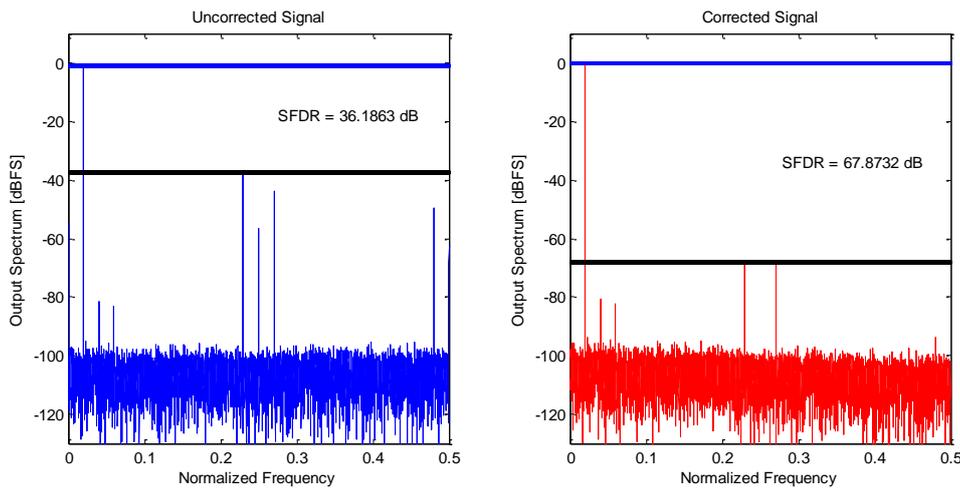


Figure 29 - Spectre du signal TIADC (avec G_k , O_k et Δt_k) non corrigé puis corrigé par interpolation

Sur la figure 29 on remarque bien que la méthode d'interpolation a bien atténué l'erreur Δt_k . On obtient alors un ENOB de 10 bits.

Cependant il faut noter que l'erreur de Clock-Skew que nous avons intégré aux différents ADC est très importante. En effet, elle est de l'ordre de 10% à 30% de T_s . Nous pouvons donc penser qu'avec une erreur de Clock Skew plus petite (proche de la réalité) on aura un SFDR plus grand.

V. Implémentation sur FPGA

La partie précédente vient de montrer que l'algorithme de calibration et de correction des erreurs de disparité fonctionne correctement en simulation sous MATLAB-SIMULINK. L'intérêt maintenant est de développer cette solution de manière physique afin de valider notre solution. Pour cela nous allons implémenter notre algorithme sur un FPGA.

V.1. Mise en place du test de notre méthode

L'objectif étant de tester seulement notre algorithme nous allons réaliser le test de la manière suivante : (cf. Fig. 30)

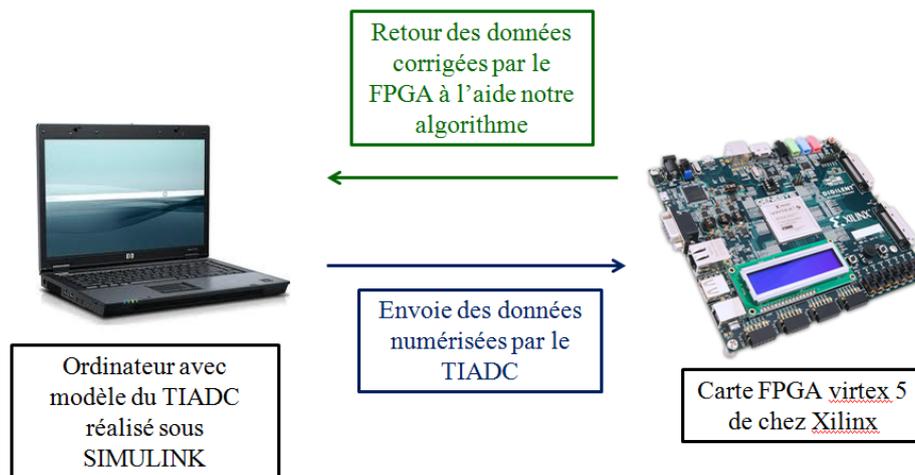


Figure 30 - Schéma bloc de test de la méthode de correction implémenté sur FPGA

Le TIADC ne sera pas un composant physique. En effet nous utiliserons les données de notre modèle '*TIADC_sim*' que nous devons relier à la carte FPGA par une liaison série. Le FPGA utilisé est un Virtex 5 de la société Xilinx [8].

- Choix des paramètres de test :

Tout d'abord, afin de se faciliter les calculs, nous avons décidé de changer notre modèle des ADC de '*TIADC_sim*'. En effet comme on peut le voir sur la figure 31 les données de sortie de chaque ADC seront un entier positif représentant le nombre de quantum.

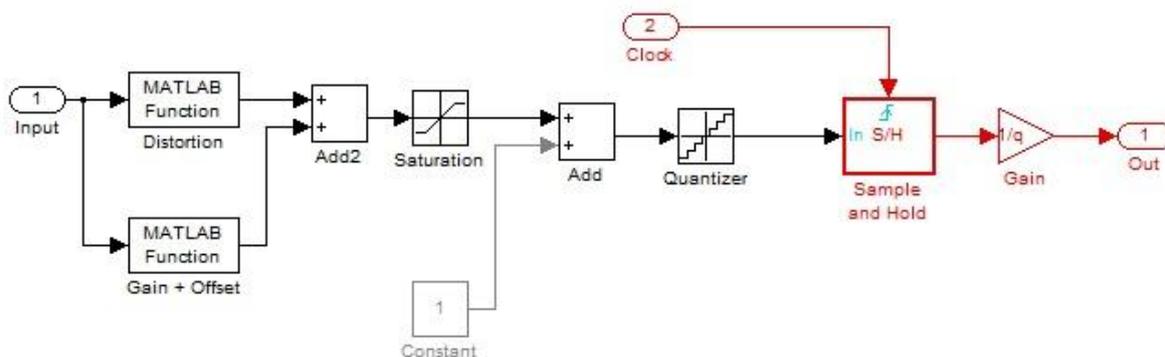


Figure 31 - Modèle des ADC utilisés pour réaliser le test sur FPGA

Ensuite, nous avons dû fixer la résolution à 14bits car dans notre algorithme nous effectuons des multiplications. Or, si on multiplie deux nombres de 14 bits on obtient alors un nombre sur 28 bits et le FPGA n'accepte pas de nombre de plus de 32 bits. De plus, on a limité à 14 bits en pensant que nous aurons peut-être besoin des 4 bits supplémentaires par la suite.

Le nombre d'échantillons est lui fixé à 64, valeur minimale pour avoir une estimation et une correction correcte. (cf. Fig. 32) Les paramètres de simulation sont représentés dans le tableau 5

f_s	f_o "wanted"	L	N_{ech}	N_{fft}	SNR_{in}	A
1	1/10	6	64	256	100dB	0,9

Tableau 5 - Paramètre de test tu TIADC avec correction sur FPGA

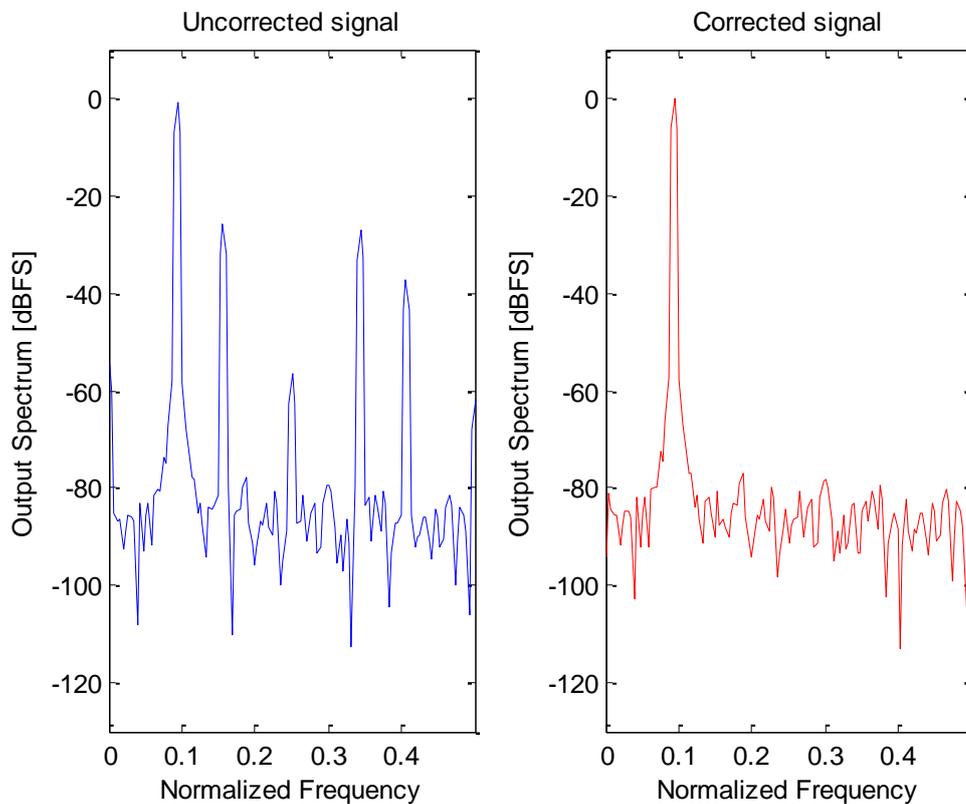


Figure 32 – Calibration du TIADC avec 64 échantillons

V.2. Décomposition du travail en modules et fonctions

Pour réaliser notre test il est nécessaire de le décomposer en plusieurs fonctions, chacune associées à un module. Nous présentons ici le principe général de ces fonctions afin de mieux comprendre les descriptions détaillées de la partie V.3.

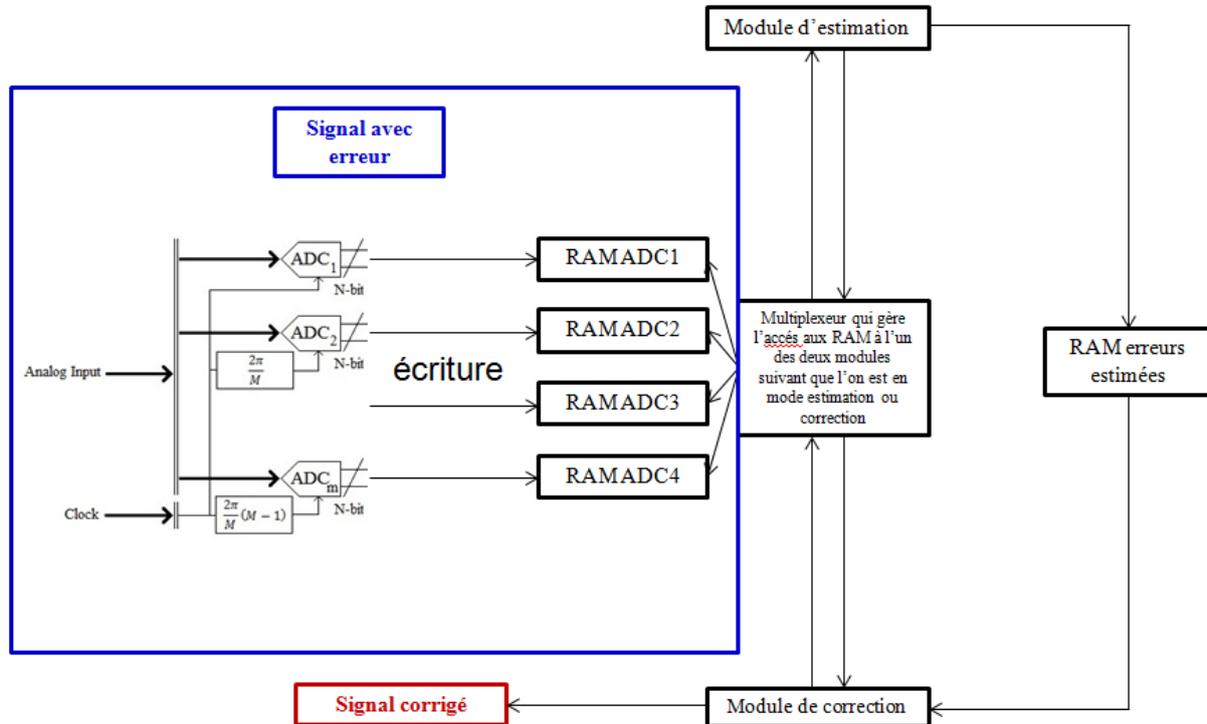


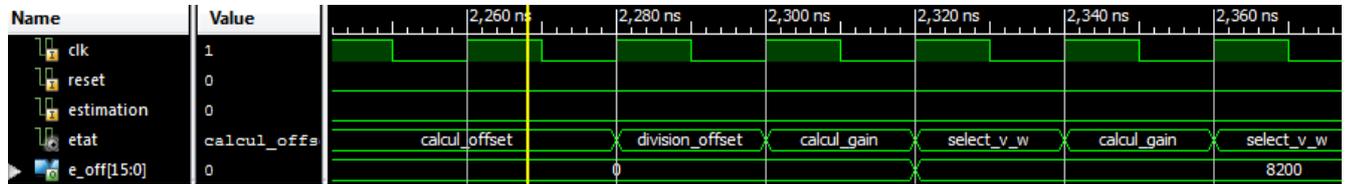
Figure 33 - Schéma simplifié du fonctionnement de notre système sur FPGA

Nous avons tout d'abord besoin d'un ensemble de RAM double accès **RAM double** qui contient les données à traiter. Cette RAM permet d'écrire et de lire en même temps, ce qui nous permet de réaliser une correction en temps réel avec un retard constant entre le signal d'entrée et le signal corrigé (dû aux différentes opérations). Cette RAM doit fournir un espace de stockage de 64 données de 16 bits.

... pas fini

V.3. Simulations

Pas fini



VI. Conclusion

Les travaux réalisés durant ce stage ont touché l'étude, la modélisation et la calibration des erreurs de disparité des convertisseurs analogiques numériques entrelacés dans le temps. Dans un contexte plus général, l'objectif de ce mémoire est de proposer une architecture de conversion assurant à la fois une bonne résolution et une grande vitesse. En effet, les systèmes de communications sans fils sont en expansion et les exigences sur les circuits de conversion sont de plus en plus sévères.

Dans un premier temps, l'étude des systèmes de réception de radio logicielle [1] a montré l'exigence d'un convertisseur permettant la conversion des signaux ayant une grande dynamique libre et une large bande passante. Or, les architectures de conversion basées sur un seul convertisseur traditionnel [4] ne peuvent pas garantir ces demandes. Cependant, une nouvelle architecture prometteuse basée sur l'entrelacement temporel de plusieurs convertisseurs permet d'augmenter la fréquence d'échantillonnage et par conséquent assurer la conversion des signaux large bande. L'amélioration de la résolution de cette architecture était l'objet de nos travaux durant ce mastère.

Les objectifs fixés ont été atteints car nous avons réussi à améliorer l'algorithme afin de diminuer la part de composant analogique dans le TIADC. Notre algorithme d'estimation est très précis et la correction permet d'améliorer considérablement l'ENOB du TIADC. De plus nous avons montré que le nombre d'échantillons (nécessaire à l'estimation des erreurs) n'a pas besoin d'être trop important d'où un gain en temps, mais également en ressource mémoire et donc en coût.

Les performances des convertisseurs entrelacés dépendent de celles des ADC. Il est indispensable de garder en tête l'évolution des ADC et les difficultés d'allier fréquence d'échantillonnage, résolution et consommation de puissance. L'entrelacement de plusieurs convertisseurs permet certes d'augmenter la cadence d'échantillonnage du système mais, les méthodes de correction existantes ou proposées ne permettent que de se rapprocher des résolutions effectives des ADC.

Le choix des convertisseurs à entrelacer dépend d'abord de l'application en question. Le nombre de convertisseurs à entrelacer est théoriquement sans limite, mais réellement la bande passante et la puissance consommée limitent le nombre d'ADC utilisés dans le TIADC. Dans le cas de la radio logicielle par exemple, la résolution requise par les ADC est estimée à 18 bits. Les ADC offrant cette résolution ne dépassent pas les quelques MS/s en fréquence d'échantillonnage et le nombre de ADC à entrelacer devient exorbitant pour atteindre la fréquence d'échantillonnage requise. Ainsi, des efforts sont indispensables pour améliorer le facteur de mérite des ADC. Ensuite, les méthodes de correction, telles que celles que nous venons de proposer pourra permettre de se rapprocher au mieux des exigences de la radio logicielle.

Il serait bien sûr intéressant de continuer le travail sur la correction de l'erreur de retard à l'ouverture. En effet l'estimation des erreurs fonctionne très bien (en simulation sous MATLAB-SIMULINK) mais le problème reste à trouver une méthode permettant d'utiliser l'estimation de Clock-Skew afin de le corriger parfaitement. Pour l'instant notre méthode d'interpolation ne permet que de corriger une partie de cette erreur.

Pour le test sur FPGA, je pense qu'il n'y aura pas de problème en ce qui concerne l'Offset et le Gain car nous avons presque fini et cela semble fonctionner. Le problème risque de venir de l'estimation et de la correction du Clock Skew car les valeurs sont très petites et présentent des virgules. Ceci va rendre les calculs bien plus difficiles à réaliser.

Ce Projet de Fin d'Etudes m'a permis d'acquérir de bonnes connaissances sur le domaine des ADC ainsi que sur l'utilisation du logiciel MATLAB-SIMULINK. La seule chose que je regrette est d'avoir perdu beaucoup de temps sur la liaison série entre MATLAB et le FPGA. Ceci, m'a empêché sur la fin de pouvoir implémenter tout l'algorithme sur FPGA. En effet, je pense qu'avec une ou deux semaines supplémentaires j'aurais pu terminer l'implémentation de notre algorithme sur le FPGA permettant l'estimation et la correction des erreurs de gain et d'offset. Ce sera plus difficile pour le Clock Skew.

Finalement, la réalisation de ce PFE a été pour moi une grande satisfaction, aussi bien personnellement que professionnellement. J'ai pu découvrir le milieu de la Recherche en laboratoire universitaire publique et cela m'a donné l'envie de réaliser une thèse. D'un point de vu relationnelle aussi ce stage a été très enrichissant car il m'a permis de rencontré des personnes avec qui j'ai pris plaisir à travailler.

VII. BIBLIOGRAPHIE

- [1] RIVET François, MARIANO André Augusto, DEVAL Yann DALLET Dominique, BÉGUERET Jean-Baptiste, “Sample Analog Signal Processing: From Software-Defined to Software Radio”, EMERGING TECHNOLOGIES AND CIRCUITS, 2010, Springer, pp. 261-287, ISBN: 9789048193783.
- [2] G. Ferré, B. L. Gal, L. Bossuet, M. Jridi, and D. Dallet, “Orthogonal Correction Implementation for Time-Interleaved Analog-To-Digital Converters: Realtime Application”. In Proceedings of the 16th EURASIP European Conference on Signal Processing, August 2008.
- [3] RIVET François, MARIANO André Augusto, DALLET Dominique, BÉGUERET Jean-Baptiste, “Mixed-Signal Built-in Self-Calibrated Time-Interleaved ADC in 65nm CMOS Technology”, IEEE INTERNATIONAL NEWCAS CONFERENCE – NEWCAS 2010, Montreal – Canada.
- [4] MARIANO André Augusto, Thèse présentée à l’université Bordeaux 1, “Mixed Simulations and Design of a Wideband Continuous-Time Bandpass Delta-Sigma Converter Dedicated to Software Defined Radio Applications”, Octobre 2008.
- [5] Pochon Jean-François, <http://www.electrons.ch/Convertisseur%20AD%20Ver1.2.pdf>, CPNV, Février 2005.
- [6] NXP Semiconductors, Datasheet, Dual 14-bit ADC, ADC1412D, Mars 2011.
- [7] Volnei A. Pedroni, “Circuit design with VHDL”, 2004 Massachusetts Institute of Technology.
- [8] DIGILENT, “GenesysTM Board Reference Manual”, February 28, 2012.
- [9] Pong P. Chu, “FPGA Prototyping by VHDL Examples”, Chapter 7, pp. 163-182, Copyright 2008 John Wiley & Sons, Inc.

VIII. ANNEXES

VIII.1. Caractéristiques des convertisseurs analogique-numérique

VIII.2. Simulation avec un signal modulé en amplitude

f_s	f_0 "wanted"	f_1 "wanted"	L	N_{ech}	N_{fft}	SNR_{in}	A
1	1/10	1/300	12	4096	16384	100dB	0,9

